

Relatório (complemento) sobre o Código Fonte do Projeto, Conjunto Ordenado

Uso da estrutura de dados, Árvore Red Black
UFES Centro Universitário Norte do Espírito Santo

Joilnen Leite
joilnen.leite@edu.ufes.br

Resumo: Relatório (complemento) sobre o conteúdo e processo de desenvolvimento da atividade conjunto ordenado

Palavras-chave: relatório, fontes, C, Red Black, Conjunto

Informações complementares relativas ao processo de desenvolvimento, constam aqui para registro do trabalho feito, durante quase dois meses diariamente, o relatório principal encontra-se no arquivo **relatorio.pdf**.

1. Histórico do desenvolvimento
2. Documentação gerada

```

1 commit 9aa5edde767e93157a9ece4b031316167fcf50e6
2 Author: Joilnen <joilnen@gmail.com>
3 Date: Thu Dec 16 00:03:59 2021 -0300
4
5 trabalho de ed2
6
7 diff --git a/.gitignore b/.gitignore
8 new file mode 100644
9 index 00000000..c442bb3
10 --- /dev/null
11 +++ b/.gitignore
12 @@ -0,0 +1,6 @@
13 *.sw*
14 +-
15 *.o
16 *.d
17 +tags
18 +rodeme
19 diff --git a/doc/spec.txt b/doc/spec.txt
20 new file mode 100644
21 index 00000000..9ceb8b9
22 --- /dev/null
23 +++ b/doc/spec.txt
24 @@ -0,0 +1,15 @@
25 +TAD Conjunto Ordenado utilizando a estrutura árvore Rubro-Negra para tipos de dados genéricos.
26 +deve ser implementado para tipos de dados genéricos
27 +o programador, usuário da biblioteca, define o tipo dos elementos do conjunto.
28 + Imprimir os elementos de um conjunto.
29 + Cardinalidade de um conjunto
30 + Verificar se um conjunto é vazio.
31 + Esvaziar conjunto:
32 + Inserir um novo elemento em um conjunto (não deve existir elementos repetidos no conjunto).
33 + Excluir um elemento de um conjunto.
34 + Verificar se um elemento pertence ao conjunto.
35 +
36 +A intesec B
37 +A uniao B
38 +A menos B
39 +Impressao da arvore chave seguida da cor
40 diff --git a/src/Makefile b/src/Makefile
41 new file mode 100644
42 index 00000000..4c47828
43 --- /dev/null
44 +++ b/src/Makefile
45 @@ -0,0 +1,16 @@
46 +#
47 +# Aluno: Joilnen Leite
48 +#
49 +
50 +C=gcc
51 +S=main.c conjunto_ordenado.c
52 +O=main.o conjunto_ordenado.o
53 +B=rodeme
54 +
55 +.c.o:
56 + $(C) -c $<
57 +all: main.o conjunto_ordenado.o
58 + $(C) -o $(B) $(O)
59 +clean:
60 + rm -f *.o $(B)
61 +
62 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
63 new file mode 100644
64 index 00000000..da7183d
65 --- /dev/null
66 +++ b/src/conjunto_ordenado.c
67 @@ -0,0 +1,112 @@
68 +#include "conjunto_ordenado.h"
69 +
70 +int imprime_conjunto(rb_n *a)
71 +{
72 +
73 + return 0;
74 +}
75 +
76 +unsigned int cardinalidade(rb_n *a)
77 +{
78 +
79 + return 0;
80 +}
81 +
82 +int ta_vazio(rb_n *a)
83 +{
84 +
85 + return 0;
86 +}
87 +
88 +int limpa_conjunto(rb_n *a)
89 +{
90 +
91 + return 0;
92 +}
93 +
94 +int insere_elemento(rb_n *a, int k, void *e)
95 +{
96 +
97 + return 0;
98 +}
99 +
100 +int exclui_elemento(rb_n *a, int k)
101 +{
102 +
103 + return 0;
104 +}
105 +
106 +int verifica_pertencimento(rb_n *a, int k)
107 +{
108 +
109 + return 0;
110 +}
111 +
112 +rb_n* intersecta_conjuntos(rb_n *a, rb_n *b)
113 +{
114 +
115 + return NULL;

```

```

116 +}
117 +
118 +rb_n* une_conjuntos(rb_n *a, rb_n *b)
119 +{
120 +
121 +    return NULL;
122 +}
123 +
124 +rb_n* subtrai_conjuntos(rb_n *a, rb_n *b)
125 +{
126 +    return NULL;
127 +}
128 +
129 +int destroi_conjunto(rb_n *a)
130 +{
131 +
132 +    return 0;
133 +}
134 +
135 +int rotaciona_esquerda(rb_n *r, rb_n *w)
136 +{
137 +    rb_n *v = w->r;
138 +    w->r = v->l;
139 +    if (v->l)
140 +        v->l->p = w;
141 +    v->p = w->p;
142 +    if (!w->p)
143 +        r = v;
144 +    else
145 +    {
146 +        if (w == w->p->l)
147 +            w->p->l = v;
148 +        else
149 +            w->p->r = v;
150 +    }
151 +    v->l = w;
152 +    w->p = v;
153 +
154 +    return 0;
155 +}
156 +
157 +int rotaciona_direita(rb_n *r, rb_n *w)
158 +{
159 +    rb_n *v = w->l;
160 +    w->l = v->r;
161 +    if (v->r)
162 +        v->r->p = w;
163 +    v->p = w->p;
164 +    if (!w->p)
165 +        r = v;
166 +    else
167 +    {
168 +        if (w == w->p->r)
169 +            w->p->r = v;
170 +        else
171 +            w->p->l = v;
172 +    }
173 +    v->r = w;
174 +    w->p = v;
175 +
176 +    return 0;
177 +}
178 +
179 +
180 +diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
181 +new file mode 100644
182 +index 0000000..8bcfb59
183 +--- /dev/null
184 +++ b/src/conjunto_ordenado.h
185 +@@ -0,0 +1,56 @@
186 + #ifndef CONJUNTO_ORDENADO_H
187 + #define CONJUNTO_ORDENADO_H
188 +
189 + /**
190 +  * Rubro Negra
191 +  * 0 - todo noh eh vermelho ou preto
192 +  * 1 - raiz sempre preta
193 +  * 2 - noh NULL considerado noh externo preto
194 +  * 3 - vermelhos tem filhos pretos
195 +  * 4 - noh para suas folhas tem o mesmo numero de nohs preto
196 +  */
197 +
198 + /**
199 +  * Composicao do noh da RB, apenas a parte
200 +  * @rb_n noh red black
201 +  * @k chave (k de key)
202 +  * @c cor (c de color ou cor, 0-black, 1-red)
203 +  * @d dados (de de data ou dados)
204 +  * @l @r @p l-eft filho a esquerda, r-ight filho a direita e p-parent, noh pai
205 +  */
206 +
207 + #include <stdlib.h>
208 +
209 + typedef struct rb_n {
210 +     int k;
211 +     char c;
212 +     void *d;
213 +     struct rb_n *l, *r, *p;
214 + } rb_n;
215 +
216 + #define CRIA_CONJUNTO_ORDENADO(a) \
217 +     rb_n a; \
218 +     a.c = 0; \
219 +     a.p = a.l = a.r = NULL;
220 +
221 + int imprime_conjunto(rb_n *a);
222 + unsigned int cardinalidade(rb_n *a);
223 + int ta_vazio(rb_n *a);
224 + int limpa_conjunto(rb_n *a);
225 +
226 + int insere_elemento(rb_n *a, int k, void *e);
227 + int exclui_elemento(rb_n *a, int k);
228 + int verifica_pertencimento(rb_n *a, int k);
229 +
230 + rb_n* intersecta_conjuntos(rb_n *a, rb_n *b);
231 + rb_n* une_conjuntos(rb_n *a, rb_n *b);

```

```

232 +rb_n* subtrai_conjuntos(rb_n *a, rb_n *b);
233 +
234 +int destroi_conjunto(rb_n *a);
235 +
236 +int troca_cor(rb_n *a);
237 +int rotaciona_esquerda(rb_n *r, rb_n *w);
238 +int rotaciona_direita(rb_n *r, rb_n *w);
239 +
240 +#endif
241 +
242 diff --git a/src/main.c b/src/main.c
243 new file mode 100644
244 index 0000000..ed6c969
245 --- /dev/null
246 +++ b/src/main.c
247 @@ -0,0 +1,13 @@
248 +/**
249 + * Aluno: Joilnen Leite
250 + */
251 +#include "conjunto_ordenado.h"
252 +
253 +int main()
254 +{
255 +    CRIA_CONJUNTO_ORDENDADO(conjunto_a);
256 +
257 +
258 +}
259 +
260 +
261 commit dd1c9e21c5f930ad2c25712d0ae3c4417b882bf5
262 Author: Joilnen <joilnen@gmail.com>
263 Date: Thu Dec 16 04:41:33 2021 -0300
264
265     codificando como serah estipulado o tipo de dados
266
267 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
268 index da7183d..9f6ff26 100644
269 --- a/src/conjunto_ordenado.c
270 +++ b/src/conjunto_ordenado.c
271 @@ -1,3 +1,7 @@
272 +/**
273 + * aluno: joilnen leite
274 + */
275 +
276 +
277 +#include "conjunto_ordenado.h"
278 +
279 +int imprime_conjunto(rb_n *a)
280 @@ -26,7 +30,20 @@ int limpa_conjunto(rb_n *a)
281 +
282 +int insere_elemento(rb_n *a, int k, void *e)
283 +{
284 +
285 +    rb_n *it = a;
286 +    while (it)
287 +    {
288 +        if (k < it->k)
289 +            it = it->l;
290 +        else if (k > it->k)
291 +            it = it->r;
292 +        if (!it)
293 +        {
294 +            it = e;
295 +            it->k = k;
296 +            it->l = it->r = NULL;
297 +        }
298 +    }
299 +    return 0;
300 +}
301 +
302 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
303 index 8bcfb59..3d839ee 100644
304 --- a/src/conjunto_ordenado.h
305 +++ b/src/conjunto_ordenado.h
306 @@ -1,6 +1,10 @@
307 +#ifndef CONJUNTO_ORDENADO_H
308 +#define CONJUNTO_ORDENADO_H
309 +
310 +/**
311 + * aluno: joilnen leite
312 + */
313 +
314 +/**
315 + * Rubro Negra
316 + * 0 - todo noh eh vermelho ou preto
317 @@ -15,6 +19,7 @@
318 + * @rb_n noh red black
319 + * @k chave (k de key)
320 + * @c cor (c de color ou cor, 0-black, 1-red)
321 + * @t tipo (t de tipo ou type)
322 + * @d dados (de de data ou dados)
323 + * @l @r @p l-efit filho a esquerda, r-right filho a direita e p-arent, noh pai
324 + */
325 @@ -28,9 +33,10 @@ typedef struct rb_n {
326 +    struct rb_n *l, *r, *p;
327 +} rb_n;
328 +
329 +#define CRIA_CONJUNTO_ORDENDADO(a) \
330 +#define CRIA_CONJUNTO_ORDENDADO(a, dado) \
331 +    rb_n a; \
332 +    a.c = 0; \
333 +    a.d = &dado; \
334 +    a.p = a.l = a.r = NULL;
335 +
336 +int imprime_conjunto(rb_n *a);
337 @@ -52,5 +58,7 @@ int troca_cor(rb_n *a);
338 +int rotaciona_esquerda(rb_n *r, rb_n *w);
339 +int rotaciona_direita(rb_n *r, rb_n *w);
340 +
341 +int seta_imprime(void (*imprime_elemento)(rb_n *n));
342 +
343 +#endif
344 +
345 diff --git a/src/main.c b/src/main.c
346 index ed6c969..f8dc951 100644
347 --- a/src/main.c

```

```

348 +++ b/src/main.c
349 @@ -5,7 +5,8 @@
350
351 int main()
352 {
353     CRIA_CONJUNTO_ORDENDADO(conjunto_a);
354     char letra = 'a';
355     CRIA_CONJUNTO_ORDENDADO(conjunto_a, letra);
356
357 }
358
359 commit cc53fc091c7f75af80b167978ab1a14f6fc801d0
360 Author: Joilnen <joilnen@gmail.com>
361 Date: Thu Dec 16 05:16:35 2021 -0300
362
363     criando funcoes de impressao setada pelo cliente
364
365 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
366 index 9f6ff26..f7a995c 100644
367 --- a/src/conjunto_ordenado.c
368 +++ b/src/conjunto_ordenado.c
369 @@ -4,6 +4,8 @@
370
371 #include "conjunto_ordenado.h"
372
373 void (*imprime_elemento)(rb_n *);
374
375 int imprime_conjunto(rb_n *a)
376 {
377     @@ -126,4 +128,9 @@ int rotaciona_direita(rb_n *r, rb_n *w)
378     return 0;
379 }
380
381 int seta_imprime(void (*i)(rb_n *n))
382 {
383     imprime_elemento = i;
384 }
385
386 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
387 index 3d839ee..44b686b 100644
388 --- a/src/conjunto_ordenado.h
389 +++ b/src/conjunto_ordenado.h
390 @@ -58,7 +58,8 @@ int troca_cor(rb_n *a);
391 int rotaciona_esquerda(rb_n *r, rb_n *w);
392 int rotaciona_direita(rb_n *r, rb_n *w);
393
394 void (*imprime_elemento)(rb_n *n);
395 extern void (*imprime_elemento)(rb_n *);
396 int seta_imprime(void (*i)(rb_n *n));
397
398 #endif
399
400 diff --git a/src/main.c b/src/main.c
401 index f8dc951..251026e 100644
402 --- a/src/main.c
403 +++ b/src/main.c
404 @@ -1,12 +1,49 @@
405
406 /**
407  * Aluno: Joilnen Leite
408  * aluno: joilnen leite
409  */
410
411 /**
412  * inclusao da biblioteca implementada baseada em arvores red black
413  */
414 #include "conjunto_ordenado.h"
415
416 /**
417  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
418  * @noh rb
419  */
420 void imprime_meu_tipo(rb_n *);
421
422 /**
423  * funcao main de teste separada do codigo da biblioteca como especificado
424  */
425 int main()
426 {
427     char letra = 'a';
428     CRIA_CONJUNTO_ORDENDADO(conjunto_a, letra);
429
430     /**
431      * definicao de um tipo exemplo pelo o usuario
432      */
433     typedef struct {
434         char letra;
435     } meu_tipo;
436     meu_tipo mt;
437
438     /**
439      * criacao do dado para o tipo do usuario
440      */
441     mt.letra = 'a';
442
443     /**
444      * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
445      */
446     seta_imprime(imprime_meu_tipo);
447
448     /**
449      * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
450      * uma interface mais intuitiva ao usuario da biblioteca
451      */
452     CRIA_CONJUNTO_ORDENDADO(conjunto_a, mt);
453 }
454
455 void imprime_meu_tipo(rb_n *n)
456 {
457 }
458
459 commit 833960f5389fd02becfc02e1083ceab3899a9f24
460 Author: Joilnen <joilnen@gmail.com>
461 Date: Thu Dec 16 05:19:03 2021 -0300

```

```

464
465      adicionado comentario a definicao de funcao, comentando tando declaracao
466      cabecalhos como definicoes
467
468 diff --git a/src/main.c b/src/main.c
469 index 251026e..9d40f24 100644
470 --- a/src/main.c
471 +++ b/src/main.c
472 @@ -42,6 +42,10 @@ int main()
473     CRIA_CONJUNTO_ORDENADO(conjunto_a, mt);
474 }
475
476 /**
477  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
478  * exemplo
479  */
480 void imprime_meu_tipo(rb_n *n)
481 {
482
483
484 commit 9f1ce14043c169c8e0c170d7a40b6f153aade3a8
485 Author: Joilnen <joilnen@gmail.com>
486 Date: Thu Dec 16 05:25:13 2021 -0300
487
488      usando um nome melhor seta_imprime_elemento
489
490 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
491 index f7a995c..40f65e1 100644
492 --- a/src/conjunto_ordenado.c
493 +++ b/src/conjunto_ordenado.c
494 @@ -128,7 +128,7 @@ int rotaciona_direita(rb_n *r, rb_n *w)
495     return 0;
496 }
497
498 -int seta_imprime(void (*i)(rb_n *n))
499 +int seta_imprime_elemento(void (*i)(rb_n *n))
500 {
501     imprime_elemento = i;
502 }
503
504 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
505 index 44b686b..e084108 100644
506 --- a/src/conjunto_ordenado.h
507 +++ b/src/conjunto_ordenado.h
508 @@ -6,7 +6,7 @@
509 */
510
511 /**
512  * Rubro Negra
513  * rubro negra
514  * 0 - todo noh eh vermelho ou preto
515  * 1 - raiz sempre preta
516  * 2 - noh NULL considerado noh externo preto
517 @@ -15,15 +15,14 @@
518 */
519
520 /**
521  * Composicao do noh da RB, apenas a parte
522  * composicao do noh da RB
523  * @rb_n noh red black
524  * @k chave (k de key)
525  * @c cor (c de color ou cor, 0-black, 1-red)
526  * @t tipo (t de tipo ou type)
527  * @d dados (de de data ou dados)
528  * @d dado (de de data ou dados)
529  * @l @r @p l-eft filho a esquerda, r-right filho a direita e p-arent, noh pai
530 */
531
532 -#include <stdlib.h>
533
534 typedef struct rb_n {
535     @@ -59,7 +58,8 @@ int rotaciona_esquerda(rb_n *r, rb_n *w);
536     int rotaciona_direita(rb_n *r, rb_n *w);
537
538     extern void (*imprime_elemento)(rb_n *);
539     -int seta_imprime(void (*i)(rb_n *n));
540     +int seta_imprime_elemento(void (*i)(rb_n *n));
541
542 #endif
543
544 +
545 diff --git a/src/main.c b/src/main.c
546 index 9d40f24..65afc58 100644
547 --- a/src/main.c
548 +++ b/src/main.c
549 @@ -33,7 +33,7 @@ int main()
550     /**
551      * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
552      */
553     - seta_imprime(imprime_meu_tipo);
554     + seta_imprime_elemento(imprime_meu_tipo);
555
556     /**
557      * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
558
559 commit e5821185616c3a11f9b74159e21f84b70a40fcb0
560 Author: Joilnen <joilnen@gmail.com>
561 Date: Thu Dec 16 05:26:26 2021 -0300
562
563      cosmetico
564
565 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
566 index 40f65e1..e66183c 100644
567 --- a/src/conjunto_ordenado.c
568 +++ b/src/conjunto_ordenado.c
569 @@ -1,7 +1,6 @@
570 /**
571  * aluno: joilnen leite
572  */
573
574 -#include "conjunto_ordenado.h"
575
576 void (*imprime_elemento)(rb_n *);
577
578 commit df0a1689321c768d19059b441654168d9050b512
579 Author: Joilnen <joilnen@gmail.com>
580 Date: Thu Dec 16 05:32:18 2021 -0300

```

```

580
581     mudancas cosmeticas em um comentario
582
583 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
584 index e66183c..939de06 100644
585 --- a/src/conjunto_ordenado.c
586 +++ b/src/conjunto_ordenado.c
587 @@ -1,5 +1,7 @@
588 /**
589  * aluno: joilnen leite
590  * * arquivo separado do main, como especificado
591  * * de implementacao das funcoes da biblioteca conjunto ordenado
592  */
593 #include "conjunto_ordenado.h"
594
595
596 commit 9db7458625c8897a3f131e320a6c1a2538a2d75c
597 Author: Joilnen <joilnen@gmail.com>
598 Date: Thu Dec 16 06:03:29 2021 -0300
599
600     adicionado instrucoes e modificado o padrao do makefile para o mostrado em aula
601
602 diff --git a/LEIAME b/LEIAME
603 new file mode 100644
604 index 0000000..17b9ab4
605 --- /dev/null
606 +++ b/LEIAME
607 @@ -0,0 +1,11 @@
608 +dentro do diretorio docs estao relatorios e arquivos
609 +relacionados ao processo de desenvolvimento
610 +
611 +binario sera gerado no mesmo diretorio dos fontes,
612 +src, atraves do comando make local
613 +dentro daquele diretorio, binario nomeado, rodeme
614 +comando a ser utilizado
615 +
616 +make # para criar o binario
617 +make clean # para limpar, apagar objetos e binario
618 +
619 diff --git a/src/Makefile b/src/makefile
620 similarity index 100%
621 rename from src/Makefile
622 rename to src/makefile
623
624 commit 8cc3261a0a8eccbe89725c0706a0e837b58807c3
625 Author: Joilnen <joilnen@gmail.com>
626 Date: Thu Dec 16 09:13:11 2021 -0300
627
628     adicionado alguns dados para teste no arquivo main.c
629
630 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
631 index e084108..aa30f43 100644
632 --- a/src/conjunto_ordenado.h
633 +++ b/src/conjunto_ordenado.h
634 @@ -32,8 +32,9 @@ typedef struct rb_n {
635     struct rb_n *l, *r, *p;
636 } rb_n;
637
638 #define CRIA_CONJUNTO_ORDENDADO(a, dado) \
639 #define CRIA_CONJUNTO_ORDENDADO(a, key, dado) \
640     rb_n a; \
641 + a.k = key; \
642 + a.c = 0; \
643 + a.d = &dado; \
644 + a.p = a.l = a.r = NULL;
645 diff --git a/src/main.c b/src/main.c
646 index 65afc58..941ff61 100644
647 --- a/src/main.c
648 +++ b/src/main.c
649 @@ -1,10 +1,12 @@
650 -/**
651 +/**
652  * aluno: joilnen leite
653  */
654 +
655 /**
656  * inclusao da bilioteca implementada baseada em arvores red black
657  */
658 #include "conjunto_ordenado.h"
659 #include <string.h>
660
661 /**
662  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
663 @@ -12,23 +14,50 @@
664  */
665 void imprime_meu_tipo(rb_n *);
666
667 /**
668  * funcao que adiciona alguns dados de teste
669  */
670 void adiciona_alguns_dados();
671
672 /**
673  * funcao main de teste separada do codigo da biblioteca como especificado
674  */
675 int main()
676 {
677     adiciona_alguns_dados();
678 }
679
680 /**
681  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
682  * exemplo
683  */
684 void imprime_meu_tipo(rb_n *n)
685 {
686     +
687     +
688     +
689     +
690 /**
691  * funcao que adiciona alguns dados de teste
692  */
693 void adiciona_alguns_dados()
694 {
695     /**

```

```

696     * definicao de um tipo exemplo pelo o usuario
697     */
698     typedef struct {
699     -     char letra;
700     +     char *nome;
701     +     char *telefone;
702     } meu_tipo;
703     meu_tipo mt;
704
705     /**
706     * criacao do dado para o tipo do usuario
707     */
708     -     mt.letra = 'a';
709     +     mt.nome = (char *) malloc(sizeof("nome para teste") + 1);
710     +     strncpy(mt.nome, "nome para teste", sizeof("nome para teste") + 1);
711     +     mt.telefone = (char *) malloc(sizeof("999999999") + 1);
712     +     strncpy(mt.telefone, "999999999", sizeof("999999999") + 1);
713
714     /**
715     * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
716     @@ -38,18 +67,11 @@ int main()
717     /**
718     * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
719     * uma interface mais intuitiva ao usuario da biblioteca
720     +     @parametro 0 nome variavel do noh raiz
721     +     @parametro 1 chave do noh raiz
722     +     @parametro 2 dado criado neste noh, pode ser de qqr tipo
723     */
724     -     CRIA_CONJUNTO_ORDENDADO(conjunto_a, mt);
725     -}
726
727     -/**
728     * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
729     * exemplo
730     */
731     -void imprime_meu_tipo(rb_n *n)
732     -{
733     -
734     -
735     +     CRIA_CONJUNTO_ORDENDADO(conjunto_a, 1, mt);
736     }
737
738
739
740     commit 0f24a3c6a52d925c62c1bf6ab19043f2313f35a7
741     Author: Joilnen <joilnen@gmail.com>
742     Date: Thu Dec 16 11:31:03 2021 -0300
743
744     rearrumando as funcoes no .h
745
746     diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
747     index aa30f43..d14b82e 100644
748     --- a/src/conjunto_ordenado.h
749     +++ b/src/conjunto_ordenado.h
750     @@ -4,6 +4,7 @@
751     /**
752     * aluno: joilnen leite
753     */
754     #include <stdlib.h>
755
756     /**
757     * rubro negra
758     @@ -23,8 +24,6 @@
759     * @d dado (de de data ou dados)
760     * @l @r @p l-est filho a esquerda, r-right filho a direita e p-parent, noh pai
761     */
762     #include <stdlib.h>
763
764     typedef struct rb_n {
765     int k;
766     char c;
767     @@ -32,6 +31,9 @@ typedef struct rb_n {
768     struct rb_n *l, *r, *p;
769     } rb_n;
770
771     /**
772     * cria o primeiro noh, o raiz da arvore rb
773     */
774     #define CRIA_CONJUNTO_ORDENDADO(a, key, dado) \
775     rb_n a; \
776     a.k = key; \
777
778     commit da567c4af9c079722d2141fe10d23c5d64bcf210
779     Author: Joilnen <joilnen@gmail.com>
780     Date: Thu Dec 16 16:11:40 2021 -0300
781
782     implementando insere noh na rb e modelo do relatorio adicionado
783
784     diff --git a/.gitignore b/.gitignore
785     index c442bb3..0446a97 100644
786     --- a/.gitignore
787     +++ b/.gitignore
788     @@ -4,3 +4,4 @@
789     *.d
790     tags
791     rodeme
792     +latex.out
793     diff --git a/doc/tex/fig/golpatterns.png b/doc/tex/fig/golpatterns.png
794     new file mode 100644
795     index 0000000..1738c62
796     Binary files /dev/null and b/doc/tex/fig/golpatterns.png differ
797     diff --git a/doc/tex/fig/lista.png b/doc/tex/fig/lista.png
798     new file mode 100644
799     index 0000000..a674c6c
800     Binary files /dev/null and b/doc/tex/fig/lista.png differ
801     diff --git a/doc/tex/fig/ontologia.pdf b/doc/tex/fig/ontologia.pdf
802     new file mode 100644
803     index 0000000..f0041bc
804     Binary files /dev/null and b/doc/tex/fig/ontologia.pdf differ
805     diff --git a/doc/tex/fig/ontologia1.eps b/doc/tex/fig/ontologia1.eps
806     new file mode 100644
807     index 0000000..caa2853
808     --- /dev/null
809     +++ b/doc/tex/fig/ontologia1.eps
810     @@ -0,0 +1,831 @@
811     %!PS-Adobe-3.0 EPSF-3.0

```



```

812|%%Title: ontologia_diagrama_bkp.fig
813|%%Creator: fig2dev Version 3.2.7b
814|%%CreationDate: 2020-12-11 09:20:12
815|%%BoundingBox: 0 0 409 369
816|%%Magnification: 1.0000
817|%%EndComments
818|%%BeginProlog
819|/$F2psDict 200 dict def
820|/$F2psDict begin
821|/$F2psDict /mtrx matrix put
822|+/col-1 {0 setgray} bind def
823|+/col0 {0.000 0.000 0.000 srgb} bind def
824|+/col7 {1.000 1.000 1.000 srgb} bind def
825|+/col16 {0.000 0.690 0.690 srgb} bind def
826|+/col27 {1.000 0.500 0.500 srgb} bind def
827|+/col29 {1.000 0.750 0.750 srgb} bind def
828|+
829|+end
830|+
831|+/cp {closepath} bind def
832|+/ef {eofill} bind def
833|+/gr {grestore} bind def
834|+/gs {gsave} bind def
835|+/sa {save} bind def
836|+/rs {restore} bind def
837|+/l {lineto} bind def
838|+/rl {rlineto} bind def
839|+/m {moveto} bind def
840|+/rm {rmoveto} bind def
841|+/n {newpath} bind def
842|+/s {stroke} bind def
843|+/sh {show} bind def
844|+/slc {setlinecap} bind def
845|+/slj {setlinejoin} bind def
846|+/slw {setlinewidth} bind def
847|+/srgb {setrgbcolor} bind def
848|+/rot {rotate} bind def
849|+/sc {scale} bind def
850|+/sd {setdash} bind def
851|+/ff {findfont} bind def
852|+/sf {setfont} bind def
853|+/scf {scalefont} bind def
854|+/sw {stringwidth} bind def
855|+/tr {translate} bind def
856|+/tnt {dup dup currentrgbcolor
857|+ 4 -2 roll dup 1 exch sub 3 -1 roll mul add
858|+ 4 -2 roll dup 1 exch sub 3 -1 roll mul add
859|+ 4 -2 roll dup 1 exch sub 3 -1 roll mul add srgb}
860|+ bind def
861|+/shd {dup dup currentrgbcolor 4 -2 roll mul 4 -2 roll mul
862|+ 4 -2 roll mul srgb} bind def
863|+/reencdict 12 dict def /ReEncode { reencdict begin
864|+/newcodesandnames exch def /newfontname exch def /basefontname exch def
865|+/basefontdict basefontname findfont def /newfont basefontdict maxlength dict def
866|+/basefontdict { exch dup /FID ne { dup /Encoding eq
867|+{ exch dup length array copy newfont 3 1 roll put }
868|+{ exch newfont 3 1 roll put } ifelse } { pop pop } ifelse } forall
869|+/newfont /FontName newfontname put newcodesandnames aload pop
870|+128 1 255 { newfont /Encoding get exch /.notdef put } for
871|+/newcodesandnames length 2 idiv { newfont /Encoding get 3 1 roll put } repeat
872|+/newfontname newfont definefont pop end } def
873|+/isovec [
874|+8#005 /minus 8#200 /grave 8#201 /acute 8#202 /circumflex 8#203 /tilde
875|+8#204 /macron 8#205 /breve 8#206 /dotaccent 8#207 /dieresis
876|+8#210 /ring 8#211 /cedilla 8#212 /hungarumlaut 8#213 /ogonek 8#214 /caron
877|+8#220 /dotlessi 8#230 /oe 8#231 /OE
878|+8#240 /space 8#241 /exclamdown 8#242 /cent 8#243 /sterling
879|+8#244 /currency 8#245 /yen 8#246 /brokenbar 8#247 /section 8#250 /dieresis
880|+8#251 /copyright 8#252 /ordfeminine 8#253 /guillemotleft 8#254 /logicalnot
881|+8#255 /hyphen 8#256 /registered 8#257 /macron 8#260 /degree 8#261 /plusminus
882|+8#262 /twosuperior 8#263 /threesuperior 8#264 /acute 8#265 /mu 8#266 /paragraph
883|+8#267 /periodcentered 8#270 /cedilla 8#271 /onesuperior 8#272 /ordmasculine
884|+8#273 /guillemotright 8#274 /onequarter 8#275 /onehalf
885|+8#276 /threequarters 8#277 /questiondown 8#300 /Agrave 8#301 /Acute
886|+8#302 /Acircumflex 8#303 /Atilde 8#304 /Adieresis 8#305 /Aring
887|+8#306 /AE 8#307 /Ccedilla 8#310 /Egrave 8#311 /Eacute
888|+8#312 /Ecircumflex 8#313 /Edieresis 8#314 /Igrave 8#315 /Iacute
889|+8#316 /Icircumflex 8#317 /Idieresis 8#320 /Eth 8#321 /Ntilde 8#322 /Ograve
890|+8#323 /Oacute 8#324 /Ocircumflex 8#325 /Otilde 8#326 /Odieresis 8#327 /multiply
891|+8#330 /Oslash 8#331 /Ugrave 8#332 /Uacute 8#333 /Ucircumflex
892|+8#334 /Udieresis 8#335 /Yacute 8#336 /Thorn 8#337 /germandbls 8#340 /agrave
893|+8#341 /aacute 8#342 /acircumflex 8#343 /atilde 8#344 /adieresis 8#345 /aring
894|+8#346 /ae 8#347 /ccedilla 8#350 /egrave 8#351 /eacute
895|+8#352 /ecircumflex 8#353 /edieresis 8#354 /igrave 8#355 /iacute
896|+8#356 /icircumflex 8#357 /idieresis 8#360 /eth 8#361 /ntilde 8#362 /ograve
897|+8#363 /oacute 8#364 /ocircumflex 8#365 /otilde 8#366 /odieresis 8#367 /divide
898|+8#370 /oslash 8#371 /ugrave 8#372 /uacute 8#373 /ucircumflex
899|+8#374 /udieresis 8#375 /yacute 8#376 /thorn 8#377 /ydieresis] def
900|+/Courier /Courier-iso isovec ReEncode
901|+ /DrawEllipse {
902|+ /endangle exch def
903|+ /startangle exch def
904|+ /yrad exch def
905|+ /xrad exch def
906|+ /y exch def
907|+ /x exch def
908|+ /savematrix mtrx currentmatrix def
909|+ x y tr xrad yrad sc 0 0 1 startangle endangle arc
910|+ closepath
911|+ savematrix setmatrix
912|+ } def
913|+
914|+/$F2psBegin {$F2psDict begin /$F2psEnteredState save def} def
915|+/$F2psEnd {$F2psEnteredState restore end} def
916|+
917|+/pageheader {
918|+sa
919|+n 0 369 m 0 0 1 409 0 1 409 369 1 cp clip
920|+0.7 372.4 tr
921|+1 -1 sc
922|+/$F2psBegin
923|+10 setmiterlimit
924|+0 slj 0 slc
925|+ 0.06000 0.06000 sc
926|+} bind def
927|+/pagefooter {

```

```

928 +$F2psEnd
929 +restore
930 +} bind def
931 +%EndProlog
932 +pageheader
933 +%
934 +% Fig objects follow
935 +%
936 +%
937 +% here starts figure with depth 50
938 +% Ellipse
939 +7.500 slw
940 +0 slc
941 +n 893 1007 413 112 0 360 DrawEllipse gs col0 s gr
942 +% Ellipse
943 +n 2018 1007 413 112 0 360 DrawEllipse gs col0 s gr
944 +/Courier-iso ff 66.67 scf sf
945 +1875 1050 m
946 +gs 1 -1 sc (Tipos) col0 sh gr
947 +% Ellipse
948 +n 3068 1307 413 112 0 360 DrawEllipse gs col0 s gr
949 +/Courier-iso ff 66.67 scf sf
950 +2775 1350 m
951 +gs 1 -1 sc (Game of Life) col0 sh gr
952 +% Ellipse
953 +n 487 1912 413 112 0 360 DrawEllipse gs col0 s gr
954 +/Courier-iso ff 66.67 scf sf
955 +189 1950 m
956 +gs 1 -1 sc (Escolha de cor) col0 sh gr
957 +% Ellipse
958 +n 1387 3167 487 133 0 360 DrawEllipse gs col0 s gr
959 +/Courier-iso ff 66.67 scf sf
960 +975 3198 m
961 +gs 1 -1 sc (Redes de Computadres) col0 sh gr
962 +% Ellipse
963 +n 3187 2887 413 112 0 360 DrawEllipse gs col0 s gr
964 +% Ellipse
965 +n 3207 2441 413 112 0 360 DrawEllipse gs col0 s gr
966 +/Courier-iso ff 66.67 scf sf
967 +3000 2475 m
968 +gs 1 -1 sc (Ocilators) col0 sh gr
969 +% Ellipse
970 +n 3221 1981 413 112 0 360 DrawEllipse gs col0 s gr
971 +/Courier-iso ff 66.67 scf sf
972 +3000 2025 m
973 +gs 1 -1 sc (Still Lives) col0 sh gr
974 +/Courier-iso ff 66.67 scf sf
975 +3000 2925 m
976 +gs 1 -1 sc (Space Ships) col0 sh gr
977 +/Courier-iso ff 66.67 scf sf
978 +3816 878 m
979 +gs 1 -1 sc (Te\363ria da Computa\347\343o) col0 sh gr
980 +% Ellipse
981 +n 4162 225 413 112 0 360 DrawEllipse gs col0 s gr
982 +/Courier-iso ff 66.67 scf sf
983 +3975 300 m
984 +gs 1 -1 sc (Matem\341tica) col0 sh gr
985 +/Courier-iso ff 66.67 scf sf
986 +3975 225 m
987 +gs 1 -1 sc (Complexidade) col0 sh gr
988 +/Courier-iso ff 66.67 scf sf
989 +4650 555 m
990 +gs 1 -1 sc (IA-Reconhecimento) col0 sh gr
991 +/Courier-iso ff 66.67 scf sf
992 +4725 630 m
993 +gs 1 -1 sc (de Par\365es) col0 sh gr
994 +% Ellipse
995 +n 4972 575 444 112 0 360 DrawEllipse gs col0 s gr
996 +% Ellipse
997 +n 5468 1157 413 112 0 360 DrawEllipse gs col0 s gr
998 +/Courier-iso ff 66.67 scf sf
999 +5400 1200 m
1000 +gs 1 -1 sc (Boat) col0 sh gr
1001 +/Courier-iso ff 66.67 scf sf
1002 +5700 1650 m
1003 +gs 1 -1 sc (Beehive) col0 sh gr
1004 +% Ellipse
1005 +n 5843 1607 413 112 0 360 DrawEllipse gs col0 s gr
1006 +% Ellipse
1007 +n 4718 1532 413 112 0 360 DrawEllipse gs col0 s gr
1008 +/Courier-iso ff 66.67 scf sf
1009 +4650 1575 m
1010 +gs 1 -1 sc (Block) col0 sh gr
1011 +% Ellipse
1012 +n 5243 1982 413 112 0 360 DrawEllipse gs col0 s gr
1013 +/Courier-iso ff 66.67 scf sf
1014 +5100 2025 m
1015 +gs 1 -1 sc (Load) col0 sh gr
1016 +% Ellipse
1017 +n 4793 2432 413 112 0 360 DrawEllipse gs col0 s gr
1018 +/Courier-iso ff 66.67 scf sf
1019 +4725 2475 m
1020 +gs 1 -1 sc (Tub) col0 sh gr
1021 +% Ellipse
1022 +n 4718 3632 413 112 0 360 DrawEllipse gs col0 s gr
1023 +/Courier-iso ff 66.67 scf sf
1024 +4413 3654 m
1025 +gs 1 -1 sc (Penta-decathlon) col0 sh gr
1026 +% Ellipse
1027 +n 6068 2807 413 112 0 360 DrawEllipse gs col0 s gr
1028 +/Courier-iso ff 66.67 scf sf
1029 +5925 2850 m
1030 +gs 1 -1 sc (Blinker) col0 sh gr
1031 +% Ellipse
1032 +n 4868 3182 413 112 0 360 DrawEllipse gs col0 s gr
1033 +/Courier-iso ff 66.67 scf sf
1034 +4800 3225 m
1035 +gs 1 -1 sc (Toad) col0 sh gr
1036 +/Courier-iso ff 66.67 scf sf
1037 +6300 3225 m
1038 +gs 1 -1 sc (Beacon) col0 sh gr
1039 +% Ellipse
1040 +n 6368 3182 413 112 0 360 DrawEllipse gs col0 s gr
1041 +/Courier-iso ff 66.67 scf sf
1042 +5700 3600 m
1043 +gs 1 -1 sc (Pulsar) col0 sh gr

```

```

1044 +% Ellipse
1045 +n 5843 3557 413 112 0 360 DrawEllipse gs col0 s gr
1046 +% Polyline
1047 +2 slj
1048 +n 3675 2475 m 3677 2476 1 3681 2477 1 3688 2479 1 3699 2483 1 3716 2489 1
1049 + 3737 2496 1 3764 2506 1 3797 2517 1 3834 2530 1 3877 2544 1
1050 + 3924 2560 1 3974 2577 1 4027 2596 1 4082 2614 1 4139 2633 1
1051 + 4196 2653 1 4253 2672 1 4309 2691 1 4365 2709 1 4418 2727 1
1052 + 4470 2744 1 4520 2761 1 4568 2777 1 4614 2792 1 4658 2806 1
1053 + 4700 2819 1 4740 2832 1 4778 2843 1 4814 2855 1 4849 2865 1
1054 + 4882 2875 1 4915 2885 1 4946 2893 1 4976 2902 1 5005 2910 1
1055 + 5034 2918 1 5063 2925 1 5098 2934 1 5132 2942 1 5167 2951 1
1056 + 5201 2958 1 5235 2966 1 5269 2973 1 5304 2979 1 5339 2986 1
1057 + 5376 2993 1 5414 2999 1 5453 3005 1 5493 3011 1 5535 3017 1
1058 + 5577 3023 1 5621 3029 1 5665 3035 1 5710 3041 1 5753 3046 1
1059 + 5795 3052 1 5835 3056 1 5872 3061 1 5904 3064 1 5932 3068 1
1060 + 5955 3070 1 5973 3072 1 5985 3073 1 5994 3074 1 5998 3075 1
1061 +
1062 + 6000 3075 1 gs col0 s gr % Ellipse
1063 +n 2693 3782 413 112 0 360 DrawEllipse gs col0 s gr
1064 +/Courier-iso ff 66.67 scf sf
1065 +2550 3825 m
1066 +gs 1 -1 sc (Glider) col0 sh gr
1067 +% Ellipse
1068 +n 2183 4709 608 166 0 360 DrawEllipse gs col0 s gr
1069 +/Courier-iso ff 66.67 scf sf
1070 +1725 4725 m
1071 +gs 1 -1 sc (Middle-weight spaceship) col0 sh gr
1072 +% Ellipse
1073 +n 3383 5159 608 166 0 360 DrawEllipse gs col0 s gr
1074 +/Courier-iso ff 66.67 scf sf
1075 +2925 5175 m
1076 +gs 1 -1 sc (Light-weight spaceship) col0 sh gr
1077 +% Ellipse
1078 +n 3329 4358 555 151 0 360 DrawEllipse gs col0 s gr
1079 +/Courier-iso ff 66.67 scf sf
1080 +2925 4380 m
1081 +gs 1 -1 sc (Heavy-weight spaceship) col0 sh gr
1082 +% Polyline
1083 +0 slj
1084 +n 105 5400 m 0 5400 0 6090 105 arcto 4 {pop} repeat
1085 + 0 6195 1995 6195 105 arcto 4 {pop} repeat
1086 + 2100 6195 2100 5505 105 arcto 4 {pop} repeat
1087 + 2100 5400 105 5400 105 arcto 4 {pop} repeat
1088 + cp gs col16 1.00 shd ef gr gs col0 s gr /Courier-iso ff 66.67 scf sf
1089 +75 5895 m
1090 +gs 1 -1 sc (de simula\347\343o de comportamento celular em) col0 sh gr
1091 +/Courier-iso ff 66.67 scf sf
1092 +75 5625 m
1093 +gs 1 -1 sc (Os termos em ingl\352s \351 como se ) col0 sh gr
1094 +/Courier-iso ff 66.67 scf sf
1095 +75 5715 m
1096 +gs 1 -1 sc (encontra na literatura representam o) col0 sh gr
1097 +/Courier-iso ff 66.67 scf sf
1098 +75 5805 m
1099 +gs 1 -1 sc (aut\364mato que deu origem a todo este estudo) col0 sh gr
1100 +/Courier-iso ff 66.67 scf sf
1101 +75 5985 m
1102 +gs 1 -1 sc (computador, Game of life, apesar da ideia) col0 sh gr
1103 +/Courier-iso ff 66.67 scf sf
1104 +75 6075 m
1105 +gs 1 -1 sc (ter surgido na matem\34itica ainda no s\351culo XIX) col0 sh gr
1106 +% Ellipse
1107 +n 1718 182 413 112 0 360 DrawEllipse gs col0 s gr
1108 +% Ellipse
1109 +n 937 637 413 112 0 360 DrawEllipse gs col0 s gr
1110 +% Ellipse
1111 +n 1943 1757 413 112 0 360 DrawEllipse gs col0 s gr
1112 +% Ellipse
1113 +n 2993 482 413 112 0 360 DrawEllipse gs col29 1.00 shd ef gr gs col27 s gr
1114 +% Ellipse
1115 +n 1462 2362 413 112 0 360 DrawEllipse gs col0 s gr
1116 +% Ellipse
1117 +n 1118 1457 413 112 0 360 DrawEllipse gs col0 s gr
1118 +% Ellipse
1119 +n 487 2662 413 112 0 360 DrawEllipse gs col0 s gr
1120 +% Ellipse
1121 +n 4193 857 444 112 0 360 DrawEllipse gs col0 s gr
1122 +% Polyline
1123 +gs clippath
1124 +2166 227 m 2173 216 1 2307 269 1 2276 321 1 cp
1125 +eoclip
1126 +n 2175 225 m
1127 + 2550 450 1 gs col0 s gr gr
1128 +% arrowhead
1129 +n 2307 269 m 2189 233 1 2276 321 1 2307 269 1 cp gs 0.00 setgray ef gr col0 s
1130 +% Polyline
1131 +gs clippath
1132 +3753 291 m 3759 303 1 3642 388 1 3615 334 1 cp
1133 +eoclip
1134 +n 3450 450 m
1135 + 3750 300 1 gs col0 s gr gr
1136 +% arrowhead
1137 +n 3642 388 m 3736 307 1 3615 334 1 3642 388 1 cp gs 0.00 setgray ef gr col0 s
1138 +% Polyline
1139 +2 slj
1140 +n 3675 1950 m
1141 + 4725 1950 1 gs col0 s gr % Polyline
1142 +gs clippath
1143 +2775 2915 m 2785 2925 1 2702 3044 1 2659 3003 1 cp
1144 +eoclip
1145 +n 2775 2925 m 2752 2949 1 2737 2965 1 2718 2985 1 2695 3009 1 2669 3038 1
1146 + 2639 3070 1 2608 3104 1 2575 3141 1 2541 3178 1 2507 3216 1
1147 + 2474 3254 1 2442 3291 1 2412 3327 1 2383 3361 1 2356 3395 1
1148 + 2331 3426 1 2308 3457 1 2287 3485 1 2268 3513 1 2250 3540 1
1149 + 2234 3566 1 2220 3591 1 2207 3615 1 2195 3639 1 2185 3663 1
1150 + 2175 3688 1 2166 3713 1 2157 3739 1 2150 3766 1 2144 3793 1
1151 + 2139 3821 1 2135 3850 1 2132 3880 1 2130 3912 1 2129 3946 1
1152 + 2128 3982 1 2129 4019 1 2130 4059 1 2133 4100 1 2135 4143 1
1153 + 2139 4186 1 2143 4230 1 2147 4274 1 2152 4316 1 2156 4355 1
1154 + 2161 4391 1 2164 4422 1 2168 4448 1 2171 4468 1 2173 4483 1
1155 + 2174 4492 1 2175 4498 1
1156 + 2175 4500 1 gs col0 s gr gr
1157 +% arrowhead
1158 +0 slj
1159 +n 2702 3044 m 2764 2937 1 2659 3003 1 2702 3044 1 cp gs 0.00 setgray ef gr col0 s

```

```

1160 +% Polyline
1161 +2 slj
1162 +gs clippath
1163 +2098 909 m 2091 898 l 2193 796 l 2228 845 l cp
1164 +eoclip
1165 +n 2625 525 m
1166 + 2100 900 l gs col0 s gr gr
1167 +% arrowhead
1168 +0 slj
1169 +n 2193 796 m 2113 891 l 2228 845 l 2193 796 l cp gs 0.00 setgray ef gr col0 s
1170 +% Polyline
1171 +2 slj
1172 +gs clippath
1173 +1734 898 m 1727 909 l 1593 856 l 1624 804 l cp
1174 +eoclip
1175 +n 1725 900 m
1176 + 1350 675 l gs col0 s gr gr
1177 +% arrowhead
1178 +0 slj
1179 +n 1593 856 m 1711 892 l 1624 804 l 1593 856 l cp gs 0.00 setgray ef gr col0 s
1180 +% Polyline
1181 +2 slj
1182 +gs clippath
1183 +1582 968 m 1582 982 l 1439 1005 l 1439 945 l cp
1184 +eoclip
1185 +n 1575 975 m
1186 + 1350 975 l gs col0 s gr gr
1187 +% arrowhead
1188 +0 slj
1189 +n 1439 1005 m 1559 975 l 1439 945 l 1439 1005 l cp gs 0.00 setgray ef gr col0 s
1190 +% Polyline
1191 +2 slj
1192 +gs clippath
1193 +1575 1040 m 1585 1050 l 1500 1168 l 1457 1125 l cp
1194 +eoclip
1195 +n 1575 1050 m
1196 + 1275 1350 l gs col0 s gr gr
1197 +% arrowhead
1198 +0 slj
1199 +n 1500 1168 m 1564 1061 l 1457 1125 l 1500 1168 l cp gs 0.00 setgray ef gr col0 s
1200 +% Polyline
1201 +2 slj
1202 +gs clippath
1203 +1943 1118 m 1957 1118 l 1980 1261 l 1920 1261 l cp
1204 +eoclip
1205 +n 1950 1125 m
1206 + 1950 1650 l gs col0 s gr gr
1207 +% arrowhead
1208 +0 slj
1209 +n 1980 1261 m 1950 1141 l 1920 1261 l 1980 1261 l cp gs 0.00 setgray ef gr col0 s
1210 +% Polyline
1211 +2 slj
1212 +gs clippath
1213 +2992 594 m 3006 592 l 3049 731 l 2990 739 l cp
1214 +eoclip
1215 +n 3000 600 m
1216 + 3075 1125 l gs col0 s gr gr
1217 +% arrowhead
1218 +0 slj
1219 +n 3049 731 m 3002 616 l 2990 739 l 3049 731 l cp gs 0.00 setgray ef gr col0 s
1220 +% Polyline
1221 +2 slj
1222 +gs clippath
1223 +4507 594 m 4506 607 l 4362 620 l 4366 560 l cp
1224 +eoclip
1225 +n 3450 525 m
1226 + 4500 600 l gs col0 s gr gr
1227 +% arrowhead
1228 +0 slj
1229 +n 4362 620 m 4484 599 l 4366 560 l 4362 620 l cp gs 0.00 setgray ef gr col0 s
1230 +% Polyline
1231 +2 slj
1232 +gs clippath
1233 +3759 823 m 3752 834 l 3618 781 l 3649 729 l cp
1234 +eoclip
1235 +n 3375 600 m
1236 + 3750 825 l gs col0 s gr gr
1237 +% arrowhead
1238 +0 slj
1239 +n 3618 781 m 3736 817 l 3649 729 l 3618 781 l cp gs 0.00 setgray ef gr col0 s
1240 +% Polyline
1241 +2 slj
1242 +gs clippath
1243 +3525 1885 m 3515 1875 l 3602 1759 l 3644 1802 l cp
1244 +eoclip
1245 +n 3525 1875 m 3552 1849 l 3568 1833 l 3588 1814 l 3612 1791 l 3638 1767 l
1246 + 3666 1740 l 3695 1714 l 3723 1697 l 3751 1662 l 3778 1638 l
1247 + 3803 1615 l 3828 1595 l 3850 1575 l 3872 1557 l 3893 1541 l
1248 + 3912 1526 l 3932 1512 l 3950 1499 l 3969 1487 l 3988 1475 l
1249 + 4008 1463 l 4029 1451 l 4050 1440 l 4072 1428 l 4094 1418 l
1250 + 4117 1407 l 4140 1397 l 4165 1387 l 4189 1377 l 4214 1368 l
1251 + 4239 1359 l 4264 1350 l 4290 1342 l 4315 1334 l 4339 1327 l
1252 + 4364 1320 l 4387 1314 l 4411 1308 l 4434 1302 l 4456 1297 l
1253 + 4478 1292 l 4500 1288 l 4522 1283 l 4544 1278 l 4567 1274 l
1254 + 4590 1269 l 4614 1265 l 4640 1260 l 4668 1256 l 4697 1251 l
1255 + 4728 1246 l 4761 1240 l 4796 1235 l 4832 1229 l 4868 1224 l
1256 + 4902 1218 l 4934 1214 l 4963 1209 l 4986 1206 l 5004 1203 l
1257 + 5015 1201 l 5022 1200 l
1258 + 5025 1200 l gs col0 s gr gr
1259 +% arrowhead
1260 +0 slj
1261 +n 3602 1759 m 3537 1864 l 3644 1802 l 3602 1759 l cp gs 0.00 setgray ef gr col0 s
1262 +% Polyline
1263 +2 slj
1264 +gs clippath
1265 +3671 1959 m 3666 1946 l 3790 1872 l 3813 1927 l cp
1266 +eoclip
1267 +n 3675 1950 m
1268 + 4425 1650 l gs col0 s gr gr
1269 +% arrowhead
1270 +0 slj
1271 +n 3790 1872 m 3690 1944 l 3813 1927 l 3790 1872 l cp gs 0.00 setgray ef gr col0 s
1272 +% Polyline
1273 +2 slj
1274 +gs clippath
1275 +3668 2031 m 3669 2018 l 3814 2009 l 3807 2069 l cp

```

```

1276 +eoclip
1277 +n 3675 2025 m 3704 2028 1 3723 2031 1 3748 2034 1 3780 2038 1 3818 2042 1
1278 + 3863 2048 1 3913 2054 1 3968 2060 1 4027 2067 1 4089 2074 1
1279 + 4153 2082 1 4219 2090 1 4285 2097 1 4351 2105 1 4416 2112 1
1280 + 4480 2119 1 4541 2126 1 4601 2133 1 4657 2139 1 4711 2145 1
1281 + 4763 2150 1 4811 2155 1 4857 2160 1 4901 2164 1 4942 2168 1
1282 + 4980 2171 1 5017 2175 1 5051 2177 1 5084 2180 1 5115 2182 1
1283 + 5144 2184 1 5172 2185 1 5199 2186 1 5225 2188 1 5264 2189 1
1284 + 5301 2189 1 5336 2189 1 5370 2189 1 5402 2187 1 5433 2186 1
1285 + 5462 2183 1 5489 2180 1 5516 2177 1 5540 2173 1 5564 2168 1
1286 + 5585 2163 1 5605 2157 1 5624 2151 1 5641 2145 1 5656 2138 1
1287 + 5670 2131 1 5683 2123 1 5694 2116 1 5704 2108 1 5714 2100 1
1288 + 5722 2092 1 5730 2083 1 5738 2075 1 5748 2062 1 5758 2048 1
1289 + 5767 2033 1 5775 2016 1 5784 1997 1 5793 1976 1 5802 1952 1
1290 + 5811 1926 1 5820 1900 1 5828 1873 1 5836 1848 1 5842 1828 1
1291 + 5846 1813 1 5849 1804 1 5850 1801 1
1292 + 5850 1800 1 gs col0 s gr gr
1293 +% arrowhead
1294 +0 slj
1295 +n 3814 2009 m 3691 2027 1 3807 2069 1 3814 2009 1 cp gs 0.00 setgray ef gr col0 s
1296 +% Polyline
1297 +2 slj
1298 +gs clippath
1299 +3666 2029 m 3671 2016 1 3813 2048 1 3790 2103 1 cp
1300 +eoclip
1301 +n 3675 2025 m
1302 + 4425 2325 1 gs col0 s gr gr
1303 +% arrowhead
1304 +0 slj
1305 +n 3813 2048 m 3690 2031 1 3790 2103 1 3813 2048 1 cp gs 0.00 setgray ef gr col0 s
1306 +% Polyline
1307 +2 slj
1308 +gs clippath
1309 +3667 2480 m 3670 2467 1 3814 2472 1 3803 2531 1 cp
1310 +eoclip
1311 +n 3675 2475 m 3705 2481 1 3724 2485 1 3750 2490 1 3781 2496 1 3817 2503 1
1312 + 3859 2511 1 3904 2520 1 3952 2529 1 4002 2539 1 4054 2549 1
1313 + 4106 2559 1 4157 2569 1 4207 2578 1 4255 2588 1 4302 2596 1
1314 + 4347 2605 1 4389 2613 1 4429 2620 1 4468 2627 1 4504 2634 1
1315 + 4539 2640 1 4572 2646 1 4604 2651 1 4635 2656 1 4664 2661 1
1316 + 4693 2666 1 4722 2671 1 4750 2675 1 4782 2680 1 4813 2684 1
1317 + 4845 2689 1 4877 2693 1 4909 2698 1 4942 2702 1 4976 2706 1
1318 + 5011 2711 1 5048 2715 1 5087 2719 1 5127 2724 1 5169 2729 1
1319 + 5212 2733 1 5257 2738 1 5302 2743 1 5348 2747 1 5393 2752 1
1320 + 5436 2756 1 5477 2760 1 5513 2764 1 5545 2767 1 5572 2770 1
1321 + 5593 2772 1 5608 2773 1 5617 2774 1 5623 2775 1
1322 + 5625 2775 1 gs col0 s gr gr
1323 +% arrowhead
1324 +0 slj
1325 +n 3814 2472 m 3691 2478 1 3803 2531 1 3814 2472 1 cp gs 0.00 setgray ef gr col0 s
1326 +% Polyline
1327 +2 slj
1328 +gs clippath
1329 +3591 2478 m 3597 2466 1 3734 2513 1 3706 2566 1 cp
1330 +eoclip
1331 +n 3600 2475 m
1332 + 4725 3075 1 gs col0 s gr gr
1333 +% arrowhead
1334 +0 slj
1335 +n 3734 2513 m 3614 2483 1 3706 2566 1 3734 2513 1 cp gs 0.00 setgray ef gr col0 s
1336 +% Polyline
1337 +2 slj
1338 +gs clippath
1339 +3515 2550 m 3525 2540 1 3643 2625 1 3600 2668 1 cp
1340 +eoclip
1341 +n 3525 2550 m
1342 + 4500 3525 1 gs col0 s gr gr
1343 +% arrowhead
1344 +0 slj
1345 +n 3643 2625 m 3536 2561 1 3600 2668 1 3643 2625 1 cp gs 0.00 setgray ef gr col0 s
1346 +% Polyline
1347 +2 slj
1348 +gs clippath
1349 +3516 2546 m 3529 2541 1 3599 2668 1 3543 2688 1 cp
1350 +eoclip
1351 +n 3525 2550 m 3535 2578 1 3542 2596 1 3550 2620 1 3561 2650 1 3573 2685 1
1352 + 3588 2725 1 3604 2769 1 3621 2817 1 3639 2867 1 3658 2919 1
1353 + 3677 2972 1 3696 3024 1 3715 3076 1 3733 3126 1 3751 3174 1
1354 + 3768 3221 1 3784 3265 1 3800 3306 1 3815 3346 1 3829 3383 1
1355 + 3842 3417 1 3855 3450 1 3867 3480 1 3878 3509 1 3889 3536 1
1356 + 3899 3561 1 3909 3585 1 3919 3608 1 3928 3629 1 3938 3650 1
1357 + 3951 3680 1 3965 3709 1 3978 3736 1 3992 3762 1 4005 3787 1
1358 + 4019 3811 1 4032 3833 1 4046 3855 1 4060 3875 1 4074 3894 1
1359 + 4088 3912 1 4101 3928 1 4115 3943 1 4129 3957 1 4143 3970 1
1360 + 4156 3982 1 4170 3993 1 4183 4003 1 4197 4013 1 4210 4021 1
1361 + 4224 4030 1 4238 4038 1 4253 4046 1 4270 4054 1 4287 4063 1
1362 + 4305 4071 1 4324 4079 1 4344 4087 1 4365 4095 1 4387 4103 1
1363 + 4409 4111 1 4432 4118 1 4455 4125 1 4479 4131 1 4502 4137 1
1364 + 4526 4142 1 4549 4147 1 4573 4151 1 4595 4155 1 4618 4158 1
1365 + 4640 4161 1 4663 4163 1 4683 4164 1 4704 4165 1 4725 4165 1
1366 + 4747 4165 1 4769 4165 1 4792 4164 1 4815 4162 1 4839 4160 1
1367 + 4863 4157 1 4888 4153 1 4913 4149 1 4937 4144 1 4962 4138 1
1368 + 4986 4132 1 5010 4125 1 5033 4118 1 5056 4110 1 5078 4101 1
1369 + 5100 4093 1 5121 4083 1 5142 4073 1 5163 4063 1 5181 4052 1
1370 + 5200 4041 1 5218 4029 1 5238 4016 1 5257 4002 1 5278 3987 1
1371 + 5300 3970 1 5322 3951 1 5347 3931 1 5372 3909 1 5399 3885 1
1372 + 5427 3860 1 5455 3834 1 5484 3808 1 5512 3782 1 5538 3758 1
1373 + 5562 3736 1 5582 3716 1 5598 3701 1 5610 3689 1 5619 3681 1
1374 + 5623 3677 1
1375 + 5625 3675 1 gs col0 s gr gr
1376 +% arrowhead
1377 +0 slj
1378 +n 3599 2668 m 3530 2565 1 3543 2688 1 3599 2668 1 cp gs 0.00 setgray ef gr col0 s
1379 +% Polyline
1380 +2 slj
1381 +gs clippath
1382 +3441 2997 m 3453 2991 1 3538 3108 1 3484 3135 1 cp
1383 +eoclip
1384 +n 3450 3000 m 3462 3024 1 3469 3040 1 3480 3062 1 3492 3088 1 3508 3121 1
1385 + 3525 3158 1 3545 3200 1 3567 3246 1 3591 3296 1 3615 3348 1
1386 + 3640 3403 1 3666 3458 1 3692 3515 1 3717 3571 1 3742 3626 1
1387 + 3767 3680 1 3790 3732 1 3812 3783 1 3833 3831 1 3852 3878 1
1388 + 3871 3922 1 3887 3964 1 3903 4004 1 3918 4042 1 3931 4078 1
1389 + 3943 4112 1 3953 4145 1 3963 4176 1 3971 4205 1 3979 4234 1
1390 + 3986 4261 1 3991 4287 1 3996 4313 1 4000 4338 1 4004 4373 1
1391 + 4007 4406 1 4008 4439 1 4007 4471 1 4005 4503 1 4000 4535 1

```

```

1392 + 3993 4566 1 3984 4598 1 3974 4631 1 3961 4664 1 3946 4698 1
1393 + 3930 4733 1 3912 4768 1 3892 4804 1 3872 4839 1 3852 4873 1
1394 + 3831 4905 1 3813 4934 1 3795 4960 1 3781 4981 1 3769 4998 1
1395 + 3760 5011 1 3755 5019 1 3751 5023 1
1396 + 3750 5025 1 gs col0 s gr gr
1397 +% arrowhead
1398 +0 slj
1399 +n 3538 3108 m 3457 3014 1 3484 3135 1 3538 3108 1 cp gs 0.00 setgray ef gr col0 s
1400 +% Polyline
1401 +2 slj
1402 +gs clippath
1403 +3143 2994 m 3156 2993 1 3188 3134 1 3129 3138 1 cp
1404 +eoclip
1405 +n 3150 3000 m
1406 + 3225 4200 1 gs col0 s gr gr
1407 +% arrowhead
1408 +0 slj
1409 +n 3188 3134 m 3151 3016 1 3129 3138 1 3188 3134 1 cp gs 0.00 setgray ef gr col0 s
1410 +% Polyline
1411 +2 slj
1412 +gs clippath
1413 +2920 2992 m 2933 2995 1 2925 3139 1 2866 3126 1 cp
1414 +eoclip
1415 +n 2925 3000 m
1416 + 2775 3675 1 gs col0 s gr gr
1417 +% arrowhead
1418 +0 slj
1419 +n 2925 3139 m 2921 3016 1 2866 3126 1 2925 3139 1 cp gs 0.00 setgray ef gr col0 s
1420 +% Polyline
1421 +2 slj
1422 +gs clippath
1423 +2859 2778 m 2847 2784 1 2762 2667 1 2816 2640 1 cp
1424 +eoclip
1425 +n 2775 1425 m 2774 1427 1 2773 1432 1 2771 1442 1 2767 1457 1 2762 1477 1
1426 + 2756 1502 1 2748 1533 1 2740 1568 1 2731 1607 1 2721 1648 1
1427 + 2711 1690 1 2701 1733 1 2692 1776 1 2683 1817 1 2675 1857 1
1428 + 2667 1895 1 2661 1931 1 2655 1965 1 2650 1997 1 2646 2027 1
1429 + 2642 2055 1 2640 2082 1 2638 2107 1 2637 2131 1 2636 2155 1
1430 + 2637 2178 1 2638 2200 1 2639 2226 1 2642 2251 1 2646 2276 1
1431 + 2651 2301 1 2657 2327 1 2664 2353 1 2672 2381 1 2682 2410 1
1432 + 2693 2440 1 2706 2471 1 2720 2504 1 2734 2538 1 2750 2572 1
1433 + 2766 2606 1 2782 2640 1 2797 2671 1 2811 2700 1 2824 2724 1
1434 + 2834 2743 1
1435 + 2850 2775 1 gs col0 s gr gr
1436 +% arrowhead
1437 +0 slj
1438 +n 2762 2667 m 2843 2761 1 2816 2640 1 2762 2667 1 cp gs 0.00 setgray ef gr col0 s
1439 +% Polyline
1440 +2 slj
1441 +gs clippath
1442 +2859 2329 m 2846 2334 1 2776 2207 1 2832 2187 1 cp
1443 +eoclip
1444 +n 2925 1425 m 2924 1428 1 2921 1434 1 2916 1446 1 2909 1463 1 2900 1486 1
1445 + 2889 1514 1 2877 1546 1 2864 1580 1 2851 1615 1 2838 1649 1
1446 + 2825 1683 1 2814 1715 1 2804 1745 1 2795 1773 1 2787 1799 1
1447 + 2781 1823 1 2775 1845 1 2771 1867 1 2767 1887 1 2764 1906 1
1448 + 2763 1925 1 2761 1945 1 2761 1966 1 2761 1986 1 2763 2007 1
1449 + 2766 2029 1 2770 2052 1 2774 2076 1 2780 2102 1 2787 2129 1
1450 + 2795 2157 1 2804 2186 1 2813 2215 1 2822 2243 1 2830 2268 1
1451 + 2837 2289 1
1452 + 2850 2325 1 gs col0 s gr gr
1453 +% arrowhead
1454 +0 slj
1455 +n 2776 2207 m 2845 2310 1 2832 2187 1 2776 2207 1 cp gs 0.00 setgray ef gr col0 s
1456 +% Polyline
1457 +2 slj
1458 +gs clippath
1459 +3233 1881 m 3219 1883 1 3173 1746 1 3232 1736 1 cp
1460 +eoclip
1461 +n 3150 1425 m
1462 + 3225 1875 1 gs col0 s gr gr
1463 +% arrowhead
1464 +0 slj
1465 +n 3173 1746 m 3222 1859 1 3232 1736 1 3173 1746 1 cp gs 0.00 setgray ef gr col0 s
1466 +% Polyline
1467 +2 slj
1468 +gs clippath
1469 +1872 2334 m 1866 2322 1 1979 2232 1 2009 2285 1 cp
1470 +eoclip
1471 +n 2850 600 m 2849 602 1 2846 608 1 2841 617 1 2833 632 1 2822 652 1
1472 + 2809 678 1 2793 708 1 2775 742 1 2756 779 1 2737 818 1
1473 + 2717 857 1 2697 896 1 2679 935 1 2661 972 1 2644 1007 1
1474 + 2628 1041 1 2614 1073 1 2601 1103 1 2589 1132 1 2579 1159 1
1475 + 2569 1186 1 2560 1212 1 2552 1237 1 2544 1262 1 2538 1288 1
1476 + 2531 1313 1 2525 1338 1 2520 1364 1 2514 1390 1 2509 1417 1
1477 + 2504 1444 1 2499 1471 1 2494 1499 1 2490 1527 1 2485 1556 1
1478 + 2480 1584 1 2475 1613 1 2470 1641 1 2465 1669 1 2459 1697 1
1479 + 2453 1724 1 2447 1751 1 2441 1776 1 2434 1801 1 2427 1825 1
1480 + 2420 1848 1 2412 1870 1 2404 1891 1 2395 1912 1 2385 1931 1
1481 + 2375 1950 1 2363 1970 1 2350 1989 1 2336 2007 1 2320 2026 1
1482 + 2302 2044 1 2283 2062 1 2261 2081 1 2237 2100 1 2211 2119 1
1483 + 2183 2139 1 2152 2160 1 2119 2181 1 2085 2203 1 2051 2224 1
1484 + 2017 2244 1 1985 2263 1 1955 2280 1 1930 2295 1 1909 2306 1
1485 +
1486 + 1875 2325 1 gs col0 s gr gr
1487 +% arrowhead
1488 +0 slj
1489 +n 1979 2232 m 1889 2317 1 2009 2285 1 1979 2232 1 cp gs 0.00 setgray ef gr col0 s
1490 +% Polyline
1491 +2 slj
1492 +gs clippath
1493 +1134 2248 m 1127 2259 1 993 2206 1 1024 2154 1 cp
1494 +eoclip
1495 +n 1125 2250 m
1496 + 750 2025 1 gs col0 s gr gr
1497 +% arrowhead
1498 +0 slj
1499 +n 993 2206 m 1111 2242 1 1024 2154 1 993 2206 1 cp gs 0.00 setgray ef gr col0 s
1500 +% Polyline
1501 +2 slj
1502 +gs clippath
1503 +1054 2391 m 1059 2404 1 930 2472 1 911 2415 1 cp
1504 +eoclip
1505 +n 1050 2400 m
1506 + 600 2550 1 gs col0 s gr gr
1507 +% arrowhead

```

```

1508 +0 slj
1509 +n 930 2472 m 1035 2405 l 911 2415 l 930 2472 l cp gs 0.00 setgray ef gr col0 s
1510 +% Polyline
1511 +2 slj
1512 +gs clippath
1513 +1418 2468 m 1432 2468 l 1455 2611 l 1395 2611 l cp
1514 +eoclip
1515 +n 1425 2475 m
1516 + 1425 3000 l gs col0 s gr gr
1517 +% arrowhead
1518 +0 slj
1519 +n 1455 2611 m 1425 2491 l 1395 2611 l 1455 2611 l cp gs 0.00 setgray ef gr col0 s
1520 +/Courier-iso ff 66.67 scf sf
1521 +1500 225 m
1522 +gs 1 -1 sc (Alger\355time) col0 sh gr
1523 +/Courier-iso ff 66.67 scf sf
1524 +600 675 m
1525 +gs 1 -1 sc (C\351lula Triangular) col0 sh gr
1526 +/Courier-iso ff 66.67 scf sf
1527 +525 1050 m
1528 +gs 1 -1 sc (C\351lula Pentagonal) col0 sh gr
1529 +/Courier-iso ff 66.67 scf sf
1530 +1125 2400 m
1531 +gs 1 -1 sc (\301rea de aplica\347\343o) col0 sh gr
1532 +/Courier-iso ff 66.67 scf sf
1533 +792 1476 m
1534 +gs 1 -1 sc (C\351lula Hexagonal) col0 sh gr
1535 +/Courier-iso ff 66.67 scf sf
1536 +1650 1776 m
1537 +gs 1 -1 sc (C\351lula Quadrada) col0 sh gr
1538 +/Courier-iso ff 66.67 scf sf
1539 +132 2688 m
1540 +gs 1 -1 sc (Teclado de Celular) col0 sh gr
1541 +/Courier-iso ff 66.67 scf sf
1542 +2688 504 m
1543 +gs 1 -1 sc (Aut\364mato Celular) col0 sh gr
1544 +/Courier-iso ff 66.67 scf sf
1545 +3600 450 m
1546 +gs 1 -1 sc (Est\341 relacionado) col0 sh gr
1547 +/Courier-iso ff 66.67 scf sf
1548 +3675 600 m
1549 +gs 1 -1 sc (Est\341 relacionado) col0 sh gr
1550 +/Courier-iso ff 66.67 scf sf
1551 +3225 750 m
1552 +gs 1 -1 sc (Est\341 relacionado) col0 sh gr
1553 +/Courier-iso ff 66.67 scf sf
1554 +3075 975 m
1555 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1556 +/Courier-iso ff 66.67 scf sf
1557 +2100 675 m
1558 +gs 1 -1 sc (Possui) col0 sh gr
1559 +/Courier-iso ff 66.67 scf sf
1560 +2400 300 m
1561 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1562 +/Courier-iso ff 66.67 scf sf
1563 +3225 1650 m
1564 +gs 1 -1 sc (Possui) col0 sh gr
1565 +/Courier-iso ff 66.67 scf sf
1566 +2850 1725 m
1567 +gs 1 -1 sc (Possui) col0 sh gr
1568 +/Courier-iso ff 66.67 scf sf
1569 +2400 2475 m
1570 +gs 1 -1 sc (Possui) col0 sh gr
1571 +/Courier-iso ff 66.67 scf sf
1572 +4050 1275 m
1573 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1574 +/Courier-iso ff 66.67 scf sf
1575 +3975 1650 m
1576 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1577 +/Courier-iso ff 66.67 scf sf
1578 +5850 2025 m
1579 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1580 +/Courier-iso ff 66.67 scf sf
1581 +4275 2025 m
1582 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1583 +/Courier-iso ff 66.67 scf sf
1584 +3900 2325 m
1585 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1586 +/Courier-iso ff 66.67 scf sf
1587 +4875 2775 m
1588 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1589 +/Courier-iso ff 66.67 scf sf
1590 +4800 3000 m
1591 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1592 +/Courier-iso ff 66.67 scf sf
1593 +4275 3000 m
1594 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1595 +/Courier-iso ff 66.67 scf sf
1596 +3900 3225 m
1597 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1598 +/Courier-iso ff 66.67 scf sf
1599 +5475 3900 m
1600 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1601 +/Courier-iso ff 66.67 scf sf
1602 +3150 3375 m
1603 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1604 +/Courier-iso ff 66.67 scf sf
1605 +2550 3450 m
1606 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1607 +/Courier-iso ff 66.67 scf sf
1608 +4050 4575 m
1609 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1610 +/Courier-iso ff 66.67 scf sf
1611 +2175 4275 m
1612 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1613 +/Courier-iso ff 66.67 scf sf
1614 +2025 1425 m
1615 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1616 +/Courier-iso ff 66.67 scf sf
1617 +1425 1275 m
1618 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1619 +/Courier-iso ff 66.67 scf sf
1620 +1275 900 m
1621 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1622 +/Courier-iso ff 66.67 scf sf
1623 +1500 750 m

```



```

1624 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1625 +/Courier-iso ff 66.67 scf sf
1626 +1950 2100 m
1627 +gs 1 -1 sc (Possui) col0 sh gr
1628 +/Courier-iso ff 66.67 scf sf
1629 +675 2175 m
1630 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1631 +/Courier-iso ff 66.67 scf sf
1632 +825 2475 m
1633 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1634 +/Courier-iso ff 66.67 scf sf
1635 +1350 2775 m
1636 +gs 1 -1 sc (\311 um\(\a\)) col0 sh gr
1637 +% here ends figure;
1638 +pagefooter
1639 +showpage
1640 +%Trailer
1641 +%EOF
1642 diff --git a/doc/tex/fig/t.jpg b/doc/tex/fig/t.jpg
1643 new file mode 100644
1644 index 0000000..9c7f0e4
1645 Binary files /dev/null and b/doc/tex/fig/t.jpg differ
1646 diff --git a/doc/tex/mybib.bib b/doc/tex/mybib.bib
1647 new file mode 100644
1648 index 0000000..124a6bf
1649 --- /dev/null
1650 +++ b/doc/tex/mybib.bib
1651 @@ -0,0 +1,23 @@
1652 +@book{book1,
1653 +  title = {Computação Gráfica: Imagem},
1654 +  author = {Jonas Gomes, Luiz Velho},
1655 +  publisher = {SBM},
1656 +  isbn = {85-244-0088-9},
1657 +  year = {1994},
1658 +  series = {},
1659 +  edition = {},
1660 +  volume = {},
1661 +  url = {}
1662 +}
1663 +@book{book2,
1664 +  title = {A Computational Introduction to Digital Image Processing},
1665 +  author = {Alasdair McAndrew},
1666 +  publisher = {Taylor and Francis},
1667 +  isbn = {978-1-4822-4735-0},
1668 +  year = {2016},
1669 +  series = {},
1670 +  edition = {Second},
1671 +  volume = {},
1672 +  url = {}
1673 +}
1674 +
1675 diff --git a/doc/tex/packages.tex b/doc/tex/packages.tex
1676 new file mode 100644
1677 index 0000000..1351cce
1678 --- /dev/null
1679 +++ b/doc/tex/packages.tex
1680 @@ -0,0 +1,127 @@
1681 +\usepackage{syntax}
1682 +\usepackage[brazilian]{babel}
1683 +\usepackage[utf8]{inputenc}
1684 +% \usepackage[usenames, dvipsnames]{xcolor}
1685 +% \usepackage[usenames, dvipsnames]{color}
1686 +% \usepackage{tgterms} times homan font similar
1687 +\usepackage{hyperref}
1688 +\usepackage{indentfirst}
1689 +\usepackage{bbding}
1690 +\usepackage{pifont}
1691 +\usepackage{makeidx}
1692 +\makeindex
1693 +\hypersetup{
1694 +  colorlinks,
1695 +  citecolor=black,
1696 +  filecolor=black,
1697 +  linkcolor=black,
1698 +  urlcolor=black
1699 +}
1700 +\usepackage{fullpage}
1701 +\usepackage{amssymb}
1702 +\usepackage{float}
1703 +\usepackage[toc,page]{appendix}
1704 +\usepackage{cite}
1705 +% \usepackage{draftwatermark}
1706 +% \SetWatermarkText{RASCUNHO}
1707 +% \SetWatermarkScale{1.0}
1708 +% \SetWatermarkColor[rgb]{0.9, 0.9, 0.9}
1709 +
1710 +% \setdefaultlanguage[babelshorthands]{brazilian}
1711 +% \usepackage{fontspec}
1712 +% Pra mostra código fonte
1713 +\usepackage{listings}
1714 +\usepackage{caption}
1715 +\usepackage[most]{tcolorbox}
1716 +\DeclareCaptionFont{white}{\color{white}}
1717 +\DeclareCaptionFormat{listing}{%
1718 +  \parbox{\textwidth}{\colorbox{gray}{\parbox{\textwidth}{\#1\#2\#3}}\vskip-4pt}}
1719 +\captionsetup[lstlisting]{format=listing,labelfont=white,textfont=white}
1720 +\lstset{frame=lrb,xleftmargin=\fboxsep,xrightmargin=-\fboxsep}
1721 +
1722 +%%% Definição da sintaxe da linguagem jSmall para o hiligh
1723 +\lstdefinlanguage{jSmall}{
1724 +  keywords={numfi, numf, true, false, loop, return, if, in, while, then, else, elif},
1725 +  keywordstyle=\color{blue}\bfseries,
1726 +  ndkeywords={boot, export, boolean, throw, implements, import, this},
1727 +  ndkeywordstyle=\color{darkgray}\bfseries,
1728 +  identifierstyle=\color{black},
1729 +  sensitive=false,
1730 +  comment=[\{]-,
1731 +  morecomment=[s]{-\{ \}-},
1732 +  commentstyle=\color{purple}\ttfamily,
1733 +  stringstyle=\color{red}\ttfamily,
1734 +  morestring=[b]',
1735 +  morestring=[b]"
1736 +}
1737 +
1738 +\lstset{language=C++,
1739 +  basicstyle=\ttfamily,

```



```

1740 + keywordstyle=\color{blue}\ttfamily,
1741 + stringstyle=\color{red}\ttfamily,
1742 + commentstyle=\color{green}\ttfamily,
1743 + morecomment=[1][\color{magenta}]{\#}
1744 +}
1745 +
1746 +\newtcblisting[auto counter]{sexylisting}[2][{}]{sharp corners,
1747 + fonttitle=\bfseries, colframe=gray, listing only,
1748 + listing options={basicstyle=\ttfamily,language=jSmall, numbers=left},
1749 + title=Listing \thetcbcounter: #2, #1}
1750 +
1751 +\newtcblisting[auto counter]{sexylistingjava}[2][{}]{sharp corners,
1752 + fonttitle=\bfseries, colframe=gray, listing only,
1753 + listing options={basicstyle=\ttfamily,language=java, numbers=left},
1754 + title=Listing \thetcbcounter: #2, #1}
1755 +
1756 +\newtcblisting[auto counter]{sexylistingcpp}[2][{}]{sharp corners,
1757 + fonttitle=\bfseries, colframe=gray, listing only,
1758 + listing options={basicstyle=\ttfamily,language=C++, numbers=left},
1759 + title=Listing \thetcbcounter: #2, #1}
1760 +
1761 +% Automata packages
1762 +\usepackage{tikz, graphics}
1763 +
1764 +\usepackage{pgfplots}
1765 +\pgfplotsset{compat=1.16}
1766 +\usetikzlibrary{tikzmark, shapes.callouts}
1767 +\usetikzlibrary{automata, positioning, arrows}
1768 +\usetikzlibrary{arrows.meta, % if the figure contains arrow-tips
1769 + bending, % arrow tips on arcs are "bent," i.e., deformed a bit
1770 + patterns % if the figure contains pattern fills
1771 +}
1772 +
1773 +\usepackage{tipa}
1774 +
1775 +% Hacking pra poder usar syntax package junto com o tikz
1776 +\usepackage{etoolbox}
1777 +\AtBeginEnvironment{tikzpicture}{\catcode'\_ =8}
1778 +
1779 +\usepackage{ifthen,xcolor,xkeyval,calc}
1780 +\newlength{\tabcont}
1781 +
1782 +\newcommand{\tab}[1]{%
1783 +\settowidth{\tabcont}{\#1}%
1784 +\ifthenelse{\lengthtest{\tabcont < .25\linewidth}}{%
1785 +\makebox[.25\linewidth][l]{\#1}\ignorespaces}%
1786 +\makebox[.5\linewidth][l]{\#1}\ignorespaces}%
1787 +}%
1788 +
1789 +\frenchspacing
1790 +
1791 +% \title{
1792 +% \textbf{\Large UNIVERSIDADE FEDERAL DE ALAGOAS}\\
1793 +% \textbf{\Large INSTITUTO DE COMPUTAÇÃO}\\
1794 +% \{ \}\\
1795 +% \textbf{\Large Slam Combinando}\\
1796 +% \textbf{\Large com Filtros de Kalman}\\
1797 +% \textbf{\Large para Robôs Móveis}\\
1798 +% % \line(1,0){250} \\
1799 +% }
1800 +\author{Joilnen Leite\ 2017.2}
1801 +
1802 +% \usepackage{colorinlistoftodos}{todonotes}\setlength{\marginparwidth}{3cm}\reversemarginpar
1803 +\usepackage{todonotes}\setlength{\marginparwidth}{3cm}\reversemarginpar
1804 +
1805 +% HACK: set length so that the paper can have better width for margin
1806 +
1807 +\usepackage{float}
1808 +diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
1809 +new file mode 100644
1810 +index 0000000..c748d51
1811 +Binary files /dev/null and b/doc/tex/relatorio.pdf differ
1812 +diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
1813 +new file mode 100644
1814 +index 0000000..c79ccb2
1815 +--- /dev/null
1816 ++++ b/doc/tex/relatorio.tex
1817 +@@ -0,0 +1,92 @@
1818 +\documentclass [a4paper, 12pt] {article}
1819 +
1820 +\input{packages}
1821 +% \usepackage{fancyhdr}
1822 +% \pagestyle{fancy}
1823 +
1824 +% \fancyhead{}
1825 +% \fancyfoot{}
1826 +
1827 +% \renewcommand{\headrulewidth}{0pt} \renewcommand{\footrulewidth}{0pt}
1828 +% \fancyhead[L0, LE]{\thepage}
1829 +
1830 +\graphicspath{{fig/}}
1831 +
1832 +\begin {document}
1833 +% \setmainfont[
1834 +% Ligatures=TeX,
1835 +% Numbers=OldStyle, Proportional}
1836 +% ]{DejaVu Sans}
1837 +
1838 +\title {
1839 + \Large{\textbf{RELATÓRIO SOBRE O CÓDIGO FONTE DESTA PROJETO, CONJUNTO ORDENADO}} \\
1840 + \large{\textbf{projeto este que faz uso da estrutura de dados Red Black}} \\
1841 + \large {UFES Centro Universitário Norte do Espírito Santo}
1842 + \author{JOILNEN LEITE \ \href{mailto:joilnen.leite@edu.ufes.br}}
1843 + {\color{blue}joilnen.leite@edu.ufes.br}}
1844 +% \footnotesize{Joilnen Leite}
1845 +% \footnotesize{UFES Centro Universitário Norte do Espírito Santo}
1846 +% \footnotesize{\href{joilnen.leite@edu.ufes.br}}
1847 + \date{}
1848 +}
1849 +\maketitle
1850 +% \thispagestyle{empty}
1851 +% \begin{center}
1852 +% \Large{\textit{\textbf{Resenha Crítica do ensaio, \ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
1853 +% Científica"}}}
1854 +% \footnote{Artigo escrito por Antonio Joaquim Severino}
1855 +% \ \ \

```

```

1855 +% \end{center}
1856 +% \hrule
1857 +% \ \ \
1858 +% \ \ \
1859 +% \normalsize{\textit{\textbf{Autor: Joilnen Leite}}}\ \ \
1860 +% \normalsize{\textit{\textbf{Disciplina: Metodologia Cientifica}}}\ \ \
1861 +% \normalsize{\textit{\textbf{Curso: Ciência da Computação}}}\ \ \
1862 +% \normalsize{\textit{\textbf{Data: 24 de junho de 2020}}}\ \ \
1863 +% \ \ \
1864 +% \hrule
1865 +% \tableofcontents
1866 +% \vspace*{2cm}
1867 +% \textbf{Resumo: } Relatório básico sobre os mais comuns tipos de imagem digital e espaço de cores\ \
1868 +% \ \ \
1869 +% \indent \textbf{Palavras-chave: } imagens, rastreo, digital, digitalização\ \
1870 +% \small
1871 +% \section {Introdução}
1872 +% Neste texto será abordado o uso do histograma com relação ao estudo de imagens
1873 +% digitais.
1874 +
1875 +% \section{Conceito de Contraste}
1876 +% Contraste é a medida da diferença entre as áreas claras e escuras de uma imagem
1877 +
1878 +
1879 +% \section{Histograma e suas utilidades}
1880 +% Histograma bla bla
1881 +
1882 +% \subsection{Modificação de Histograma}
1883 +% Mod de Histograma bla bla
1884 +
1885 +% \subsection{Equalização do Histograma}
1886 +% Equali de Histograma bla bla
1887 +
1888 +% \subsection{Algoritmo de Equalização do Histograma}
1889 +% Descrever
1890 +
1891 +% \begin{itemize}
1892 + \item CIE 1931 XYZ é um dos primeiros espaços de cores, onde se criou a relação
1893 + entre a distribuição de ondas eletromagnéticas no espectro visível e a percepção
1894 + humana das cores. Onde XYZ se refere a um estímulo ternário no olho.
1895 + \item HSV ou HSL é Matiz, Saturação e Valor ou Luminância, distribuído numa geometria conica ou cilíndrica
1896 + permite uma distribuição mais intuitiva das cores
1897 + diferente do cubo RGB.
1898 + \item CMYK Ciano, Magenta, Amarela e Preto, diferente do espaço RGB que é aditivo
1899 + este espaço de cores funciona subtrativamente e utilizado no caso de impressão
1900 + em papel.
1901 + \end{itemize}
1902 +
1903 +% \section{Referências Bibliográficas}
1904 +% \nocite{*}
1905 +% \bibliography{mybib.bib}{ }
1906 +% \bibliographystyle{plain}
1907 +
1908 +% \end {document}
1909 +
1910 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
1911 index d14b82e..95550ee 100644
1912 --- a/src/conjunto_ordenado.h
1913 +++ b/src/conjunto_ordenado.h
1914 @@ -33,6 +33,8 @@ typedef struct rb_n {
1915
1916 /**
1917  * cria o primeiro noh, o raiz da arvore rb
1918  * * serah o primeiro elemento criado na arvore
1919  * * e o primeiro do conjunto ordenado
1920  */
1921 #define CRIA_CONJUNTO_ORDENDADO(a, key, dado) \
1922     rb_n a; \
1923     @@ -49,6 +51,7 @@ int limpa_conjunto(rb_n *a);
1924     int insere_elemento(rb_n *a, int k, void *e);
1925     int exclui_elemento(rb_n *a, int k);
1926     int verifica_pertencimento(rb_n *a, int k);
1927     int imprime_arvore(rb_n *a, short h);
1928
1929     rb_n* intersecta_conjuntos(rb_n *a, rb_n *b);
1930     rb_n* une_conjuntos(rb_n *a, rb_n *b);
1931 diff --git a/src/main.c b/src/main.c
1932 index 941ff61..2a3b16c 100644
1933 --- a/src/main.c
1934 +++ b/src/main.c
1935 @@ -6,6 +6,7 @@
1936  * inclusao da biblioteca implementada baseada em arvores red black
1937  */
1938 #include "conjunto_ordenado.h"
1939 #include <stdio.h>
1940 #include <string.h>
1941
1942 /**
1943 @@ -49,15 +50,16 @@ void adiciona_alguns_dados()
1944     char *nome;
1945     char *telefone;
1946     } meu_tipo;
1947     - meu_tipo mt;
1948     +
1949     + meu_tipo mt0, mt1;
1950
1951 /**
1952  * criacao do dado para o tipo do usuario
1953  */
1954 - mt.nome = (char *) malloc(sizeof("nome para teste") + 1);
1955 - strcpy(mt.nome, "nome para teste", sizeof("nome para teste") + 1);
1956 - mt.telefone = (char *) malloc(sizeof("999999999") + 1);
1957 - strcpy(mt.telefone, "999999999", sizeof("999999999") + 1);
1958 + mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
1959 + strcpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
1960 + mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
1961 + strcpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
1962
1963 /**
1964  * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
1965 @@ -71,7 +73,42 @@ void adiciona_alguns_dados()
1966     * @parametro 1 chave do noh raiz
1967     * @parametro 2 dado criado neste noh, pode ser de qqr tipo
1968     */
1969 - CRIA_CONJUNTO_ORDENDADO(conjunto_a, 1, mt);

```

```

1970 +     CRIA_CONJUNTO_ORDENDADO(conjunto_a, 10, mt0);
1971 +
1972 +     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
1973 +     strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
1974 +     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
1975 +     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
1976 +
1977 +     insere_elemento(&conjunto_a, 9, &mt1);
1978 + }
1979 +
1980 + int imprime_arvore(rb_n *a, short h)
1981 + {
1982 +     rb_n *it = a;
1983 +     int count = h;
1984 +     if(it)
1985 +     {
1986 +         for(; count; --count) printf(" ");
1987 +         printf("%d%d", it->k, it->c);
1988 +         if(it->l)
1989 +         {
1990 +             for(count = h - 2; h; --h) printf(" ");
1991 +             printf("%d%d", it->l->k, it->l->c);
1992 +         }
1993 +         if(it->r)
1994 +         {
1995 +             for(count = h + 2; h; --h) printf(" ");
1996 +             printf("%d%d", it->r->k, it->r->c);
1997 +         }
1998 +     }
1999 +     printf("\n");
2000 +     imprime_arvore(it->r->r, h - 1);
2001 +     imprime_arvore(it->r->l, h - 1);
2002 +     imprime_arvore(it->l->r, h - 1);
2003 +     imprime_arvore(it->l->l, h - 1);
2004 + }
2005 + }
2006 + }
2007 +
2008 +
2009 + diff --git a/src/makefile b/src/makefile
2010 + index 4c47828..bf6a1bd 100644
2011 + --- a/src/makefile
2012 + +++ b/src/makefile
2013 + @@ -6,9 +6,10 @@ C=gcc
2014 + S=main.c conjunto_ordenado.c
2015 + O=main.o conjunto_ordenado.o
2016 + B=rodeme
2017 + D=-g
2018 +
2019 + .c.o:
2020 + - $(C) -c $<
2021 + + $(C) -c $(D) $<
2022 + all: main.o conjunto_ordenado.o
2023 +     $(C) -o $(B) $(O)
2024 + clean:
2025 +
2026 + commit a2ad083e76c0b0ed593fc547e8e45194c5a28ebb
2027 + Author: Joilnen <joilnen@gmail.com>
2028 + Date: Thu Dec 16 16:22:28 2021 -0300
2029 +
2030 +     trabalhando o insere
2031 +
2032 + diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
2033 + index 939de06..c040e29 100644
2034 + --- a/src/conjunto_ordenado.c
2035 + +++ b/src/conjunto_ordenado.c
2036 + @@ -9,25 +9,21 @@ void (*imprime_elemento)(rb_n *);
2037 +
2038 + int imprime_conjunto(rb_n *a)
2039 + {
2040 +     -
2041 +     return 0;
2042 + }
2043 +
2044 + unsigned int cardinalidade(rb_n *a)
2045 + {
2046 +     -
2047 +     return 0;
2048 + }
2049 +
2050 + int ta_vazio(rb_n *a)
2051 + {
2052 +     -
2053 +     return 0;
2054 + }
2055 +
2056 + int limpa_conjunto(rb_n *a)
2057 + {
2058 +     -
2059 +     return 0;
2060 + }
2061 +
2062 + @@ -52,25 +48,21 @@ int insere_elemento(rb_n *a, int k, void *e)
2063 +
2064 + int exclui_elemento(rb_n *a, int k)
2065 + {
2066 +     -
2067 +     return 0;
2068 + }
2069 +
2070 + int verifica_pertencimento(rb_n *a, int k)
2071 + {
2072 +     -
2073 +     return 0;
2074 + }
2075 +
2076 + rb_n* intersecta_conjuntos(rb_n *a, rb_n *b)
2077 + {
2078 +     -
2079 +     return NULL;
2080 + }
2081 +
2082 + rb_n* une_conjuntos(rb_n *a, rb_n *b)
2083 + {
2084 +     -
2085 +     return NULL;

```

```

2086 }
2087
2088 @@ -81,7 +73,6 @@ rb_n* subtrai_conjuntos(rb_n *a, rb_n *b)
2089
2090 int destroi_conjunto(rb_n *a)
2091 {
2092     return 0;
2093 }
2094
2095 @@ -135,3 +126,4 @@ int seta_imprime_elemento(void (*i)(rb_n *n))
2096 }
2097
2098
2099
2100 +
2101
2102 commit bdba4da62308482975f9a2fc8f7ac829e0507880
2103 Author: Joilnen <joilnen@gmail.com>
2104 Date: Fri Dec 17 10:41:30 2021 -0300
2105
2106     começo da documentacao (relatorio) e foi consertado o insere
2107
2108 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
2109 index c748d51..05f6b10 100644
2110 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
2111 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
2112 index c79ccbd..e79cb0a 100644
2113 --- a/doc/tex/relatorio.tex
2114 +++ b/doc/tex/relatorio.tex
2115 @@ -1,14 +1,45 @@
2116 \documentclass[a4paper, 12pt]{article}
2117
2118 \input{packages}
2119 +\usepackage{fancyvrb}
2120 % \usepackage{fancyhdr}
2121 % \pagestyle{fancy}
2122
2123 % \fancyhead{}
2124 % \fancyfoot{}
2125 +\usepackage{listings}
2126 +\usepackage{xcolor}
2127 +
2128 +\definecolor{codegreen}{rgb}{0,0.6,0}
2129 +\definecolor{codegray}{rgb}{0.5,0.5,0.5}
2130 +\definecolor{codepurple}{rgb}{0.58,0,0.82}
2131 +\definecolor{backcolour}{rgb}{0.95,0.95,0.92}
2132 +
2133 +\lstdefinestyle{mystyle}{
2134 +    backgroundcolor=\color{backcolour},
2135 +    commentstyle=\color{codegreen},
2136 +    keywordstyle=\color{magenta},
2137 +    numberstyle=\tiny\color{codegray},
2138 +    stringstyle=\color{codepurple},
2139 +    basicstyle=\ttfamily\tiny,
2140 +    breakatwhitespace=false,
2141 +    breaklines=true,
2142 +    captionpos=b,
2143 +    keepspaces=true,
2144 +    numbers=left,
2145 +    numbersep=5pt,
2146 +    showspaces=false,
2147 +    showstringspaces=false,
2148 +    showtabs=false,
2149 +    frame=single,
2150 +    tabsize=2
2151 +}
2152 +
2153 +\lstset{style=mystyle}
2154
2155 % \renewcommand{\headrulewidth}{0pt} \renewcommand{\footrulewidth}{0pt}
2156 % \fancyhead[L0, LE]{\thepage}
2157 +\newcommand{\enf}[1]{\emph{\textbf{#1}}}
2158
2159 \graphicspath{{fig/}}
2160
2161 @@ -32,7 +63,7 @@
2162 \maketitle
2163 % \thispagestyle{empty}
2164 % \begin{center}
2165 -% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2166 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2167 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2168 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2169 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2170 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2171 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2172 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2173 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2174 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2175 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2176 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2177 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2178 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2179 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2180 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2181 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2182 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2183 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2184 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2185 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2186 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2187 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2188 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2189 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2190 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2191 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2192 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2193 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2194 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2195 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2196 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2197 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2198 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática
2199 +% \Large\textit{\textbf{Resenha Crítica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática

```

```

2200 -
2201 -\begin{itemize}
2202 - \item CIE 1931 XYZ é um dos primeiros espaços de cores, onde se criou a relação
2203 -entre a distribuição de ondas eletromagnéticas no espectro visível e a percepção
2204 -humana das cores. Onde XYZ se refere a um estímulo ternário no olho.
2205 - \item HSV ou HSL é Matiz, Saturação e Valor ou Luminância, distribuído numa geometria cônica ou cilíndrica
2206 -permite uma distribuição mais intuitiva das cores
2207 -diferente do cubo RGB.
2208 - \item CMYK Ciano, Magenta, Amarelo e Preto, diferente do espaço RGB que é aditivo
2209 -este espaço de cores funciona subtrativamente e é utilizado no caso de impressão
2210 -em papel.
2211 -\end{itemize}
2212 -\section{Referências Bibliográficas}
2213 +Esta biblioteca é composta por dois arquivos, conjunto\_ordenado.c e seu cabeçalho,
2214 +conjunto\_ordenado.h, sendo necessário apenas para seu uso a inclusão via, \\\
2215 +\enfn{\include "conjunto\_ordenado.h"} e compilação conjunta com conjunto\_ordenado.c
2216 +
2217 +Além da convencional inclusão e compilação, o usuário da biblioteca, pessoa
2218 +que escreve o programa cliente, aqui neste trabalho exemplificado pelo o arquivo \enfn{main.c}
2219 +deverá escolher definir seu tipo dado, através de uma struct ou utilizar um tipo de dado nativo da
2220 +linguagem.
2221 +\renewcommand{\lstlistingname}{Fonte}
2222 +\begin{lstlisting}[language=c, caption={Exemplo do uso da biblioteca conjunto\_ordenado}]
2223 +
2224 +\#include "conjunto\_ordenado.h"
2225 +
2226 +int main()
2227 +{
2228 + typedef struct {
2229 + char *nome;
2230 + char *telefone;
2231 + } meu_tipo;
2232 + meu_tipo mt0, mt1;
2233 + mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
2234 + strncpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
2235 + mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
2236 + strncpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
2237 +
2238 + seta_imprime_elemento(imprime_meu_tipo);
2239 +
2240 + CRIA_CONJUNTO_ORDENDADO(conjunto_a, 10, mt0);
2241 + mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
2242 + strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
2243 + mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
2244 + strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
2245 +
2246 + insere_elemento(&conjunto_a, 9, &mt1);
2247 +}
2248 +
2249 +
2250 +\end{lstlisting}
2251 +% \section{Referências Bibliográficas}
2252 +\nocite{*}
2253 +\bibliography{mybib.bib}{*}
2254 +\bibliographystyle{plain}
2255 +diff --git a/src/conjunto\_ordenado.c b/src/conjunto\_ordenado.c
2256 +index c040e29..67596da 100644
2257 +--- a/src/conjunto\_ordenado.c
2258 +++ b/src/conjunto\_ordenado.c
2259 @@ -4,6 +4,7 @@
2260 + * de implementacao das funcoes da biblioteca conjunto ordenado
2261 + */
2262 +\#include "conjunto\_ordenado.h"
2263 +\#include <stdio.h>
2264 +
2265 +void (*imprime_elemento)(rb_n *);
2266 +
2267 @@ -30,6 +31,7 @@ int limpa_conjunto(rb_n *a)
2268 +int insere_elemento(rb_n *a, int k, void *e)
2269 +{
2270 + rb_n *it = a;
2271 + printf("k %d it->k %d\n", k, it->k);
2272 + while (it)
2273 + {
2274 + if (k < it->k)
2275 @@ -41,6 +43,7 @@ int insere_elemento(rb_n *a, int k, void *e)
2276 + it = e;
2277 + it->k = k;
2278 + it->l = it->r = NULL;
2279 + it = NULL;
2280 + }
2281 + }
2282 + return 0;
2283 +
2284 +commit 6874ee55997cb34d17ac04f65a2e294903af3747
2285 +Author: Joilnen <joilnen@gmail.com>
2286 +Date: Fri Dec 17 13:56:50 2021 -0300
2287 +
2288 +melhorando o relatorio
2289 +
2290 +diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
2291 +index 05f6b10..aec8bba 100644
2292 +Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
2293 +diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
2294 +index e79cb0a..829d600 100644
2295 +--- a/doc/tex/relatorio.tex
2296 +++ b/doc/tex/relatorio.tex
2297 @@ -94,7 +94,6 @@ deverá escolher definir seu tipo dado, através de uma struct ou utilizar um ti
2298 + linguagem.
2299 + \renewcommand{\lstlistingname}{Fonte}
2300 + \begin{lstlisting}[language=c, caption={Exemplo do uso da biblioteca conjunto\_ordenado}]
2301 + -\include "conjunto\_ordenado.h"
2302 +
2303 + int main()
2304 @@ -119,9 +118,8 @@ int main()
2305 +
2306 + insere_elemento(&conjunto_a, 9, &mt1);
2307 + }
2308 + \end{lstlisting}
2309 +
2310 +\section{Lista de Estruturas e Funções Implementadas}
2311 +Como foi especificado a estrutura de dados utilizada para guardar os dados neste projeto
2312 +é a árvore binária de busca Red Black, no código esta árvore e seus nós são representados
2313 +pela estrutura \enfn{rb_n}\footnote{Red Black Node ou nó Red Black}, a semântica escolhida

```

```

2315 +é a seguinte, representamos apenas os nós e seus relacionamentos, a árvore completa fica
2316 +referenciada no ato da criação do nó raiz no código cliente, logo no exemplo acima
2317 +temos uma árvore nomeada \enf{conjunto_a}, enquanto nós filhos nessa árvore é adicionado
2318 +com \enf{insere_elemento}
2319 +\ \\
2320
2321 +\noindent Estrutura principal
2322 +\begin{lstlisting}[language=c, caption={Nó Red Black}]
2323 +/**
2324 + * composicao do noh da RB
2325 + * @rb_n noh red black
2326 + * @k chave (k de key)
2327 + * @c cor (c de color ou cor, 0-black, 1-red)
2328 + * @t tipo (t de tipo ou type)
2329 + * @d dado (de de data ou dados)
2330 + * @l @r @p l-left filho a esquerda, r-right filho a direita e p-parent, noh pai
2331 + */
2332 +typedef struct rb_n {
2333 +     int k;
2334 +     char c;
2335 +     void *d;
2336 +     struct rb_n *l, *r, *p;
2337 +} rb_n;
2338 +\end{lstlisting}
2339 +Significado dos componentes da estrutura \enf{rb_n}
2340 +\begin{itemize}
2341 +     \item \enf{k} - chave
2342 +     \item \enf{c} - cor (0 - preto e 1 - vermelho, apenas o primeiro bit utilizado)
2343 +     \item \enf{d} - dado generico
2344 +     \item \enf{l, r, p} - ponteiro para filho a esquerda, filho a direita e nó pai
2345 +\end{itemize}
2346 +Funções que manipulam os nós na Árvore
2347 +\begin{lstlisting}[language=c, caption={Funções da biblioteca}]
2348 +/**
2349 + * cria o primeiro noh, o raiz da arvore rb
2350 + * serah o primeiro elemento criado na arvore
2351 + * e o primeiro do conjunto ordenado
2352 + */
2353 +#define CRIA_CONJUNTO_ORDENDADO(a, key, dado) \
2354 +     rb_n a; \
2355 +     a.k = key; \
2356 +     a.c = 0; \
2357 +     a.d = &dado; \
2358 +     a.p = a.l = a.r = NULL;
2359 +int imprime_conjunto(rb_n *a);
2360 +unsigned int cardinalidade(rb_n *a);
2361 +int ta_vazio(rb_n *a);
2362 +int limpa_conjunto(rb_n *a);
2363 +
2364 +int insere_elemento(rb_n *a, int k, void *e);
2365 +int exclui_elemento(rb_n *a, int k);
2366 +int verifica_pertencimento(rb_n *a, int k);
2367 +int imprime_arvore(rb_n *a, short h);
2368 +
2369 +rb_n* intersecta_conjuntos(rb_n *a, rb_n *b);
2370 +rb_n* une_conjuntos(rb_n *a, rb_n *b);
2371 +rb_n* subtrai_conjuntos(rb_n *a, rb_n *b);
2372 +
2373 +int destroi_conjunto(rb_n *a);
2374 +
2375 +int troca_cor(rb_n *a);
2376 +int rotaciona_esquerda(rb_n *r, rb_n *w);
2377 +int rotaciona_direita(rb_n *r, rb_n *w);
2378 +
2379 +void (*imprime_elemento)(rb_n *);
2380 +int seta_imprime_elemento(void (*i)(rb_n *));
2381 +\end{lstlisting}
2382 +
2383 +% \section{Referências Bibliográficas}
2384 +\nocite{*}
2385 +\bibliography{mybib.bib}{*}
2386 +diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
2387 +index 67596da..bc239c3 100644
2388 +--- a/src/conjunto_ordenado.c
2389 ++++ b/src/conjunto_ordenado.c
2390 +@@ -32,6 +32,8 @@ int insere_elemento(rb_n *a, int k, void *e)
2391 + {
2392 +     rb_n *it = a;
2393 +     printf("k %d it->k %d\n", k, it->k);
2394 +     if((int)k == (int)(it->k))
2395 +         return -1;
2396 +     while (it)
2397 +     {
2398 +         if (k < it->k)
2399 +
2400 +commit 14a0694781b3e7d9c64213ed69ee844863c48a96
2401 +Author: Joilnen <joilnen@gmail.com>
2402 +Date: Fri Dec 17 15:54:03 2021 -0300
2403 +
2404 +     atualizando ignore file from git
2405 +
2406 +diff --git a/.gitignore b/.gitignore
2407 +index 0446a97..15638f0 100644
2408 +--- a/.gitignore
2409 ++++ b/.gitignore
2410 +@@ -5,3 +5,7 @@
2411 + tags
2412 + rodeme
2413 + latex.out
2414 + *.aux
2415 + *.idx
2416 + *.log
2417 + *.out
2418 +
2419 +commit a6ddf09704424642cc4cf37772a27843e3af636a
2420 +Author: Joilnen <joilnen@gmail.com>
2421 +Date: Fri Dec 17 17:28:30 2021 -0300
2422 +
2423 +     modificado para deixar o TAD mais explicito embora a abstracao ja feita
2424 +     sobre rb estritamente ja seria um TAD
2425 +
2426 +diff --git a/rd.h b/rd.h
2427 +new file mode 100644
2428 +index 0000000..e69de29
2429 +diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
2430 +index bc239c3..8e3abe4 100644

```

```

2431 --- a/src/conjunto_ordenado.c
2432 +++ b/src/conjunto_ordenado.c
2433 @@ -1,134 +1,20 @@
2434 -/**
2435  * aluno: joilnen leite
2436  * arquivo separado do main, como especificado
2437  * de implementacao das funcoes da biblioteca conjunto ordenado
2438  */
2439 #include "conjunto_ordenado.h"
2440 #include <stdio.h>
2441
2442 -void (*imprime_elemento)(rb_n *);
2443 -
2444 -int imprime_conjunto(rb_n *a)
2445 +conjord *conjord_cria()
2446 {
2447     return 0;
2448 }
2449 -
2450 -unsigned int cardinalidade(rb_n *a)
2451 {-{
2452     return 0;
2453 }
2454 + conjord *c = (conjord *) malloc (sizeof(conjord));
2455 + c->raiz = NULL;
2456 + c->cardinalidade = 0;
2457
2458 -int ta_vazio(rb_n *a)
2459 {-{
2460     return 0;
2461 + return c;
2462 }
2463
2464 -int limpa_conjunto(rb_n *a)
2465 +int conjord_ah_vazio(conjord *a)
2466 {
2467     return 0;
2468 + return !a->raiz;
2469 }
2470
2471 -int insere_elemento(rb_n *a, int k, void *e)
2472 +int conjord_esvazia(conjord *a)
2473 {
2474     rb_n *it = a;
2475     printf("k %d it->k %d\n", k, it->k);
2476     if((int)k == (int)(it->k))
2477         return -1;
2478     while (it)
2479     {
2480         if (k < it->k)
2481             it = it->l;
2482         else if (k > it->k)
2483             it = it->r;
2484         if (!it)
2485         {
2486             it = e;
2487             it->k = k;
2488             it->l = it->r = NULL;
2489             it = NULL;
2490         }
2491     }
2492     return 0;
2493 }
2494
2495 -int exclui_elemento(rb_n *a, int k)
2496 {-{
2497     return 0;
2498 }
2499
2500 -int verifica_pertencimento(rb_n *a, int k)
2501 {-{
2502     return 0;
2503 }
2504
2505 -rb_n* intersecta_conjuntos(rb_n *a, rb_n *b)
2506 {-{
2507     return NULL;
2508 }
2509
2510 -rb_n* une_conjuntos(rb_n *a, rb_n *b)
2511 {-{
2512     return NULL;
2513 }
2514
2515 -rb_n* subtrai_conjuntos(rb_n *a, rb_n *b)
2516 {-{
2517     return NULL;
2518 }
2519
2520 -int destroi_conjunto(rb_n *a)
2521 {-{
2522     return 0;
2523 }
2524
2525 -int rotaciona_esquerda(rb_n *r, rb_n *w)
2526 {-{
2527     rb_n *v = w->r;
2528     w->r = v->l;
2529     if (v->l)
2530         v->l->p = w;
2531     v->p = w->p;
2532     if (!w->p)
2533         r = v;
2534     else
2535     {
2536         if (w == w->p->l)
2537             w->p->l = v;
2538         else
2539             w->p->r = v;
2540     }
2541     v->l = w;
2542     w->p = v;
2543
2544     return 0;
2545 }
2546 -

```

```

2547 -int rotaciona_direita(rb_n *r, rb_n *w)
2548 -{
2549     rb_n *v = w->l;
2550     w->l = v->r;
2551     if (v->r)
2552         v->r->p = w;
2553     v->p = w->p;
2554     if (!w->p)
2555         r = v;
2556     else
2557     {
2558         if (w == w->p->r)
2559             w->p->r = v;
2560         else
2561             w->p->l = v;
2562     }
2563     v->r = w;
2564     w->p = v;
2565     return 0;
2566 -}
2567 -}
2568 -
2569 -int seta_imprime_elemento(void (*i)(rb_n *n))
2570 -{
2571     imprime_elemento = i;
2572 -}
2573 -
2574 -
2575 -
2576 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
2577 index 95550ee..d8ac556 100644
2578 --- a/src/conjunto_ordenado.h
2579 +++ b/src/conjunto_ordenado.h
2580 @@ -1,70 +1,34 @@
2581 #ifndef CONJUNTO_ORDENADO_H
2582 #define CONJUNTO_ORDENADO_H
2583
2584 /**
2585  * aluno: joilnen leite
2586  */
2587 #include <stdlib.h>
2588
2589 /**
2590  * rubro negra
2591  * 0 - todo noh eh vermelho ou preto
2592  * 1 - raiz sempre preta
2593  * 2 - noh NULL considerado noh externo preto
2594  * 3 - vermelhos tem filhos pretos
2595  * 4 - noh para suas folhas tem o mesmo numero de nohs preto
2596  */
2597
2598 /**
2599  * composicao do noh da RB
2600  * @rb_n noh red black
2601  * @k chave (k de key)
2602  * @c cor (c de color ou cor, 0-black, 1-red)
2603  * @t tipo (t de tipo ou type)
2604  * @d dado (de de data ou dados)
2605  * @l @r @p l-efit filho a esquerda, r-ight filho a direita e p-arent, noh pai
2606  */
2607 #typedef struct rb_n {
2608     int k;
2609     char c;
2610     void *d;
2611     struct rb_n *l, *r, *p;
2612 } rb_n;
2613 #include "red_black.h"
2614
2615 /**
2616  * cria o primeiro noh, o raiz da arvore rb
2617  * serah o primeiro elemento criado na arvore
2618  * e o primeiro do conjunto ordenado
2619  * estrutura da TAD explicitamente
2620  * esta estrutura oculta o uso da red black
2621  * suas funcoes e funcionamento interno do cliente
2622  */
2623 #define CRIA_CONJUNTO_ORDENADO(a, key, dado) \
2624     rb_n a; \
2625     a.k = key; \
2626     a.c = 0; \
2627     a.d = &dado; \
2628     a.p = a.l = a.r = NULL;
2629
2630 -int imprime_conjunto(rb_n *a);
2631 -unsigned int cardinalidade(rb_n *a);
2632 -int ta_vazio(rb_n *a);
2633 -int limpa_conjunto(rb_n *a);
2634
2635 -int insere_elemento(rb_n *a, int k, void *e);
2636 -int exclui_elemento(rb_n *a, int k);
2637 -int verifica_pertencimento(rb_n *a, int k);
2638 -int imprime_arvore(rb_n *a, short h);
2639
2640 -rb_n* intersecta_conjuntos(rb_n *a, rb_n *b);
2641 -rb_n* une_conjuntos(rb_n *a, rb_n *b);
2642 -rb_n* subtrai_conjuntos(rb_n *a, rb_n *b);
2643
2644 -int destroi_conjunto(rb_n *a);
2645
2646 -int troca_cor(rb_n *a);
2647 -int rotaciona_esquerda(rb_n *r, rb_n *w);
2648 -int rotaciona_direita(rb_n *r, rb_n *w);
2649
2650 -extern void (*imprime_elemento)(rb_n *);
2651 -int seta_imprime_elemento(void (*i)(rb_n *n));
2652 #typedef struct conjord {
2653     rb_n *raiz;
2654     int cardinalidade;
2655 } conjord;
2656
2657 +conjord *conjord_cria();
2658 +void conjord_imprime();
2659 +unsigned int conjord_cardinalidade();
2660 +int conjord_ah_vazio();
2661 +int conjord_esvazia();
2662 +void conjord_destroi(conjord **a);

```



```

2663 +
2664 +void conjord_insererelemento(conjord *a, int chave, void *elemento);
2665 +void conjord_exclui_elemento(conjord *a, int chave);
2666 +int conjord_existe_elemento(conjord *a, int chave);
2667 +
2668 +conjord *conjord_intersecciona(conjord *a, conjord *b);
2669 +conjord *conjord_une(conjord *a, conjord *b);
2670 +conjord *conjord_subtrai(conjord *a, conjord *b);
2671 +
2672 +void conjord_imprime_rb();
2673
2674 #endif
2675
2676 diff --git a/src/main.c b/src/main.c
2677 index 2a3b16c..d3744d9 100644
2678 --- a/src/main.c
2679 +++ b/src/main.c
2680 @@ -15,33 +15,10 @@
2681  */
2682 void imprime_meu_tipo(rb_n *);
2683
2684 /**
2685  * funcao que adiciona alguns dados de teste
2686  */
2687 -void adiciona_alguns_dados();
2688 -
2689 /**
2690  * funcao main de teste separada do codigo da biblioteca como especificado
2691  */
2692 int main()
2693 {
2694     adiciona_alguns_dados();
2695 }
2696
2697 /**
2698  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
2699  * exemplo
2700  */
2701 -void imprime_meu_tipo(rb_n *n)
2702 {
2703     -
2704     -
2705     -
2706     -
2707     /**
2708     * funcao que adiciona alguns dados de teste
2709     */
2710     -void adiciona_alguns_dados()
2711     {
2712         /**
2713         * definicao de um tipo exemplo pelo o usuario
2714         @@ -64,7 +41,7 @@ void adiciona_alguns_dados()
2715         /**
2716         * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
2717         */
2718         - seta_imprime_elemento(imprime_meu_tipo);
2719         + rb_sete_impr_elem(imprime_meu_tipo);
2720
2721         /**
2722         * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
2723         @@ -80,7 +57,10 @@ void adiciona_alguns_dados()
2724         mt1.telefone = (char *) malloc(sizeof("9999999999") + 1);
2725         strncpy(mt1.telefone, "9999999999", sizeof("9999999999") + 1);
2726
2727         - insere_elemento(&conjunto_a, 9, &mt1);
2728         + rb_ins_elem(&conjunto_a, 9, &mt1);
2729         +
2730         + conjord *conj_a = conjord_cria();
2731         + printf("eh vazio %d", conjord_eh_vazio(conj_a));
2732     }
2733
2734 int imprime_arvore(rb_n *a, short h)
2735 @@ -111,4 +91,14 @@ int imprime_arvore(rb_n *a, short h)
2736     }
2737 }
2738
2739 /**
2740  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
2741  * exemplo
2742  */
2743 +void imprime_meu_tipo(rb_n *n)
2744 +{
2745 +
2746 +
2747 +
2748 +
2749 +
2750 diff --git a/src/makefile b/src/makefile
2751 index bf6a1bd..6fe7410 100644
2752 --- a/src/makefile
2753 +++ b/src/makefile
2754 @@ -3,14 +3,14 @@
2755  #
2756
2757  C=gcc
2758  S=main.c conjunto_ordenado.c
2759  O=main.o conjunto_ordenado.o
2760  S+=main.c conjunto_ordenado.c red_black.c
2761  O+=main.o conjunto_ordenado.o red_black.o
2762  B=rodeme
2763  D=-g
2764
2765  .c.o:
2766  $(C) -c $(D) $<
2767  -all: main.o conjunto_ordenado.o
2768  +all: $(D)
2769  $(C) -o $(B) $(D)
2770  clean:
2771  rm -f *.o $(B)
2772 diff --git a/src/red_black.c b/src/red_black.c
2773 new file mode 100644
2774 index 0000000..83b4bbe
2775 --- /dev/null
2776 +++ b/src/red_black.c
2777 @@ -0,0 +1,108 @@
2778 /**

```

```

2779  * * aluno: joilnen leite
2780  * * arquivo separado do main, como especificado
2781  * * de implementacao das funcoes da redblack
2782  * * utilizada no conjunto ordenado
2783  * */
2784  #include "red_black.h"
2785  #include <stdio.h>
2786
2787  void (*rb_impr_elem)(rb_n *);
2788
2789  int rb_ins_elem(rb_n *a, int k, void *e)
2790  {
2791      rb_n *it = a;
2792      printf("k %d it->k %d\n", k, it->k);
2793      if((int)k == (int)(it->k))
2794          return -1;
2795      while (it)
2796      {
2797          if (k < it->k)
2798              it = it->l;
2799          else if (k > it->k)
2800              it = it->r;
2801          if (!it)
2802          {
2803              it = e;
2804              it->k = k;
2805              it->l = it->r = NULL;
2806              it = NULL;
2807          }
2808      }
2809      return 0;
2810  }
2811
2812  int rb_exclui_elem(rb_n *a, int k)
2813  {
2814      return 0;
2815  }
2816
2817  rb_n* rb_intersec(rb_n *a, rb_n *b)
2818  {
2819      return NULL;
2820  }
2821
2822  rb_n* rb_une(rb_n *a, rb_n *b)
2823  {
2824      return NULL;
2825  }
2826
2827  rb_n* rb_subtrai(rb_n *a, rb_n *b)
2828  {
2829      return NULL;
2830  }
2831
2832  int rb_destroi(rb_n **a)
2833  {
2834      return 0;
2835  }
2836
2837  int rb_rot_esq(rb_n *r, rb_n *w)
2838  {
2839      rb_n *v = w->r;
2840      w->r = v->l;
2841      if (v->l)
2842          v->l->p = w;
2843      v->p = w->p;
2844      if (!w->p)
2845          r = v;
2846      else
2847      {
2848          if (w == w->p->l)
2849              w->p->l = v;
2850          else
2851              w->p->r = v;
2852      }
2853      v->l = w;
2854      w->p = v;
2855
2856      return 0;
2857  }
2858
2859  int rb_rot_dir(rb_n *r, rb_n *w)
2860  {
2861      rb_n *v = w->l;
2862      w->l = v->r;
2863      if (v->r)
2864          v->r->p = w;
2865      v->p = w->p;
2866      if (!w->p)
2867          r = v;
2868      else
2869      {
2870          if (w == w->p->r)
2871              w->p->r = v;
2872          else
2873              w->p->l = v;
2874      }
2875      v->r = w;
2876      w->p = v;
2877
2878      return 0;
2879  }
2880
2881  int rb_sete_impr_elem(void (*i)(rb_n *n))
2882  {
2883      rb_impr_elem = i;
2884  }
2885
2886  diff --git a/src/red_black.h b/src/red_black.h
2887  new file mode 100644
2888  index 0000000..dd06325
2889  --- /dev/null
2890  +++ b/src/red_black.h
2891  @@ -0,0 +1,65 @@
2892  #ifndef REDBLACK_H
2893  #define REDBLACK_H
2894

```

```

2895 +/**
2896 + * aluno: joilnen leite
2897 + */
2898 #include <stdlib.h>
2899 +
2900 +/**
2901 + * rubro negra
2902 + * 0 - todo noh eh vermelho ou preto
2903 + * 1 - raiz sempre preta
2904 + * 2 - noh NULL considerado noh externo preto
2905 + * 3 - vermelhos tem filhos pretos
2906 + * 4 - noh para suas folhas tem o mesmo numero de nohs preto
2907 + */
2908 +
2909 +/**
2910 + * composicao do noh da RB
2911 + * @rb_n noh red black
2912 + * @k chave (k de key)
2913 + * @c cor (c de color ou cor, 0-black, 1-red)
2914 + * @t tipo (t de tipo ou type)
2915 + * @d dado (de de data ou dados)
2916 + * @l @r @p l-eft filho a esquerda, r-ight filho a direita e p-arent, noh pai
2917 + */
2918 +typedef struct rb_n {
2919 +     int k;
2920 +     char c:1;
2921 +     void *d;
2922 +     struct rb_n *l, *r, *p;
2923 +} rb_n;
2924 +
2925 +/**
2926 + * cria o primeiro noh, o raiz da arvore rb
2927 + * serah o primeiro elemento criado na arvore
2928 + * e o primeiro do conjunto ordenado
2929 + */
2930 #define CRIA_CONJUNTO_ORDENDADO(a, key, dado) \
2931 +     rb_n a; \
2932 +     a.k = key; \
2933 +     a.c = 0; \
2934 +     a.d = &dado; \
2935 +     a.p = a.l = a.r = NULL;
2936 +
2937 +unsigned int rb_nelem(rb_n *a);
2938 +
2939 +int rb_ins_elem(rb_n *a, int k, void *e);
2940 +int rb_exclui_elem(rb_n *a, int k);
2941 +int rb_exite_elem(rb_n *a, int k);
2942 +
2943 +rb_n* rb_intersec(rb_n *a, rb_n *b);
2944 +rb_n* rb_une(rb_n *a, rb_n *b);
2945 +rb_n* rb_subtrai(rb_n *a, rb_n *b);
2946 +
2947 +int rb_destroi(rb_n **a);
2948 +
2949 +int rb_troca_cor(rb_n *a);
2950 +int rb_rot_esq(rb_n *r, rb_n *w);
2951 +int rb_rot_dir(rb_n *r, rb_n *w);
2952 +
2953 +extern void (*rb_impr_elem)(rb_n *);
2954 +int rb_sete_impr_elem(void (*i)(rb_n *n));
2955 +
2956 #endif
2957
2958 commit c041733699cd48be8fa20ea17c05818bd2b8029d
2959 Author: Joilnen <joilnen@gmail.com>
2960 Date:   Fri Dec 17 17:34:49 2021 -0300
2961
2962     deletado arquivo n utilizado
2963
2964 diff --git a/rd.h b/rd.h
2965 deleted file mode 100644
2966 index e69de29..0000000
2967
2968 commit 4f9bda9d86348ebbb9321d64f9adca5e1e00ac74
2969 Author: Joilnen <joilnen@gmail.com>
2970 Date:   Fri Dec 17 18:02:48 2021 -0300
2971
2972     acentuando as palavras em portugues
2973
2974 diff --git a/LEIAME b/LEIAME
2975 index 17b9ab4..9c99c20 100644
2976 --- a/LEIAME
2977 +++ b/LEIAME
2978 @@ -1,10 +1,10 @@
2979 -dentro do diretorio docs estao relatorios e arquivos
2980 +dentro do diretorio docs estão relatórios e arquivos
2981 -relacionados ao processo de desenvolvimento
2982
2983 -binario sera gerado no mesmo diretorio dos fontes,
2984 -src, atraves do comando make local
2985 -dentro daquele diretorio, binario nomeado, rodeme
2986 -comando a ser utilizado
2987 +executável será gerado no mesmo diretório dos fontes,
2988 +src, através do comando, make
2989 +dentro daquele diretorio, executável nomeado, rodeme
2990 +comandos a ser utilizado
2991
2992 make # para criar o binario
2993 make clean # para limpar, apagar objetos e binario
2994
2995 commit 2de82428748b3dee13cda5a9602f720f71f6cbce
2996 Author: Joilnen <joilnen@gmail.com>
2997 Date:   Fri Dec 17 18:27:36 2021 -0300
2998
2999     uma comparacao de chaves
3000
3001 diff --git a/src/red_black.c b/src/red_black.c
3002 index 83b4bbe..269668f 100644
3003 --- a/src/red_black.c
3004 +++ b/src/red_black.c
3005 @@ -106,3 +106,4 @@ int rb_sete_impr_elem(void (*i)(rb_n *n))
3006     rb_impr_elem = i;
3007 }
3008 +
3009 +
3010

```

```

3011 commit bfeb9970eb1c1217637327a65d437731aa68fbd3
3012 Author: Joilnen <joilnen@gmail.com>
3013 Date: Sat Dec 18 06:28:01 2021 -0300
3014
3015 reescrevendo rb para ficar oculta em outra camada que serah o TAD
3016
3017 diff --git a/src/main.c b/src/main.c
3018 index d3744d9..858819d 100644
3019 --- a/src/main.c
3020 +++ b/src/main.c
3021 @@ -50,17 +50,20 @@ int main()
3022      * @parametro 1 chave do noh raiz
3023      * @parametro 2 dado criado neste noh, pode ser de qqr tipo
3024      */
3025 - CRIA_CONJUNTO_ORDENDADO(conjunto_a, 10, mt0);
3026 + rb_n *conjunto_a = rb_cr(10, &mt0);
3027
3028 mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
3029 strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
3030 mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
3031 strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
3032
3033 - rb_ins_elem(&conjunto_a, 9, &mt1);
3034 + rb_ins_elem(conjunto_a, 9, (meu_tipo *)&mt1);
3035
3036 + /**
3037 + * testando apenas rb no momento
3038 conjord *conj_a = conjord_cria();
3039 printf("eh vazio %d", conjord_eh_vazio(conj_a));
3040 */
3041 }
3042
3043 int imprime_arvore(rb_n *a, short h)
3044 diff --git a/src/red_black.c b/src/red_black.c
3045 index 269668f..4967af2 100644
3046 --- a/src/red_black.c
3047 +++ b/src/red_black.c
3048 @@ -9,6 +9,17 @@
3049
3050 void (*rb_impr_elem)(rb_n *);
3051
3052 +rb_n* rb_cr(int key, void *dado)
3053 +{
3054 + rb_n *a = (rb_n *) malloc(sizeof(rb_n));
3055 + a->k = key;
3056 + a->c = 0;
3057 + a->d = dado;
3058 + a->p = a->l = a->r = NULL;
3059 +
3060 + return a;
3061 +}
3062 +
3063 int rb_ins_elem(rb_n *a, int k, void *e)
3064 {
3065     rb_n *it = a;
3066     @@ -23,13 +34,16 @@ int rb_ins_elem(rb_n *a, int k, void *e)
3067         it = it->r;
3068         if (!it)
3069         {
3070             it = e;
3071             it = (rb_n *) malloc(sizeof(rb_n));
3072             it->d = e;
3073             it->k = k;
3074             it->l = it->r = NULL;
3075             it = NULL;
3076             return 0;
3077         }
3078     }
3079     return 0;
3080 +
3081 + return -1;
3082 }
3083
3084 int rb_exclui_elem(rb_n *a, int k)
3085 diff --git a/src/red_black.h b/src/red_black.h
3086 index dd06325..fa661c7 100644
3087 --- a/src/red_black.h
3088 +++ b/src/red_black.h
3089 @@ -36,12 +36,7 @@ typedef struct rb_n {
3090     * serah o primeiro elemento criado na arvore
3091     * e o primeiro do conjunto ordenado
3092     */
3093 -#define CRIA_CONJUNTO_ORDENDADO(a, key, dado) \
3094 - rb_n a; \
3095 - a.k = key; \
3096 - a.c = 0; \
3097 - a.d = &dado; \
3098 - a.p = a.l = a.r = NULL;
3099 +rb_n* rb_cr(int key, void *dado);
3100
3101 unsigned int rb_nelem(rb_n *a);
3102
3103
3104 commit 36248abc22a056b90f828d4580dd668c9e3fc08e
3105 Author: Joilnen <joilnen@gmail.com>
3106 Date: Sat Dec 18 07:18:33 2021 -0300
3107
3108 inserir e imprimir funcionando ainda bem rudimentarmente, mas ja funciona
3109
3110 diff --git a/src/main.c b/src/main.c
3111 index 858819d..817bc54 100644
3112 --- a/src/main.c
3113 +++ b/src/main.c
3114 @@ -28,11 +28,11 @@ int main()
3115     char *telefone;
3116     } meu_tipo;
3117
3118 - meu_tipo mt0, mt1;
3119
3120 /**
3121 * criacao do dado para o tipo do usuario
3122 */
3123 + meu_tipo mt0;
3124 mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
3125 strncpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
3126 mt0.telefone = (char *) malloc(sizeof("999999999") + 1);

```

```

3127 @@ -52,12 +52,21 @@ int main()
3128 /*
3129     rb_n *conjunto_a = rb_cr(10, &mt0);
3130
3131 +     meu_tipo mt1;
3132 +     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
3133 +     strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
3134 +     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
3135 +     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
3136 +     rb_ins_elem(conjunto_a, 9, &mt1);
3137 -
3138 -     rb_ins_elem(conjunto_a, 9, (meu_tipo *)&mt1);
3139 +     meu_tipo mt2;
3140 +     mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
3141 +     strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
3142 +     mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
3143 +     strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
3144 +     rb_ins_elem(conjunto_a, 11, &mt2);
3145 +
3146 +     rb_pr(conjunto_a);
3147
3148     /**
3149     * testando apenas rb no momento
3150 @@ -101,7 +110,6 @@ int imprime_arvore(rb_n *a, short h)
3151 void imprime_meu_tipo(rb_n *n)
3152 {
3153 -
3154 -
3155 }
3156
3157
3158 diff --git a/src/red_black.c b/src/red_black.c
3159 index 4967af2..c658162 100644
3160 --- a/src/red_black.c
3161 +++ b/src/red_black.c
3162 @@ -24,21 +24,29 @@ int rb_ins_elem(rb_n *a, int k, void *e)
3163 {
3164     rb_n *it = a;
3165     printf("k %d it->k %d\n", k, it->k);
3166     if((int)k == (int)(it->k))
3167         return -1;
3168     while (it)
3169     {
3170 +         rb_n *tmp = it;
3171 +         char rl = 0;
3172 +         if (k < it->k)
3173 -             it = it->l;
3174 +             it = it->l, rl = 0 << 0;
3175 +         else if (k > it->k)
3176 -             it = it->r;
3177 +             it = it->r, rl = 1 << 0;
3178 +         else return -1;
3179 +
3180 +         printf("rl %d\n", rl);
3181 +
3182 +         if (!it)
3183 +         {
3184             it = (rb_n *) malloc(sizeof(rb_n));
3185             it->d = e;
3186             it->k = k;
3187             it->l = it->r = NULL;
3188             it = NULL;
3189 -
3190 +             if (rl == 0)
3191 +                 tmp->l = it;
3192 +             else if (rl == 1)
3193 +                 tmp->r = it;
3194 +             return 0;
3195         }
3196     }
3197 @@ -120,4 +128,14 @@ int rb_set_a_impr_elem(void (*i)(rb_n *n))
3198     rb_impr_elem = i;
3199 }
3200
3201 +void rb_pr(rb_n *a)
3202 +{
3203 +    if (a)
3204 +    {
3205 +        printf("k:%dc:%d\n", a->k, a->c);
3206 +        rb_pr(a->l);
3207 +        rb_pr(a->r);
3208 +    }
3209 +}
3210 +
3211
3212 diff --git a/src/red_black.h b/src/red_black.h
3213 index fa661c7..b16d42f 100644
3214 --- a/src/red_black.h
3215 +++ b/src/red_black.h
3216 @@ -57,4 +57,6 @@ int rb_rot_dir(rb_n *r, rb_n *w);
3217 extern void (*rb_impr_elem)(rb_n *);
3218 int rb_set_a_impr_elem(void (*i)(rb_n *n));
3219
3220 +void rb_pr(rb_n *a);
3221 +
3222 #endif
3223
3224 commit e90e4e2fa485991daa656fb16ca4f655c898e815
3225 Author: Joilnen <joilnen@gmail.com>
3226 Date: Sat Dec 18 13:32:42 2021 -0300
3227
3228     adicionando doxygen para a documentacao auxiliar
3229
3230 diff --git a/.gitignore b/.gitignore
3231 index 15638f0..03aa2e7 100644
3232 --- a/.gitignore
3233 +++ b/.gitignore
3234 @@ -9,3 +9,5 @@ latex.out
3235 *.idx
3236 *.log
3237 *.out
3238 +doc/latex
3239 +doc/html
3240 diff --git a/Doxyfile b/Doxyfile
3241 new file mode 100644
3242 index 0000000..4f86eef

```

```

3243| --- /dev/null
3244| ++ b/Doxyfile
3245| @0 -0,0 +1,2642 @0
3246| # Doxyfile 1.9.2
3247| +
3248| # This file describes the settings to be used by the documentation system
3249| # doxygen (www.doxygen.org) for a project.
3250| #
3251| # All text after a double hash (##) is considered a comment and is placed in
3252| # front of the TAG it is preceding.
3253| #
3254| # All text after a single hash (#) is considered a comment and will be ignored.
3255| # The format is:
3256| # TAG = value [value, ...]
3257| # For lists, items can also be appended using:
3258| # TAG += value [value, ...]
3259| # Values that contain spaces should be placed between quotes (" ").
3260| +
3261| #-----
3262| # Project related configuration options
3263| #-----
3264| +
3265| # This tag specifies the encoding used for all characters in the configuration
3266| # file that follow. The default is UTF-8 which is also the encoding used for all
3267| # text before the first occurrence of this tag. Doxygen uses libiconv (or the
3268| # iconv built into libc) for the transcoding. See
3269| # https://www.gnu.org/software/libiconv/ for the list of possible encodings.
3270| # The default value is: UTF-8.
3271| +
3272| +DOXYFILE_ENCODING      = UTF-8
3273| +
3274| # The PROJECT_NAME tag is a single word (or a sequence of words surrounded by
3275| # double-quotes, unless you are using Doxywizard) that should identify the
3276| # project for which the documentation is generated. This name is used in the
3277| # title of most generated pages and in a few other places.
3278| # The default value is: My Project.
3279| +
3280| +PROJECT_NAME           = "Conjunto Ordenado"
3281| +
3282| # The PROJECT_NUMBER tag can be used to enter a project or revision number. This
3283| # could be handy for archiving the generated documentation or if some version
3284| # control system is used.
3285| +
3286| +PROJECT_NUMBER         =
3287| +
3288| # Using the PROJECT_BRIEF tag one can provide an optional one line description
3289| # for a project that appears at the top of each page and should give viewer a
3290| # quick idea about the purpose of the project. Keep the description short.
3291| +
3292| +PROJECT_BRIEF          =
3293| +
3294| # With the PROJECT_LOGO tag one can specify a logo or an icon that is included
3295| # in the documentation. The maximum height of the logo should not exceed 65
3296| # pixels and the maximum width should not exceed 200 pixels. Doxygen will copy
3297| # the logo to the output directory.
3298| +
3299| +PROJECT_LOGO           =
3300| +
3301| # The OUTPUT_DIRECTORY tag is used to specify the (relative or absolute) path
3302| # into which the generated documentation will be written. If a relative path is
3303| # entered, it will be relative to the location where doxygen was started. If
3304| # left blank the current directory will be used.
3305| +
3306| +OUTPUT_DIRECTORY       = doc
3307| +
3308| # If the CREATE_SUBDIRS tag is set to YES then doxygen will create 4096 sub-
3309| # directories (in 2 levels) under the output directory of each output format and
3310| # will distribute the generated files over these directories. Enabling this
3311| # option can be useful when feeding doxygen a huge amount of source files, where
3312| # putting all generated files in the same directory would otherwise causes
3313| # performance problems for the file system.
3314| # The default value is: NO.
3315| +
3316| +CREATE_SUBDIRS         = NO
3317| +
3318| # If the ALLOW_UNICODE_NAMES tag is set to YES, doxygen will allow non-ASCII
3319| # characters to appear in the names of generated files. If set to NO, non-ASCII
3320| # characters will be escaped, for example _xE3_x81_x84 will be used for Unicode
3321| # U+3044.
3322| # The default value is: NO.
3323| +
3324| +ALLOW_UNICODE_NAMES    = NO
3325| +
3326| # The OUTPUT_LANGUAGE tag is used to specify the language in which all
3327| # documentation generated by doxygen is written. Doxygen will use this
3328| # information to generate all constant output in the proper language.
3329| # Possible values are: Afrikaans, Arabic, Armenian, Brazilian, Catalan, Chinese,
3330| # Chinese-Traditional, Croatian, Czech, Danish, Dutch, English (United States),
3331| # Esperanto, Farsi (Persian), Finnish, French, German, Greek, Hungarian,
3332| # Indonesian, Italian, Japanese, Japanese-en (Japanese with English messages),
3333| # Korean, Korean-en (Korean with English messages), Latvian, Lithuanian,
3334| # Macedonian, Norwegian, Persian (Farsi), Polish, Portuguese, Romanian, Russian,
3335| # Serbian, Serbian-Cyrillic, Slovak, Slovene, Spanish, Swedish, Turkish,
3336| # Ukrainian and Vietnamese.
3337| # The default value is: English.
3338| +
3339| +OUTPUT_LANGUAGE        = Portuguese
3340| +
3341| # If the BRIEF_MEMBER_DESC tag is set to YES, doxygen will include brief member
3342| # descriptions after the members that are listed in the file and class
3343| # documentation (similar to Javadoc). Set to NO to disable this.
3344| # The default value is: YES.
3345| +
3346| +BRIEF_MEMBER_DESC      = YES
3347| +
3348| # If the REPEAT_BRIEF tag is set to YES, doxygen will prepend the brief
3349| # description of a member or function before the detailed description
3350| +
3351| # Note: If both HIDE_UNDOC_MEMBERS and BRIEF_MEMBER_DESC are set to NO, the
3352| # brief descriptions will be completely suppressed.
3353| # The default value is: YES.
3354| +
3355| +REPEAT_BRIEF           = YES
3356| +
3357| # This tag implements a quasi-intelligent brief description abbreviator that is
3358| # used to form the text in various listings. Each string in this list, if found

```

```

3359  + # as the leading text of the brief description, will be stripped from the text
3360  + # and the result, after processing the whole list, is used as the annotated
3361  + # text. Otherwise, the brief description is used as-is. If left blank, the
3362  + # following values are used ($name is automatically replaced with the name of
3363  + # the entity): The $name class, The $name widget, The $name file, is, provides,
3364  + # specifies, contains, represents, a, an and the.
3365  +
3366  + ABBREVIATE_BRIEF      = "The $name class" \
3367  +                        "The $name widget" \
3368  +                        "The $name file" \
3369  +                        is \
3370  +                        provides \
3371  +                        specifies \
3372  +                        contains \
3373  +                        represents \
3374  +                        a \
3375  +                        an \
3376  +                        the
3377  +
3378  + # If the ALWAYS_DETAILED_SEC and REPEAT_BRIEF tags are both set to YES then
3379  + # doxygen will generate a detailed section even if there is only a brief
3380  + # description.
3381  + # The default value is: NO.
3382  +
3383  + ALWAYS_DETAILED_SEC  = NO
3384  +
3385  + # If the INLINE_INHERITED_MEMB tag is set to YES, doxygen will show all
3386  + # inherited members of a class in the documentation of that class as if those
3387  + # members were ordinary class members. Constructors, destructors and assignment
3388  + # operators of the base classes will not be shown.
3389  + # The default value is: NO.
3390  +
3391  + INLINE_INHERITED_MEMB = NO
3392  +
3393  + # If the FULL_PATH_NAMES tag is set to YES, doxygen will prepend the full path
3394  + # before files name in the file list and in the header files. If set to NO the
3395  + # shortest path that makes the file name unique will be used
3396  + # The default value is: YES.
3397  +
3398  + FULL_PATH_NAMES      = YES
3399  +
3400  + # The STRIP_FROM_PATH tag can be used to strip a user-defined part of the path.
3401  + # Stripping is only done if one of the specified strings matches the left-hand
3402  + # part of the path. The tag can be used to show relative paths in the file list.
3403  + # If left blank the directory from which doxygen is run is used as the path to
3404  + # strip.
3405  +
3406  + # Note that you can specify absolute paths here, but also relative paths, which
3407  + # will be relative from the directory where doxygen is started.
3408  + # This tag requires that the tag FULL_PATH_NAMES is set to YES.
3409  +
3410  + STRIP_FROM_PATH       =
3411  +
3412  + # The STRIP_FROM_INC_PATH tag can be used to strip a user-defined part of the
3413  + # path mentioned in the documentation of a class, which tells the reader which
3414  + # header file to include in order to use a class. If left blank only the name of
3415  + # the header file containing the class definition is used. Otherwise one should
3416  + # specify the list of include paths that are normally passed to the compiler
3417  + # using the -I flag.
3418  +
3419  + STRIP_FROM_INC_PATH   =
3420  +
3421  + # If the SHORT_NAMES tag is set to YES, doxygen will generate much shorter (but
3422  + # less readable) file names. This can be useful if your file systems doesn't
3423  + # support long names like on DOS, Mac, or CD-ROM.
3424  + # The default value is: NO.
3425  +
3426  + SHORT_NAMES           = NO
3427  +
3428  + # If the JAVADOC_AUTOBRIEF tag is set to YES then doxygen will interpret the
3429  + # first line (until the first dot) of a Javadoc-style comment as the brief
3430  + # description. If set to NO, the Javadoc-style will behave just like regular Qt-
3431  + # style comments (thus requiring an explicit @brief command for a brief
3432  + # description.)
3433  + # The default value is: NO.
3434  +
3435  + JAVADOC_AUTOBRIEF     = NO
3436  +
3437  + # If the JAVADOC_BANNER tag is set to YES then doxygen will interpret a line
3438  + # such as
3439  + # /*****
3440  + # as being the beginning of a Javadoc-style comment "banner". If set to NO, the
3441  + # Javadoc-style will behave just like regular comments and it will not be
3442  + # interpreted by doxygen.
3443  + # The default value is: NO.
3444  +
3445  + JAVADOC_BANNER        = NO
3446  +
3447  + # If the QT_AUTOBRIEF tag is set to YES then doxygen will interpret the first
3448  + # line (until the first dot) of a Qt-style comment as the brief description. If
3449  + # set to NO, the Qt-style will behave just like regular Qt-style comments (thus
3450  + # requiring an explicit \brief command for a brief description.)
3451  + # The default value is: NO.
3452  +
3453  + QT_AUTOBRIEF          = NO
3454  +
3455  + # The MULTILINE_CPP_IS_BRIEF tag can be set to YES to make doxygen treat a
3456  + # multi-line C++ special comment block (i.e. a block of /*! or ///  

3457  + # a brief description. This used to be the default behavior. The new default is
3458  + # to treat a multi-line C++ comment block as a detailed description. Set this
3459  + # tag to YES if you prefer the old behavior instead.
3460  +
3461  + # Note that setting this tag to YES also means that rational rose comments are
3462  + # not recognized any more.
3463  + # The default value is: NO.
3464  +
3465  + MULTILINE_CPP_IS_BRIEF = NO
3466  +
3467  + # By default Python docstrings are displayed as preformatted text and doxygen's
3468  + # special commands cannot be used. By setting PYTHON_DOCSTRING to NO the
3469  + # doxygen's special commands can be used and the contents of the docstring
3470  + # documentation blocks is shown as doxygen documentation.
3471  + # The default value is: YES.
3472  +
3473  + PYTHON_DOCSTRING      = NO
3474  +

```

```

3475|## If the INHERIT_DOCS tag is set to YES then an undocumented member inherits the
3476|## documentation from any documented member that it re-implements.
3477|## The default value is: YES.
3478|+
3479|+INHERIT_DOCS          = YES
3480|+
3481|## If the SEPARATE_MEMBER_PAGES tag is set to YES then doxygen will produce a new
3482|## page for each member. If set to NO, the documentation of a member will be part
3483|## of the file/class/namespace that contains it.
3484|## The default value is: NO.
3485|+
3486|+SEPARATE_MEMBER_PAGES = NO
3487|+
3488|## The TAB_SIZE tag can be used to set the number of spaces in a tab. Doxygen
3489|## uses this value to replace tabs by spaces in code fragments.
3490|## Minimum value: 1, maximum value: 16, default value: 4.
3491|+
3492|+TAB_SIZE              = 4
3493|+
3494|## This tag can be used to specify a number of aliases that act as commands in
3495|## the documentation. An alias has the form:
3496|## name=value
3497|## For example adding
3498|## "sideeffect=@par Side Effects:^^"
3499|## will allow you to put the command \sideeffect (or @sideeffect) in the
3500|## documentation, which will result in a user-defined paragraph with heading
3501|## "Side Effects:". Note that you cannot put \n's in the value part of an alias
3502|## to insert newlines (in the resulting output). You can put ^^ in the value part
3503|## of an alias to insert a newline as if a physical newline was in the original
3504|## file. When you need a literal { or } or , in the value part of an alias you
3505|## have to escape them by means of a backslash (\), this can lead to conflicts
3506|## with the commands \{ and \} for these it is advised to use the version @{ and
3507|## @} or use a double escape (\{ and \})
3508|+
3509|+ALIASES               =
3510|+
3511|## Set the OPTIMIZE_OUTPUT_FOR_C tag to YES if your project consists of C sources
3512|## only. Doxygen will then generate output that is more tailored for C. For
3513|## instance, some of the names that are used will be different. The list of all
3514|## members will be omitted, etc.
3515|## The default value is: NO.
3516|+
3517|+OPTIMIZE_OUTPUT_FOR_C = YES
3518|+
3519|## Set the OPTIMIZE_OUTPUT_JAVA tag to YES if your project consists of Java or
3520|## Python sources only. Doxygen will then generate output that is more tailored
3521|## for that language. For instance, namespaces will be presented as packages,
3522|## qualified scopes will look different, etc.
3523|## The default value is: NO.
3524|+
3525|+OPTIMIZE_OUTPUT_JAVA  = NO
3526|+
3527|## Set the OPTIMIZE_FOR_FORTRAN tag to YES if your project consists of Fortran
3528|## sources. Doxygen will then generate output that is tailored for Fortran.
3529|## The default value is: NO.
3530|+
3531|+OPTIMIZE_FOR_FORTRAN  = NO
3532|+
3533|## Set the OPTIMIZE_OUTPUT_VHDL tag to YES if your project consists of VHDL
3534|## sources. Doxygen will then generate output that is tailored for VHDL.
3535|## The default value is: NO.
3536|+
3537|+OPTIMIZE_OUTPUT_VHDL  = NO
3538|+
3539|## Set the OPTIMIZE_OUTPUT_SLICE tag to YES if your project consists of Slice
3540|## sources only. Doxygen will then generate output that is more tailored for that
3541|## language. For instance, namespaces will be presented as modules, types will be
3542|## separated into more groups, etc.
3543|## The default value is: NO.
3544|+
3545|+OPTIMIZE_OUTPUT_SLICE = NO
3546|+
3547|## Doxygen selects the parser to use depending on the extension of the files it
3548|## parses. With this tag you can assign which parser to use for a given
3549|## extension. Doxygen has a built-in mapping, but you can override or extend it
3550|## using this tag. The format is ext=language, where ext is a file extension, and
3551|## language is one of the parsers supported by doxygen: IDL, Java, JavaScript,
3552|## Csharp (C#), C, C++, Lex, D, PHP, md (Markdown), Objective-C, Python, Slice,
3553|## VHDL, Fortran (fixed format Fortran: FortranFixed, free formatted Fortran:
3554|## FortranFree, unknown formatted Fortran: Fortran. In the later case the parser
3555|## tries to guess whether the code is fixed or free formatted code, this is the
3556|## default for Fortran type files). For instance to make doxygen treat .inc files
3557|## as Fortran files (default is PHP), and .f files as C (default is Fortran),
3558|## use: inc=Fortran f=C.
3559|+
3560|## Note: For files without extension you can use no_extension as a placeholder.
3561|+
3562|## Note that for custom extensions you also need to set FILE_PATTERNS otherwise
3563|## the files are not read by doxygen. When specifying no_extension you should add
3564|## * to the FILE_PATTERNS.
3565|+
3566|## Note see also the list of default file extension mappings.
3567|+
3568|+EXTENSION_MAPPING     =
3569|+
3570|## If the MARKDOWN_SUPPORT tag is enabled then doxygen pre-processes all comments
3571|## according to the Markdown format, which allows for more readable
3572|## documentation. See https://daringfireball.net/projects/markdown/ for details.
3573|## The output of markdown processing is further processed by doxygen, so you can
3574|## mix doxygen, HTML, and XML commands with Markdown formatting. Disable only in
3575|## case of backward compatibilities issues.
3576|## The default value is: YES.
3577|+
3578|+MARKDOWN_SUPPORT      = YES
3579|+
3580|## When the TOC_INCLUDE_HEADINGS tag is set to a non-zero value, all headings up
3581|## to that level are automatically included in the table of contents, even if
3582|## they do not have an id attribute.
3583|## Note: This feature currently applies only to Markdown headings.
3584|## Minimum value: 0, maximum value: 99, default value: 5.
3585|## This tag requires that the tag MARKDOWN_SUPPORT is set to YES.
3586|+
3587|+TOC_INCLUDE_HEADINGS  = 5
3588|+
3589|## When enabled doxygen tries to link words that correspond to documented
3590|## classes, or namespaces to their corresponding documentation. Such a link can

```



```

3591  ## be prevented in individual cases by putting a % sign in front of the word or
3592  ## globally by setting AUTOLINK_SUPPORT to NO.
3593  ## The default value is: YES.
3594  +
3595  +AUTOLINK_SUPPORT      = YES
3596  +
3597  ## If you use STL classes (i.e. std::string, std::vector, etc.) but do not want
3598  ## to include (a tag file for) the STL sources as input, then you should set this
3599  ## tag to YES in order to let doxygen match functions declarations and
3600  ## definitions whose arguments contain STL classes (e.g. func(std::string);
3601  ## versus func(std::string) {}). This also make the inheritance and collaboration
3602  ## diagrams that involve STL classes more complete and accurate.
3603  ## The default value is: NO.
3604  +
3605  +BUILTIN_STL_SUPPORT   = NO
3606  +
3607  ## If you use Microsoft's C++/CLI language, you should set this option to YES to
3608  ## enable parsing support.
3609  ## The default value is: NO.
3610  +
3611  +CPP_CLI_SUPPORT       = NO
3612  +
3613  ## Set the SIP_SUPPORT tag to YES if your project consists of sip (see:
3614  ## https://www.riverbankcomputing.com/software/sip/intro) sources only. Doxygen
3615  ## will parse them like normal C++ but will assume all classes use public instead
3616  ## of private inheritance when no explicit protection keyword is present.
3617  ## The default value is: NO.
3618  +
3619  +SIP_SUPPORT           = NO
3620  +
3621  ## For Microsoft's IDL there are proppget and propput attributes to indicate
3622  ## getter and setter methods for a property. Setting this option to YES will make
3623  ## doxygen to replace the get and set methods by a property in the documentation.
3624  ## This will only work if the methods are indeed getting or setting a simple
3625  ## type. If this is not the case, or you want to show the methods anyway, you
3626  ## should set this option to NO.
3627  ## The default value is: YES.
3628  +
3629  +IDL_PROPERTY_SUPPORT  = YES
3630  +
3631  ## If member grouping is used in the documentation and the DISTRIBUTE_GROUP_DOC
3632  ## tag is set to YES then doxygen will reuse the documentation of the first
3633  ## member in the group (if any) for the other members of the group. By default
3634  ## all members of a group must be documented explicitly.
3635  ## The default value is: NO.
3636  +
3637  +DISTRIBUTE_GROUP_DOC  = NO
3638  +
3639  ## If one adds a struct or class to a group and this option is enabled, then also
3640  ## any nested class or struct is added to the same group. By default this option
3641  ## is disabled and one has to add nested compounds explicitly via \ingroup.
3642  ## The default value is: NO.
3643  +
3644  +GROUP_NESTED_COMPOUNDS = NO
3645  +
3646  ## Set the SUBGROUPING tag to YES to allow class member groups of the same type
3647  ## (for instance a group of public functions) to be put as a subgroup of that
3648  ## type (e.g. under the Public Functions section). Set it to NO to prevent
3649  ## subgrouping. Alternatively, this can be done per class using the
3650  ## \nosubgrouping command.
3651  ## The default value is: YES.
3652  +
3653  +SUBGROUPING           = YES
3654  +
3655  ## When the INLINE_GROUPED_CLASSES tag is set to YES, classes, structs and unions
3656  ## are shown inside the group in which they are included (e.g. using \ingroup)
3657  ## instead of on a separate page (for HTML and Man pages) or section (for LaTeX
3658  ## and RTF).
3659  +
3660  ## Note that this feature does not work in combination with
3661  ## SEPARATE_MEMBER_PAGES.
3662  ## The default value is: NO.
3663  +
3664  +INLINE_GROUPED_CLASSES = NO
3665  +
3666  ## When the INLINE_SIMPLE_STRUCTS tag is set to YES, structs, classes, and unions
3667  ## with only public data fields or simple typedef fields will be shown inline in
3668  ## the documentation of the scope in which they are defined (i.e. file,
3669  ## namespace, or group documentation), provided this scope is documented. If set
3670  ## to NO, structs, classes, and unions are shown on a separate page (for HTML and
3671  ## Man pages) or section (for LaTeX and RTF).
3672  ## The default value is: NO.
3673  +
3674  +INLINE_SIMPLE_STRUCTS  = NO
3675  +
3676  ## When TYPEDEF_HIDES_STRUCT tag is enabled, a typedef of a struct, union, or
3677  ## enum is documented as struct, union, or enum with the name of the typedef. So
3678  ## typedef struct TypeS {} TypeT, will appear in the documentation as a struct
3679  ## with name TypeT. When disabled the typedef will appear as a member of a file,
3680  ## namespace, or class. And the struct will be named TypeS. This can typically be
3681  ## useful for C code in case the coding convention dictates that all compound
3682  ## types are typedef'ed and only the typedef is referenced, never the tag name.
3683  ## The default value is: NO.
3684  +
3685  +TYPEDEF_HIDES_STRUCT   = NO
3686  +
3687  ## The size of the symbol lookup cache can be set using LOOKUP_CACHE_SIZE. This
3688  ## cache is used to resolve symbols given their name and scope. Since this can be
3689  ## an expensive process and often the same symbol appears multiple times in the
3690  ## code, doxygen keeps a cache of pre-resolved symbols. If the cache is too small
3691  ## doxygen will become slower. If the cache is too large, memory is wasted. The
3692  ## cache size is given by this formula: 2(16+LOOKUP_CACHE_SIZE). The valid range
3693  ## is 0..9, the default is 0, corresponding to a cache size of 216=65536
3694  ## symbols. At the end of a run doxygen will report the cache usage and suggest
3695  ## the optimal cache size from a speed point of view.
3696  ## Minimum value: 0, maximum value: 9, default value: 0.
3697  +
3698  +LOOKUP_CACHE_SIZE      = 0
3699  +
3700  ## The NUM_PROC_THREADS specifies the number threads doxygen is allowed to use
3701  ## during processing. When set to 0 doxygen will based this on the number of
3702  ## cores available in the system. You can set it explicitly to a value larger
3703  ## than 0 to get more control over the balance between CPU load and processing
3704  ## speed. At this moment only the input processing can be done using multiple
3705  ## threads. Since this is still an experimental feature the default is set to 1,
3706  ## which effectively disables parallel processing. Please report any issues you

```

```

3707 | # encounter. Generating dot graphs in parallel is controlled by the
3708 | # DOT_NUM_THREADS setting.
3709 | # Minimum value: 0, maximum value: 32, default value: 1.
3710 | +
3711 | +NUM_PROC_THREADS      = 1
3712 | +
3713 | #-----
3714 | # Build related configuration options
3715 | #-----
3716 | +
3717 | # If the EXTRACT_ALL tag is set to YES, doxygen will assume all entities in
3718 | # documentation are documented, even if no documentation was available. Private
3719 | # class members and static file members will be hidden unless the
3720 | # EXTRACT_PRIVATE respectively EXTRACT_STATIC tags are set to YES.
3721 | # Note: This will also disable the warnings about undocumented members that are
3722 | # normally produced when WARNINGS is set to YES.
3723 | # The default value is: NO.
3724 | +
3725 | +EXTRACT_ALL           = YES
3726 | +
3727 | # If the EXTRACT_PRIVATE tag is set to YES, all private members of a class will
3728 | # be included in the documentation.
3729 | # The default value is: NO.
3730 | +
3731 | +EXTRACT_PRIVATE       = NO
3732 | +
3733 | # If the EXTRACT_PRIV_VIRTUAL tag is set to YES, documented private virtual
3734 | # methods of a class will be included in the documentation.
3735 | # The default value is: NO.
3736 | +
3737 | +EXTRACT_PRIV_VIRTUAL  = NO
3738 | +
3739 | # If the EXTRACT_PACKAGE tag is set to YES, all members with package or internal
3740 | # scope will be included in the documentation.
3741 | # The default value is: NO.
3742 | +
3743 | +EXTRACT_PACKAGE       = NO
3744 | +
3745 | # If the EXTRACT_STATIC tag is set to YES, all static members of a file will be
3746 | # included in the documentation.
3747 | # The default value is: NO.
3748 | +
3749 | +EXTRACT_STATIC        = NO
3750 | +
3751 | # If the EXTRACT_LOCAL_CLASSES tag is set to YES, classes (and structs) defined
3752 | # locally in source files will be included in the documentation. If set to NO,
3753 | # only classes defined in header files are included. Does not have any effect
3754 | # for Java sources.
3755 | # The default value is: YES.
3756 | +
3757 | +EXTRACT_LOCAL_CLASSES = YES
3758 | +
3759 | # This flag is only useful for Objective-C code. If set to YES, local methods,
3760 | # which are defined in the implementation section but not in the interface are
3761 | # included in the documentation. If set to NO, only methods in the interface are
3762 | # included.
3763 | # The default value is: NO.
3764 | +
3765 | +EXTRACT_LOCAL_METHODS = NO
3766 | +
3767 | # If this flag is set to YES, the members of anonymous namespaces will be
3768 | # extracted and appear in the documentation as a namespace called
3769 | # 'anonymous_namespace{file}', where file will be replaced with the base name of
3770 | # the file that contains the anonymous namespace. By default anonymous namespace
3771 | # are hidden.
3772 | # The default value is: NO.
3773 | +
3774 | +EXTRACT_ANON_NSPACES  = NO
3775 | +
3776 | # If this flag is set to YES, the name of an unnamed parameter in a declaration
3777 | # will be determined by the corresponding definition. By default unnamed
3778 | # parameters remain unnamed in the output.
3779 | # The default value is: YES.
3780 | +
3781 | +RESOLVE_UNNAMED_PARAMS = YES
3782 | +
3783 | # If the HIDE_UNDOC_MEMBERS tag is set to YES, doxygen will hide all
3784 | # undocumented members inside documented classes or files. If set to NO these
3785 | # members will be included in the various overviews, but no documentation
3786 | # section is generated. This option has no effect if EXTRACT_ALL is enabled.
3787 | # The default value is: NO.
3788 | +
3789 | +HIDE_UNDOC_MEMBERS    = NO
3790 | +
3791 | # If the HIDE_UNDOC_CLASSES tag is set to YES, doxygen will hide all
3792 | # undocumented classes that are normally visible in the class hierarchy. If set
3793 | # to NO, these classes will be included in the various overviews. This option
3794 | # has no effect if EXTRACT_ALL is enabled.
3795 | # The default value is: NO.
3796 | +
3797 | +HIDE_UNDOC_CLASSES    = NO
3798 | +
3799 | # If the HIDE_FRIEND_COMPOUNDS tag is set to YES, doxygen will hide all friend
3800 | # declarations. If set to NO, these declarations will be included in the
3801 | # documentation.
3802 | # The default value is: NO.
3803 | +
3804 | +HIDE_FRIEND_COMPOUNDS = NO
3805 | +
3806 | # If the HIDE_IN_BODY_DOCS tag is set to YES, doxygen will hide any
3807 | # documentation blocks found inside the body of a function. If set to NO, these
3808 | # blocks will be appended to the function's detailed documentation block.
3809 | # The default value is: NO.
3810 | +
3811 | +HIDE_IN_BODY_DOCS     = NO
3812 | +
3813 | # The INTERNAL_DOCS tag determines if documentation that is typed after a
3814 | # \internal command is included. If the tag is set to NO then the documentation
3815 | # will be excluded. Set it to YES to include the internal documentation.
3816 | # The default value is: NO.
3817 | +
3818 | +INTERNAL_DOCS         = NO
3819 | +
3820 | # With the correct setting of option CASE_SENSE_NAMES doxygen will better be
3821 | # able to match the capabilities of the underlying filesystem. In case the
3822 | # filesystem is case sensitive (i.e. it supports files in the same directory

```

```

3823  ## whose names only differ in casing), the option must be set to YES to properly
3824  ## deal with such files in case they appear in the input. For filesystems that
3825  ## are not case sensitive the option should be set to NO to properly deal with
3826  ## output files written for symbols that only differ in casing, such as for two
3827  ## classes, one named CLASS and the other named Class, and to also support
3828  ## references to files without having to specify the exact matching casing. On
3829  ## Windows (including Cygwin) and MacOS, users should typically set this option
3830  ## to NO, whereas on Linux or other Unix flavors it should typically be set to
3831  ## YES.
3832  ## The default value is: system dependent.
3833  +
3834  +CASE_SENSE_NAMES      = YES
3835  +
3836  ## If the HIDE_SCOPE_NAMES tag is set to NO then doxygen will show members with
3837  ## their full class and namespace scopes in the documentation. If set to YES, the
3838  ## scope will be hidden.
3839  ## The default value is: NO.
3840  +
3841  +HIDE_SCOPE_NAMES      = YES
3842  +
3843  ## If the HIDE_COMPOUND_REFERENCE tag is set to NO (default) then doxygen will
3844  ## append additional text to a page's title, such as Class Reference. If set to
3845  ## YES the compound reference will be hidden.
3846  ## The default value is: NO.
3847  +
3848  +HIDE_COMPOUND_REFERENCE = NO
3849  +
3850  ## If the SHOW_HEADERFILE tag is set to YES then the documentation for a class
3851  ## will show which file needs to be included to use the class.
3852  ## The default value is: YES.
3853  +
3854  +SHOW_HEADERFILE       = YES
3855  +
3856  ## If the SHOW_INCLUDE_FILES tag is set to YES then doxygen will put a list of
3857  ## the files that are included by a file in the documentation of that file.
3858  ## The default value is: YES.
3859  +
3860  +SHOW_INCLUDE_FILES    = YES
3861  +
3862  ## If the SHOW_GROUPED_MEMB_INC tag is set to YES then Doxygen will add for each
3863  ## grouped member an include statement to the documentation, telling the reader
3864  ## which file to include in order to use the member.
3865  ## The default value is: NO.
3866  +
3867  +SHOW_GROUPED_MEMB_INC = NO
3868  +
3869  ## If the FORCE_LOCAL_INCLUDES tag is set to YES then doxygen will list include
3870  ## files with double quotes in the documentation rather than with sharp brackets.
3871  ## The default value is: NO.
3872  +
3873  +FORCE_LOCAL_INCLUDES  = NO
3874  +
3875  ## If the INLINE_INFO tag is set to YES then a tag [inline] is inserted in the
3876  ## documentation for inline members.
3877  ## The default value is: YES.
3878  +
3879  +INLINE_INFO           = YES
3880  +
3881  ## If the SORT_MEMBER_DOCS tag is set to YES then doxygen will sort the
3882  ## (detailed) documentation of file and class members alphabetically by member
3883  ## name. If set to NO, the members will appear in declaration order.
3884  ## The default value is: YES.
3885  +
3886  +SORT_MEMBER_DOCS      = YES
3887  +
3888  ## If the SORT_BRIEF_DOCS tag is set to YES then doxygen will sort the brief
3889  ## descriptions of file, namespace and class members alphabetically by member
3890  ## name. If set to NO, the members will appear in declaration order. Note that
3891  ## this will also influence the order of the classes in the class list.
3892  ## The default value is: NO.
3893  +
3894  +SORT_BRIEF_DOCS       = NO
3895  +
3896  ## If the SORT_MEMBERS_CTORS_1ST tag is set to YES then doxygen will sort the
3897  ## (brief and detailed) documentation of class members so that constructors and
3898  ## destructors are listed first. If set to NO the constructors will appear in the
3899  ## respective orders defined by SORT_BRIEF_DOCS and SORT_MEMBER_DOCS.
3900  ## Note: If SORT_BRIEF_DOCS is set to NO this option is ignored for sorting brief
3901  ## member documentation.
3902  ## Note: If SORT_MEMBER_DOCS is set to NO this option is ignored for sorting
3903  ## detailed member documentation.
3904  ## The default value is: NO.
3905  +
3906  +SORT_MEMBERS_CTORS_1ST = NO
3907  +
3908  ## If the SORT_GROUP_NAMES tag is set to YES then doxygen will sort the hierarchy
3909  ## of group names into alphabetical order. If set to NO the group names will
3910  ## appear in their defined order.
3911  ## The default value is: NO.
3912  +
3913  +SORT_GROUP_NAMES      = NO
3914  +
3915  ## If the SORT_BY_SCOPE_NAME tag is set to YES, the class list will be sorted by
3916  ## fully-qualified names, including namespaces. If set to NO, the class list will
3917  ## be sorted only by class name, not including the namespace part.
3918  ## Note: This option is not very useful if HIDE_SCOPE_NAMES is set to YES.
3919  ## Note: This option applies only to the class list, not to the alphabetical
3920  ## list.
3921  ## The default value is: NO.
3922  +
3923  +SORT_BY_SCOPE_NAME    = NO
3924  +
3925  ## If the STRICT_PROTO_MATCHING option is enabled and doxygen fails to do proper
3926  ## type resolution of all parameters of a function it will reject a match between
3927  ## the prototype and the implementation of a member function even if there is
3928  ## only one candidate or it is obvious which candidate to choose by doing a
3929  ## simple string match. By disabling STRICT_PROTO_MATCHING doxygen will still
3930  ## accept a match between prototype and implementation in such cases.
3931  ## The default value is: NO.
3932  +
3933  +STRICT_PROTO_MATCHING = NO
3934  +
3935  ## The GENERATE_TODOLIST tag can be used to enable (YES) or disable (NO) the todo
3936  ## list. This list is created by putting \todo commands in the documentation.
3937  ## The default value is: YES.
3938  +

```

```

3939 +GENERATE_TODOLIST      = YES
3940 +
3941 ++ The GENERATE_TESTLIST tag can be used to enable (YES) or disable (NO) the test
3942 ++ list. This list is created by putting \test commands in the documentation.
3943 ++ The default value is: YES.
3944 +
3945 +GENERATE_TESTLIST      = YES
3946 +
3947 ++ The GENERATE_BUGLIST tag can be used to enable (YES) or disable (NO) the bug
3948 ++ list. This list is created by putting \bug commands in the documentation.
3949 ++ The default value is: YES.
3950 +
3951 +GENERATE_BUGLIST       = YES
3952 +
3953 ++ The GENERATE_DEPRECATEDLIST tag can be used to enable (YES) or disable (NO)
3954 ++ the deprecated list. This list is created by putting \deprecated commands in
3955 ++ the documentation.
3956 ++ The default value is: YES.
3957 +
3958 +GENERATE_DEPRECATEDLIST= YES
3959 +
3960 ++ The ENABLED_SECTIONS tag can be used to enable conditional documentation
3961 ++ sections, marked by \if <section_label> ... \endif and \cond <section_label>
3962 ++ ... \endcond blocks.
3963 +
3964 +ENABLED_SECTIONS       =
3965 +
3966 ++ The MAX_INITIALIZER_LINES tag determines the maximum number of lines that the
3967 ++ initial value of a variable or macro / define can have for it to appear in the
3968 ++ documentation. If the initializer consists of more lines than specified here
3969 ++ it will be hidden. Use a value of 0 to hide initializers completely. The
3970 ++ appearance of the value of individual variables and macros / defines can be
3971 ++ controlled using \showinitializer or \hideinitializer command in the
3972 ++ documentation regardless of this setting.
3973 ++ Minimum value: 0, maximum value: 10000, default value: 30.
3974 +
3975 +MAX_INITIALIZER_LINES  = 30
3976 +
3977 ++ Set the SHOW_USED_FILES tag to NO to disable the list of files generated at
3978 ++ the bottom of the documentation of classes and structs. If set to YES, the
3979 ++ list will mention the files that were used to generate the documentation.
3980 ++ The default value is: YES.
3981 +
3982 +SHOW_USED_FILES        = YES
3983 +
3984 ++ Set the SHOW_FILES tag to NO to disable the generation of the Files page. This
3985 ++ will remove the Files entry from the Quick Index and from the Folder Tree View
3986 ++ (if specified).
3987 ++ The default value is: YES.
3988 +
3989 +SHOW_FILES             = YES
3990 +
3991 ++ Set the SHOW_NAMESPACES tag to NO to disable the generation of the Namespaces
3992 ++ page. This will remove the Namespaces entry from the Quick Index and from the
3993 ++ Folder Tree View (if specified).
3994 ++ The default value is: YES.
3995 +
3996 +SHOW_NAMESPACES        = YES
3997 +
3998 ++ The FILE_VERSION_FILTER tag can be used to specify a program or script that
3999 ++ doxygen should invoke to get the current version for each file (typically from
4000 ++ the version control system). Doxygen will invoke the program by executing (via
4001 ++ popen()) the command command input-file, where command is the value of the
4002 ++ FILE_VERSION_FILTER tag, and input-file is the name of an input file provided
4003 ++ by doxygen. Whatever the program writes to standard output is used as the file
4004 ++ version. For an example see the documentation.
4005 +
4006 +FILE_VERSION_FILTER    =
4007 +
4008 ++ The LAYOUT_FILE tag can be used to specify a layout file which will be parsed
4009 ++ by doxygen. The layout file controls the global structure of the generated
4010 ++ output files in an output format independent way. To create the layout file
4011 ++ that represents doxygen's defaults, run doxygen with the -l option. You can
4012 ++ optionally specify a file name after the option, if omitted DoxygenLayout.xml
4013 ++ will be used as the name of the layout file. See also section "Changing the
4014 ++ layout of pages" for information.
4015 ++
4016 ++ Note that if you run doxygen from a directory containing a file called
4017 ++ DoxygenLayout.xml, doxygen will parse it automatically even if the LAYOUT_FILE
4018 ++ tag is left empty.
4019 +
4020 +LAYOUT_FILE            = DoxygenLayout.xml
4021 +
4022 ++ The CITE_BIB_FILES tag can be used to specify one or more bib files containing
4023 ++ the reference definitions. This must be a list of .bib files. The .bib
4024 ++ extension is automatically appended if omitted. This requires the bibtex tool
4025 ++ to be installed. See also https://en.wikipedia.org/wiki/BibTeX for more info.
4026 ++ For LaTeX the style of the bibliography can be controlled using
4027 ++ LATEX_BIB_STYLE. To use this feature you need bibtex and perl available in the
4028 ++ search path. See also \cite for info how to create references.
4029 +
4030 +CITE_BIB_FILES         =
4031 +
4032 +-----
4033 ++ Configuration options related to warning and progress messages
4034 +-----
4035 +
4036 ++ The QUIET tag can be used to turn on/off the messages that are generated to
4037 ++ standard output by doxygen. If QUIET is set to YES this implies that the
4038 ++ messages are off.
4039 ++ The default value is: NO.
4040 +
4041 +QUIET                  = NO
4042 +
4043 ++ The WARNINGS tag can be used to turn on/off the warning messages that are
4044 ++ generated to standard error (stderr) by doxygen. If WARNINGS is set to YES
4045 ++ this implies that the warnings are on.
4046 ++
4047 ++ Tip: Turn warnings on while writing the documentation.
4048 ++ The default value is: YES.
4049 +
4050 +WARNINGS               = YES
4051 +
4052 ++ If the WARN_IF_UNDOCUMENTED tag is set to YES then doxygen will generate
4053 ++ warnings for undocumented members. If EXTRACT_ALL is set to YES then this flag
4054 ++ will automatically be disabled.

```

```

4055|+# The default value is: YES.
4056|+
4057|+WARN_IF_UNDOCUMENTED = YES
4058|+
4059|+# If the WARN_IF_DOC_ERROR tag is set to YES, doxygen will generate warnings for
4060|+# potential errors in the documentation, such as documenting some parameters in
4061|+# a documented function twice, or documenting parameters that don't exist or
4062|+# using markup commands wrongly.
4063|+# The default value is: YES.
4064|+
4065|+WARN_IF_DOC_ERROR = YES
4066|+
4067|+# If WARN_IF_INCOMPLETE_DOC is set to YES, doxygen will warn about incomplete
4068|+# function parameter documentation. If set to NO, doxygen will accept that some
4069|+# parameters have no documentation without warning.
4070|+# The default value is: YES.
4071|+
4072|+WARN_IF_INCOMPLETE_DOC = YES
4073|+
4074|+# This WARN_NO_PARAMDOC option can be enabled to get warnings for functions that
4075|+# are documented, but have no documentation for their parameters or return
4076|+# value. If set to NO, doxygen will only warn about wrong parameter
4077|+# documentation, but not about the absence of documentation. If EXTRACT_ALL is
4078|+# set to YES then this flag will automatically be disabled. See also
4079|+# WARN_IF_INCOMPLETE_DOC
4080|+# The default value is: NO.
4081|+
4082|+WARN_NO_PARAMDOC = NO
4083|+
4084|+# If the WARN_AS_ERROR tag is set to YES then doxygen will immediately stop when
4085|+# a warning is encountered. If the WARN_AS_ERROR tag is set to FAIL_ON_WARNINGS
4086|+# then doxygen will continue running as if WARN_AS_ERROR tag is set to NO, but
4087|+# at the end of the doxygen process doxygen will return with a non-zero status.
4088|+# Possible values are: NO, YES and FAIL_ON_WARNINGS.
4089|+# The default value is: NO.
4090|+
4091|+WARN_AS_ERROR = NO
4092|+
4093|+# The WARN_FORMAT tag determines the format of the warning messages that doxygen
4094|+# can produce. The string should contain the $file, $line, and $text tags, which
4095|+# will be replaced by the file and line number from which the warning originated
4096|+# and the warning text. Optionally the format may contain $version, which will
4097|+# be replaced by the version of the file (if it could be obtained via
4098|+# FILE_VERSION_FILTER)
4099|+# The default value is: $file:$line: $text.
4100|+
4101|+WARN_FORMAT = "$file:$line: $text"
4102|+
4103|+# The WARN_LOGFILE tag can be used to specify a file to which warning and error
4104|+# messages should be written. If left blank the output is written to standard
4105|+# error (stderr).
4106|+
4107|+WARN_LOGFILE =
4108|+
4109|+-----
4110|+# Configuration options related to the input files
4111|+-----
4112|+
4113|+# The INPUT tag is used to specify the files and/or directories that contain
4114|+# documented source files. You may enter file names like myfile.cpp or
4115|+# directories like /usr/src/myproject. Separate the files or directories with
4116|+# spaces. See also FILE_PATTERNS and EXTENSION_MAPPING
4117|+# Note: If this tag is empty the current directory is searched.
4118|+
4119|+INPUT = src
4120|+
4121|+# This tag can be used to specify the character encoding of the source files
4122|+# that doxygen parses. Internally doxygen uses the UTF-8 encoding. Doxygen uses
4123|+# libiconv (or the iconv built into libc) for the transcoding. See the libiconv
4124|+# documentation (see:
4125|+# https://www.gnu.org/software/libiconv/) for the list of possible encodings.
4126|+# The default value is: UTF-8.
4127|+
4128|+INPUT_ENCODING = UTF-8
4129|+
4130|+# If the value of the INPUT tag contains directories, you can use the
4131|+# FILE_PATTERNS tag to specify one or more wildcard patterns (like *.cpp and
4132|+# *.h) to filter out the source-files in the directories.
4133|+
4134|+# Note that for custom extensions or not directly supported extensions you also
4135|+# need to set EXTENSION_MAPPING for the extension otherwise the files are not
4136|+# read by doxygen.
4137|+
4138|+# Note the list of default checked file patterns might differ from the list of
4139|+# default file extension mappings.
4140|+
4141|+# If left blank the following patterns are tested:*.c, *.cc, *.cxx, *.cpp,
4142|+# *.c++, *.java, *.ii, *.ixx, *.ipp, *.i++, *.inl, *.idl, *.ddl, *.odl, *.h,
4143|+# *.hh, *.hxx, *.hpp, *.h++, *.l, *.cs, *.d, *.php, *.php4, *.php5, *.phtml,
4144|+# *.inc, *.m, *.markdown, *.md, *.mm, *.dox (to be provided as doxygen C
4145|+# comment), *.py, *.pyw, *.f90, *.f95, *.f03, *.f08, *.f18, *.f, *.for, *.vhd,
4146|+# *.vhdl, *.ucf, *.qsf and *.ice.
4147|+
4148|+FILE_PATTERNS = *.c \
4149|+                *.cc \
4150|+                *.cxx \
4151|+                *.cpp \
4152|+                *.c++ \
4153|+                *.java \
4154|+                *.ii \
4155|+                *.ixx \
4156|+                *.ipp \
4157|+                *.i++ \
4158|+                *.inl \
4159|+                *.idl \
4160|+                *.ddl \
4161|+                *.odl \
4162|+                *.h \
4163|+                *.hh \
4164|+                *.hxx \
4165|+                *.hpp \
4166|+                *.h++ \
4167|+                *.l \
4168|+                *.cs \
4169|+                *.d \
4170|+                *.php \

```

```

4171 + *.php4 \
4172 + *.php5 \
4173 + *.phtml \
4174 + *.inc \
4175 + *.m \
4176 + *.markdown \
4177 + *.md \
4178 + *.mm \
4179 + *.dox \
4180 + *.py \
4181 + *.pyw \
4182 + *.f90 \
4183 + *.f95 \
4184 + *.f03 \
4185 + *.f08 \
4186 + *.f18 \
4187 + *.f \
4188 + *.for \
4189 + *.vhd \
4190 + *.vhd1 \
4191 + *.ucf \
4192 + *.qsf \
4193 + *.ice
4194 +
4195 ## The RECURSIVE tag can be used to specify whether or not subdirectories should
4196 ## be searched for input files as well.
4197 ## The default value is: NO.
4198 +
4199 +RECURSIVE = YES
4200 +
4201 ## The EXCLUDE tag can be used to specify files and/or directories that should be
4202 ## excluded from the INPUT source files. This way you can easily exclude a
4203 ## subdirectory from a directory tree whose root is specified with the INPUT tag.
4204 +
4205 ## Note that relative paths are relative to the directory from which doxygen is
4206 ## run.
4207 +
4208 +EXCLUDE =
4209 +
4210 ## The EXCLUDE_SYMLINKS tag can be used to select whether or not files or
4211 ## directories that are symbolic links (a Unix file system feature) are excluded
4212 ## from the input.
4213 ## The default value is: NO.
4214 +
4215 +EXCLUDE_SYMLINKS = NO
4216 +
4217 ## If the value of the INPUT tag contains directories, you can use the
4218 ## EXCLUDE_PATTERNS tag to specify one or more wildcard patterns to exclude
4219 ## certain files from those directories.
4220 +
4221 ## Note that the wildcards are matched against the file with absolute path, so to
4222 ## exclude all test directories for example use the pattern */test/*
4223 +
4224 +EXCLUDE_PATTERNS =
4225 +
4226 ## The EXCLUDE_SYMBOLS tag can be used to specify one or more symbol names
4227 ## (namespaces, classes, functions, etc.) that should be excluded from the
4228 ## output. The symbol name can be a fully qualified name, a word, or if the
4229 ## wildcard * is used, a substring. Examples: ANamespace, AClass,
4230 ## AClass::ANamespace, ANamespace::*Test
4231 +
4232 ## Note that the wildcards are matched against the file with absolute path, so to
4233 ## exclude all test directories use the pattern */test/*
4234 +
4235 +EXCLUDE_SYMBOLS =
4236 +
4237 ## The EXAMPLE_PATH tag can be used to specify one or more files or directories
4238 ## that contain example code fragments that are included (see the \include
4239 ## command).
4240 +
4241 +EXAMPLE_PATH =
4242 +
4243 ## If the value of the EXAMPLE_PATH tag contains directories, you can use the
4244 ## EXAMPLE_PATTERNS tag to specify one or more wildcard pattern (like *.cpp and
4245 ## *.h) to filter out the source-files in the directories. If left blank all
4246 ## files are included.
4247 +
4248 +EXAMPLE_PATTERNS = *
4249 +
4250 ## If the EXAMPLE_RECURSIVE tag is set to YES then subdirectories will be
4251 ## searched for input files to be used with the \include or \dontinclude commands
4252 ## irrespective of the value of the RECURSIVE tag.
4253 ## The default value is: NO.
4254 +
4255 +EXAMPLE_RECURSIVE = NO
4256 +
4257 ## The IMAGE_PATH tag can be used to specify one or more files or directories
4258 ## that contain images that are to be included in the documentation (see the
4259 ## \image command).
4260 +
4261 +IMAGE_PATH =
4262 +
4263 ## The INPUT_FILTER tag can be used to specify a program that doxygen should
4264 ## invoke to filter for each input file. Doxygen will invoke the filter program
4265 ## by executing (via popen()) the command:
4266 +
4267 ## <filter> <input-file>
4268 +
4269 ## where <filter> is the value of the INPUT_FILTER tag, and <input-file> is the
4270 ## name of an input file. Doxygen will then use the output that the filter
4271 ## program writes to standard output. If FILTER_PATTERNS is specified, this tag
4272 ## will be ignored.
4273 +
4274 ## Note that the filter must not add or remove lines; it is applied before the
4275 ## code is scanned, but not when the output code is generated. If lines are added
4276 ## or removed, the anchors will not be placed correctly.
4277 +
4278 ## Note that for custom extensions or not directly supported extensions you also
4279 ## need to set EXTENSION_MAPPING for the extension otherwise the files are not
4280 ## properly processed by doxygen.
4281 +
4282 +INPUT_FILTER =
4283 +
4284 ## The FILTER_PATTERNS tag can be used to specify filters on a per file pattern
4285 ## basis. Doxygen will compare the file name with each pattern and apply the
4286 ## filter if there is a match. The filters are a list of the form: pattern=filter

```



```

4287| ## (like *.cpp=my_cpp_filter). See INPUT_FILTER for further information on how
4288| ## filters are used. If the FILTER_PATTERNS tag is empty or if none of the
4289| ## patterns match the file name, INPUT_FILTER is applied.
4290| ##
4291| ## Note that for custom extensions or not directly supported extensions you also
4292| ## need to set EXTENSION_MAPPING for the extension otherwise the files are not
4293| ## properly processed by doxygen.
4294| +
4295| +FILTER_PATTERNS =
4296| +
4297| + If the FILTER_SOURCE_FILES tag is set to YES, the input filter (if set using
4298| + INPUT_FILTER) will also be used to filter the input files that are used for
4299| + producing the source files to browse (i.e. when SOURCE_BROWSER is set to YES).
4300| + The default value is: NO.
4301| +
4302| +FILTER_SOURCE_FILES = NO
4303| +
4304| + The FILTER_SOURCE_PATTERNS tag can be used to specify source filters per file
4305| + pattern. A pattern will override the setting for FILTER_PATTERN (if any) and
4306| + it is also possible to disable source filtering for a specific pattern using
4307| + *.ext= (so without naming a filter).
4308| + This tag requires that the tag FILTER_SOURCE_FILES is set to YES.
4309| +
4310| +FILTER_SOURCE_PATTERNS =
4311| +
4312| + If the USE_MDFILE_AS_MAINPAGE tag refers to the name of a markdown file that
4313| + is part of the input, its contents will be placed on the main page
4314| + (index.html). This can be useful if you have a project on for instance GitHub
4315| + and want to reuse the introduction page also for the doxygen output.
4316| +
4317| +USE_MDFILE_AS_MAINPAGE =
4318| +
4319| +-----
4320| + Configuration options related to source browsing
4321| +-----
4322| +
4323| + If the SOURCE_BROWSER tag is set to YES then a list of source files will be
4324| + generated. Documented entities will be cross-referenced with these sources.
4325| +
4326| + Note: To get rid of all source code in the generated output, make sure that
4327| + also VERBATIM_HEADERS is set to NO.
4328| + The default value is: NO.
4329| +
4330| +SOURCE_BROWSER = YES
4331| +
4332| + Setting the INLINE_SOURCES tag to YES will include the body of functions,
4333| + classes and enums directly into the documentation.
4334| + The default value is: NO.
4335| +
4336| +INLINE_SOURCES = NO
4337| +
4338| + Setting the STRIP_CODE_COMMENTS tag to YES will instruct doxygen to hide any
4339| + special comment blocks from generated source code fragments. Normal C, C++ and
4340| + Fortran comments will always remain visible.
4341| + The default value is: YES.
4342| +
4343| +STRIP_CODE_COMMENTS = YES
4344| +
4345| + If the REFERENCED_BY_RELATION tag is set to YES then for each documented
4346| + entity all documented functions referencing it will be listed.
4347| + The default value is: NO.
4348| +
4349| +REFERENCED_BY_RELATION = NO
4350| +
4351| + If the REFERENCES_RELATION tag is set to YES then for each documented function
4352| + all documented entities called/used by that function will be listed.
4353| + The default value is: NO.
4354| +
4355| +REFERENCES_RELATION = NO
4356| +
4357| + If the REFERENCES_LINK_SOURCE tag is set to YES and SOURCE_BROWSER tag is set
4358| + to YES then the hyperlinks from functions in REFERENCES_RELATION and
4359| + REFERENCED_BY_RELATION lists will link to the source code. Otherwise they will
4360| + link to the documentation.
4361| + The default value is: YES.
4362| +
4363| +REFERENCES_LINK_SOURCE = YES
4364| +
4365| + If SOURCE_TOOLTIPS is enabled (the default) then hovering a hyperlink in the
4366| + source code will show a tooltip with additional information such as prototype,
4367| + brief description and links to the definition and documentation. Since this
4368| + will make the HTML file larger and loading of large files a bit slower, you
4369| + can opt to disable this feature.
4370| + The default value is: YES.
4371| + This tag requires that the tag SOURCE_BROWSER is set to YES.
4372| +
4373| +SOURCE_TOOLTIPS = YES
4374| +
4375| + If the USE_HTAGS tag is set to YES then the references to source code will
4376| + point to the HTML generated by the htags(1) tool instead of doxygen built-in
4377| + source browser. The htags tool is part of GNU's global source tagging system
4378| + (see https://www.gnu.org/software/global/global.html). You will need version
4379| + 4.8.6 or higher.
4380| +
4381| + To use it do the following:
4382| + - Install the latest version of global
4383| + - Enable SOURCE_BROWSER and USE_HTAGS in the configuration file
4384| + - Make sure the INPUT points to the root of the source tree
4385| + - Run doxygen as normal
4386| +
4387| + Doxygen will invoke htags (and that will in turn invoke gtags), so these
4388| + tools must be available from the command line (i.e. in the search path).
4389| +
4390| + The result: instead of the source browser generated by doxygen, the links to
4391| + source code will now point to the output of htags.
4392| + The default value is: NO.
4393| + This tag requires that the tag SOURCE_BROWSER is set to YES.
4394| +
4395| +USE_HTAGS = NO
4396| +
4397| + If the VERBATIM_HEADERS tag is set the YES then doxygen will generate a
4398| + verbatim copy of the header file for each class for which an include is
4399| + specified. Set to NO to disable this.
4400| + See also: Section \class
4401| + The default value is: YES.
4402| +

```

```

4403 +VERBATIM_HEADERS      = YES
4404 +
4405 +-----
4406 +# Configuration options related to the alphabetical class index
4407 +-----
4408 +
4409 +# If the ALPHABETICAL_INDEX tag is set to YES, an alphabetical index of all
4410 +# compounds will be generated. Enable this if the project contains a lot of
4411 +# classes, structs, unions or interfaces.
4412 +# The default value is: YES.
4413 +
4414 +ALPHABETICAL_INDEX      = YES
4415 +
4416 +# In case all classes in a project start with a common prefix, all classes will
4417 +# be put under the same header in the alphabetical index. The IGNORE_PREFIX tag
4418 +# can be used to specify a prefix (or a list of prefixes) that should be ignored
4419 +# while generating the index headers.
4420 +# This tag requires that the tag ALPHABETICAL_INDEX is set to YES.
4421 +
4422 +IGNORE_PREFIX           =
4423 +
4424 +-----
4425 +# Configuration options related to the HTML output
4426 +-----
4427 +
4428 +# If the GENERATE_HTML tag is set to YES, doxygen will generate HTML output
4429 +# The default value is: YES.
4430 +
4431 +GENERATE_HTML            = YES
4432 +
4433 +# The HTML_OUTPUT tag is used to specify where the HTML docs will be put. If a
4434 +# relative path is entered the value of OUTPUT_DIRECTORY will be put in front of
4435 +# it
4436 +# The default directory is: html.
4437 +# This tag requires that the tag GENERATE_HTML is set to YES.
4438 +
4439 +HTML_OUTPUT              = html
4440 +
4441 +# The HTML_FILE_EXTENSION tag can be used to specify the file extension for each
4442 +# generated HTML page (for example: .htm, .php, .asp).
4443 +# The default value is: .html.
4444 +# This tag requires that the tag GENERATE_HTML is set to YES.
4445 +
4446 +HTML_FILE_EXTENSION      = .html
4447 +
4448 +# The HTML_HEADER tag can be used to specify a user-defined HTML header file for
4449 +# each generated HTML page. If the tag is left blank doxygen will generate a
4450 +# standard header.
4451 +
4452 +# To get valid HTML the header file that includes any scripts and style sheets
4453 +# that doxygen needs, which is dependent on the configuration options used (e.g.
4454 +# the setting GENERATE_TREEVIEW). It is highly recommended to start with a
4455 +# default header using
4456 +# doxygen -w html new_header.html new_footer.html new_stylesheet.css
4457 +# YourConfigFile
4458 +# and then modify the file new_header.html. See also section "Doxygen usage"
4459 +# for information on how to generate the default header that doxygen normally
4460 +# uses.
4461 +# Note: The header is subject to change so you typically have to regenerate the
4462 +# default header when upgrading to a newer version of doxygen. For a description
4463 +# of the possible markers and block names see the documentation.
4464 +# This tag requires that the tag GENERATE_HTML is set to YES.
4465 +
4466 +HTML_HEADER              =
4467 +
4468 +# The HTML_FOOTER tag can be used to specify a user-defined HTML footer for each
4469 +# generated HTML page. If the tag is left blank doxygen will generate a standard
4470 +# footer. See HTML_HEADER for more information on how to generate a default
4471 +# footer and what special commands can be used inside the footer. See also
4472 +# section "Doxygen usage" for information on how to generate the default footer
4473 +# that doxygen normally uses.
4474 +# This tag requires that the tag GENERATE_HTML is set to YES.
4475 +
4476 +HTML_FOOTER              =
4477 +
4478 +# The HTML_STYLESHEET tag can be used to specify a user-defined cascading style
4479 +# sheet that is used by each HTML page. It can be used to fine-tune the look of
4480 +# the HTML output. If left blank doxygen will generate a default style sheet.
4481 +# See also section "Doxygen usage" for information on how to generate the style
4482 +# sheet that doxygen normally uses.
4483 +# Note: It is recommended to use HTML_EXTRA_STYLESHEET instead of this tag, as
4484 +# it is more robust and this tag (HTML_STYLESHEET) will in the future become
4485 +# obsolete.
4486 +# This tag requires that the tag GENERATE_HTML is set to YES.
4487 +
4488 +HTML_STYLESHEET          =
4489 +
4490 +# The HTML_EXTRA_STYLESHEET tag can be used to specify additional user-defined
4491 +# cascading style sheets that are included after the standard style sheets
4492 +# created by doxygen. Using this option one can overrule certain style aspects.
4493 +# This is preferred over using HTML_STYLESHEET since it does not replace the
4494 +# standard style sheet and is therefore more robust against future updates.
4495 +# Doxygen will copy the style sheet files to the output directory.
4496 +# Note: The order of the extra style sheet files is of importance (e.g. the last
4497 +# style sheet in the list overrules the setting of the previous ones in the
4498 +# list). For an example see the documentation.
4499 +# This tag requires that the tag GENERATE_HTML is set to YES.
4500 +
4501 +HTML_EXTRA_STYLESHEET    =
4502 +
4503 +# The HTML_EXTRA_FILES tag can be used to specify one or more extra images or
4504 +# other source files which should be copied to the HTML output directory. Note
4505 +# that these files will be copied to the base HTML output directory. Use the
4506 +# $relpath$ marker in the HTML_HEADER and/or HTML_FOOTER files to load these
4507 +# files. In the HTML_STYLESHEET file, use the file name only. Also note that the
4508 +# files will be copied as-is; there are no commands or markers available.
4509 +# This tag requires that the tag GENERATE_HTML is set to YES.
4510 +
4511 +HTML_EXTRA_FILES         =
4512 +
4513 +# The HTML_COLORSTYLE_HUE tag controls the color of the HTML output. Doxygen
4514 +# will adjust the colors in the style sheet and background images according to
4515 +# this color. Hue is specified as an angle on a color-wheel, see
4516 +# https://en.wikipedia.org/wiki/Hue for more information. For instance the value
4517 +# 0 represents red, 60 is yellow, 120 is green, 180 is cyan, 240 is blue, 300
4518 +# purple, and 360 is red again.

```



```

4519 | ## Minimum value: 0, maximum value: 359, default value: 220.
4520 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4521 | +
4522 | +HTML_COLORSTYLE_HUE      = 234
4523 | +
4524 | ## The HTML_COLORSTYLE_SAT tag controls the purity (or saturation) of the colors
4525 | ## in the HTML output. For a value of 0 the output will use gray-scales only. A
4526 | ## value of 255 will produce the most vivid colors.
4527 | ## Minimum value: 0, maximum value: 255, default value: 100.
4528 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4529 | +
4530 | +HTML_COLORSTYLE_SAT      = 48
4531 | +
4532 | ## The HTML_COLORSTYLE_GAMMA tag controls the gamma correction applied to the
4533 | ## luminance component of the colors in the HTML output. Values below 100
4534 | ## gradually make the output lighter, whereas values above 100 make the output
4535 | ## darker. The value divided by 100 is the actual gamma applied, so 80 represents
4536 | ## a gamma of 0.8. The value 220 represents a gamma of 2.2, and 100 does not
4537 | ## change the gamma.
4538 | ## Minimum value: 40, maximum value: 240, default value: 80.
4539 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4540 | +
4541 | +HTML_COLORSTYLE_GAMMA    = 158
4542 | +
4543 | ## If the HTML_TIMESTAMP tag is set to YES then the footer of each generated HTML
4544 | ## page will contain the date and time when the page was generated. Setting this
4545 | ## to YES can help to show when doxygen was last run and thus if the
4546 | ## documentation is up to date.
4547 | ## The default value is: NO.
4548 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4549 | +
4550 | +HTML_TIMESTAMP           = NO
4551 | +
4552 | ## If the HTML_DYNAMIC_MENUS tag is set to YES then the generated HTML
4553 | ## documentation will contain a main index with vertical navigation menus that
4554 | ## are dynamically created via JavaScript. If disabled, the navigation index will
4555 | ## consist of multiple levels of tabs that are statically embedded in every HTML
4556 | ## page. Disable this option to support browsers that do not have JavaScript,
4557 | ## like the Qt help browser.
4558 | ## The default value is: YES.
4559 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4560 | +
4561 | +HTML_DYNAMIC_MENUS       = YES
4562 | +
4563 | ## If the HTML_DYNAMIC_SECTIONS tag is set to YES then the generated HTML
4564 | ## documentation will contain sections that can be hidden and shown after the
4565 | ## page has loaded.
4566 | ## The default value is: NO.
4567 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4568 | +
4569 | +HTML_DYNAMIC_SECTIONS    = NO
4570 | +
4571 | ## With HTML_INDEX_NUM_ENTRIES one can control the preferred number of entries
4572 | ## shown in the various tree structured indices initially; the user can expand
4573 | ## and collapse entries dynamically later on. Doxygen will expand the tree to
4574 | ## such a level that at most the specified number of entries are visible (unless
4575 | ## a fully collapsed tree already exceeds this amount). So setting the number of
4576 | ## entries 1 will produce a full collapsed tree by default. 0 is a special value
4577 | ## representing an infinite number of entries and will result in a full expanded
4578 | ## tree by default.
4579 | ## Minimum value: 0, maximum value: 9999, default value: 100.
4580 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4581 | +
4582 | +HTML_INDEX_NUM_ENTRIES   = 100
4583 | +
4584 | ## If the GENERATE_DOCSET tag is set to YES, additional index files will be
4585 | ## generated that can be used as input for Apple's Xcode 3 integrated development
4586 | ## environment (see:
4587 | ## https://developer.apple.com/xcode/), introduced with OSX 10.5 (Leopard). To
4588 | ## create a documentation set, doxygen will generate a Makefile in the HTML
4589 | ## output directory. Running make will produce the docset in that directory and
4590 | ## running make install will install the docset in
4591 | ## ~/Library/Developer/Shared/Documentation/DocSets so that Xcode will find it at
4592 | ## startup. See https://developer.apple.com/library/archive/featuredarticles/Doxy
4593 | ## genXcode/_index.html for more information.
4594 | ## The default value is: NO.
4595 | ## This tag requires that the tag GENERATE_HTML is set to YES.
4596 | +
4597 | +GENERATE_DOCSET           = NO
4598 | +
4599 | ## This tag determines the name of the docset feed. A documentation feed provides
4600 | ## an umbrella under which multiple documentation sets from a single provider
4601 | ## (such as a company or product suite) can be grouped.
4602 | ## The default value is: Doxygen generated docs.
4603 | ## This tag requires that the tag GENERATE_DOCSET is set to YES.
4604 | +
4605 | +DOCSET_FEEDNAME           =
4606 | +
4607 | ## This tag specifies a string that should uniquely identify the documentation
4608 | ## set bundle. This should be a reverse domain-name style string, e.g.
4609 | ## com.mycompany.MyDocSet. Doxygen will append .docset to the name.
4610 | ## The default value is: org.doxygen.Project.
4611 | ## This tag requires that the tag GENERATE_DOCSET is set to YES.
4612 | +
4613 | +DOCSET_BUNDLE_ID          = org.doxygen.Project
4614 | +
4615 | ## The DOCSET_PUBLISHER_ID tag specifies a string that should uniquely identify
4616 | ## the documentation publisher. This should be a reverse domain-name style
4617 | ## string, e.g. com.mycompany.MyDocSet.documentation.
4618 | ## The default value is: org.doxygen.Publisher.
4619 | ## This tag requires that the tag GENERATE_DOCSET is set to YES.
4620 | +
4621 | +DOCSET_PUBLISHER_ID       = org.doxygen.Publisher
4622 | +
4623 | ## The DOCSET_PUBLISHER_NAME tag identifies the documentation publisher.
4624 | ## The default value is: Publisher.
4625 | ## This tag requires that the tag GENERATE_DOCSET is set to YES.
4626 | +
4627 | +DOCSET_PUBLISHER_NAME     = Publisher
4628 | +
4629 | ## If the GENERATE_HTMLHELP tag is set to YES then doxygen generates three
4630 | ## additional HTML index files: index.hpp, index.hhc, and index.hhk. The
4631 | ## index.hpp is a project file that can be read by Microsoft's HTML Help Workshop
4632 | ## on Windows. In the beginning of 2021 Microsoft took the original page, with
4633 | ## a.o. the download links, offline the HTML help workshop was already many years
4634 | ## in maintenance mode). You can download the HTML help workshop from the web

```

```

4635| ## archives at Installation executable (see:
4636| ## http://web.archive.org/web/20160201063255/http://download.microsoft.com/downlo
4637| ## ad/0/A/9/0A939EF6-E31C-430F-A3DF-DFAE7960D564/htmlhelp.exe).
4638| ##
4639| ## The HTML Help Workshop contains a compiler that can convert all HTML output
4640| ## generated by doxygen into a single compiled HTML file (.chm). Compiled HTML
4641| ## files are now used as the Windows 98 help format, and will replace the old
4642| ## Windows help format (.hlp) on all Windows platforms in the future. Compressed
4643| ## HTML files also contain an index, a table of contents, and you can search for
4644| ## words in the documentation. The HTML workshop also contains a viewer for
4645| ## compressed HTML files.
4646| ## The default value is: NO.
4647| ## This tag requires that the tag GENERATE_HTML is set to YES.
4648| +
4649| +GENERATE_HTMLHELP = NO
4650| +
4651| ## The CHM_FILE tag can be used to specify the file name of the resulting .chm
4652| ## file. You can add a path in front of the file if the result should not be
4653| ## written to the html output directory.
4654| ## This tag requires that the tag GENERATE_HTMLHELP is set to YES.
4655| +
4656| +CHM_FILE =
4657| +
4658| ## The HHC_LOCATION tag can be used to specify the location (absolute path
4659| ## including file name) of the HTML help compiler (hhc.exe). If non-empty,
4660| ## doxygen will try to run the HTML help compiler on the generated index.hhp.
4661| ## The file has to be specified with full path.
4662| ## This tag requires that the tag GENERATE_HTMLHELP is set to YES.
4663| +
4664| +HHC_LOCATION =
4665| +
4666| ## The GENERATE_CHI flag controls if a separate .chi index file is generated
4667| ## (YES) or that it should be included in the main .chm file (NO).
4668| ## The default value is: NO.
4669| ## This tag requires that the tag GENERATE_HTMLHELP is set to YES.
4670| +
4671| +GENERATE_CHI = NO
4672| +
4673| ## The CHM_INDEX_ENCODING is used to encode HtmlHelp index (hhk), content (hhc)
4674| ## and project file content.
4675| ## This tag requires that the tag GENERATE_HTMLHELP is set to YES.
4676| +
4677| +CHM_INDEX_ENCODING =
4678| +
4679| ## The BINARY_TOC flag controls whether a binary table of contents is generated
4680| ## (YES) or a normal table of contents (NO) in the .chm file. Furthermore it
4681| ## enables the Previous and Next buttons.
4682| ## The default value is: NO.
4683| ## This tag requires that the tag GENERATE_HTMLHELP is set to YES.
4684| +
4685| +BINARY_TOC = NO
4686| +
4687| ## The TOC_EXPAND flag can be set to YES to add extra items for group members to
4688| ## the table of contents of the HTML help documentation and to the tree view.
4689| ## The default value is: NO.
4690| ## This tag requires that the tag GENERATE_HTMLHELP is set to YES.
4691| +
4692| +TOC_EXPAND = NO
4693| +
4694| ## If the GENERATE_QHP tag is set to YES and both QHP_NAMESPACE and
4695| ## QHP_VIRTUAL_FOLDER are set, an additional index file will be generated that
4696| ## can be used as input for Qt's qhelpgenerator to generate a Qt Compressed Help
4697| ## (.qch) of the generated HTML documentation.
4698| ## The default value is: NO.
4699| ## This tag requires that the tag GENERATE_HTML is set to YES.
4700| +
4701| +GENERATE_QHP = NO
4702| +
4703| ## If the QHG_LOCATION tag is specified, the QCH_FILE tag can be used to specify
4704| ## the file name of the resulting .qch file. The path specified is relative to
4705| ## the HTML output folder.
4706| ## This tag requires that the tag GENERATE_QHP is set to YES.
4707| +
4708| +QCH_FILE =
4709| +
4710| ## The QHP_NAMESPACE tag specifies the namespace to use when generating Qt Help
4711| ## Project output. For more information please see Qt Help Project / Namespace
4712| ## (see:
4713| ## https://doc.qt.io/archives/qt-4.8/qthelpproject.html#namespace).
4714| ## The default value is: org.doxygen.Project.
4715| ## This tag requires that the tag GENERATE_QHP is set to YES.
4716| +
4717| +QHP_NAMESPACE = org.doxygen.Project
4718| +
4719| ## The QHP_VIRTUAL_FOLDER tag specifies the namespace to use when generating Qt
4720| ## Help Project output. For more information please see Qt Help Project / Virtual
4721| ## Folders (see:
4722| ## https://doc.qt.io/archives/qt-4.8/qthelpproject.html#virtual-folders).
4723| ## The default value is: doc.
4724| ## This tag requires that the tag GENERATE_QHP is set to YES.
4725| +
4726| +QHP_VIRTUAL_FOLDER = doc
4727| +
4728| ## If the QHP_CUST_FILTER_NAME tag is set, it specifies the name of a custom
4729| ## filter to add. For more information please see Qt Help Project / Custom
4730| ## Filters (see:
4731| ## https://doc.qt.io/archives/qt-4.8/qthelpproject.html#custom-filters).
4732| ## This tag requires that the tag GENERATE_QHP is set to YES.
4733| +
4734| +QHP_CUST_FILTER_NAME =
4735| +
4736| ## The QHP_CUST_FILTER_ATTRS tag specifies the list of the attributes of the
4737| ## custom filter to add. For more information please see Qt Help Project / Custom
4738| ## Filters (see:
4739| ## https://doc.qt.io/archives/qt-4.8/qthelpproject.html#custom-filters).
4740| ## This tag requires that the tag GENERATE_QHP is set to YES.
4741| +
4742| +QHP_CUST_FILTER_ATTRS =
4743| +
4744| ## The QHP_SECT_FILTER_ATTRS tag specifies the list of the attributes this
4745| ## project's filter section matches. Qt Help Project / Filter Attributes (see:
4746| ## https://doc.qt.io/archives/qt-4.8/qthelpproject.html#filter-attributes).
4747| ## This tag requires that the tag GENERATE_QHP is set to YES.
4748| +
4749| +QHP_SECT_FILTER_ATTRS =
4750| +

```

```

4751|+ # The QHG_LOCATION tag can be used to specify the location (absolute path
4752|+ # including file name) of Qt's qhelpgenerator. If non-empty doxygen will try to
4753|+ # run qhelpgenerator on the generated .qhp file.
4754|+ # This tag requires that the tag GENERATE_QHP is set to YES.
4755|+
4756|+QHLOCATION =
4757|+
4758|+ # If the GENERATE_ECLIPSEHELP tag is set to YES, additional index files will be
4759|+ # generated, together with the HTML files, they form an Eclipse help plugin. To
4760|+ # install this plugin and make it available under the help contents menu in
4761|+ # Eclipse, the contents of the directory containing the HTML and XML files needs
4762|+ # to be copied into the plugins directory of eclipse. The name of the directory
4763|+ # within the plugins directory should be the same as the ECLIPSE_DOC_ID value.
4764|+ # After copying Eclipse needs to be restarted before the help appears.
4765|+ # The default value is: NO
4766|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4767|+
4768|+GENERATE_ECLIPSEHELP = NO
4769|+
4770|+ # A unique identifier for the Eclipse help plugin. When installing the plugin
4771|+ # the directory name containing the HTML and XML files should also have this
4772|+ # name. Each documentation set should have its own identifier.
4773|+ # The default value is: org.doxygen.Project
4774|+ # This tag requires that the tag GENERATE_ECLIPSEHELP is set to YES.
4775|+
4776|+ECLIPSE_DOC_ID = org.doxygen.Project
4777|+
4778|+ # If you want full control over the layout of the generated HTML pages it might
4779|+ # be necessary to disable the index and replace it with your own. The
4780|+ # DISABLE_INDEX tag can be used to turn on/off the condensed index (tabs) at top
4781|+ # of each HTML page. A value of NO enables the index and the value YES disables
4782|+ # it. Since the tabs in the index contain the same information as the navigation
4783|+ # tree, you can set this option to YES if you also set GENERATE_TREEVIEW to YES.
4784|+ # The default value is: NO
4785|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4786|+
4787|+DISABLE_INDEX = NO
4788|+
4789|+ # The GENERATE_TREEVIEW tag is used to specify whether a tree-like index
4790|+ # structure should be generated to display hierarchical information. If the tag
4791|+ # value is set to YES, a side panel will be generated containing a tree-like
4792|+ # index structure (just like the one that is generated for HTML Help). For this
4793|+ # to work a browser that supports JavaScript, DHTML, CSS and frames is required
4794|+ # (i.e. any modern browser). Windows users are probably better off using the
4795|+ # HTML help feature. Via custom style sheets (see HTML_EXTRA_STYLESHEET) one can
4796|+ # further fine tune the look of the index (see "Fine-tuning the output"). As an
4797|+ # example, the default style sheet generated by doxygen has an example that
4798|+ # shows how to put an image at the root of the tree instead of the PROJECT_NAME.
4799|+ # Since the tree basically has the same information as the tab index, you could
4800|+ # consider setting DISABLE_INDEX to YES when enabling this option.
4801|+ # The default value is: NO
4802|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4803|+
4804|+GENERATE_TREEVIEW = NO
4805|+
4806|+ # When both GENERATE_TREEVIEW and DISABLE_INDEX are set to YES, then the
4807|+ # FULL_SIDEBAR option determines if the side bar is limited to only the treeview
4808|+ # area (value NO) or if it should extend to the full height of the window (value
4809|+ # YES). Setting this to YES gives a layout similar to
4810|+ # https://docs.readthedocs.io with more room for contents, but less room for the
4811|+ # project logo, title, and description. If either GENERATOR_TREEVIEW or
4812|+ # DISABLE_INDEX is set to NO, this option has no effect.
4813|+ # The default value is: NO
4814|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4815|+
4816|+FULL_SIDEBAR = NO
4817|+
4818|+ # The ENUM_VALUES_PER_LINE tag can be used to set the number of enum values that
4819|+ # doxygen will group on one line in the generated HTML documentation.
4820|+
4821|+ # Note that a value of 0 will completely suppress the enum values from appearing
4822|+ # in the overview section.
4823|+ # Minimum value: 0, maximum value: 20, default value: 4.
4824|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4825|+
4826|+ENUM_VALUES_PER_LINE = 4
4827|+
4828|+ # If the treeview is enabled (see GENERATE_TREEVIEW) then this tag can be used
4829|+ # to set the initial width (in pixels) of the frame in which the tree is shown.
4830|+ # Minimum value: 0, maximum value: 1500, default value: 250.
4831|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4832|+
4833|+TREEVIEW_WIDTH = 250
4834|+
4835|+ # If the EXT_LINKS_IN_WINDOW option is set to YES, doxygen will open links to
4836|+ # external symbols imported via tag files in a separate window.
4837|+ # The default value is: NO
4838|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4839|+
4840|+EXT_LINKS_IN_WINDOW = NO
4841|+
4842|+ # If the HTML_FORMULA_FORMAT option is set to svg, doxygen will use the pdf2svg
4843|+ # tool (see https://github.com/dawbarton/pdf2svg) or inkscape (see
4844|+ # https://inkscape.org) to generate formulas as SVG images instead of PNGs for
4845|+ # the HTML output. These images will generally look nicer at scaled resolutions.
4846|+ # Possible values are: png (the default) and svg (looks nicer but requires the
4847|+ # pdf2svg or inkscape tool).
4848|+ # The default value is: png.
4849|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4850|+
4851|+HTML_FORMULA_FORMAT = png
4852|+
4853|+ # Use this tag to change the font size of LaTeX formulas included as images in
4854|+ # the HTML documentation. When you change the font size after a successful
4855|+ # doxygen run you need to manually remove any form_*.png images from the HTML
4856|+ # output directory to force them to be regenerated.
4857|+ # Minimum value: 8, maximum value: 50, default value: 10.
4858|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4859|+
4860|+FORMULA_FONTSIZE = 10
4861|+
4862|+ # Use the FORMULA_TRANSPARENT tag to determine whether or not the images
4863|+ # generated for formulas are transparent PNGs. Transparent PNGs are not
4864|+ # supported properly for IE 6.0, but are supported on all modern browsers.
4865|+
4866|+ # Note that when changing this option you need to delete any form_*.png files in

```

```

4867|+ # the HTML output directory before the changes have effect.
4868|+ # The default value is: YES.
4869|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4870|+
4871|+FORMULA_TRANSPARENT      = YES
4872|+
4873|+ # The FORMULA_MACROFILE can contain LaTeX \newcommand and \renewcommand commands
4874|+ # to create new LaTeX commands to be used in formulas as building blocks. See
4875|+ # the section "Including formulas" for details.
4876|+
4877|+FORMULA_MACROFILE        =
4878|+
4879|+ # Enable the USE_MATHJAX option to render LaTeX formulas using MathJax (see
4880|+ # https://www.mathjax.org) which uses client side JavaScript for the rendering
4881|+ # instead of using pre-rendered bitmaps. Use this if you do not have LaTeX
4882|+ # installed or if you want to formulas look prettier in the HTML output. When
4883|+ # enabled you may also need to install MathJax separately and configure the path
4884|+ # to it using the MATHJAX_RELPATH option.
4885|+ # The default value is: NO.
4886|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4887|+
4888|+USE_MATHJAX              = NO
4889|+
4890|+ # With MATHJAX_VERSION it is possible to specify the MathJax version to be used.
4891|+ # Note that the different versions of MathJax have different requirements with
4892|+ # regards to the different settings, so it is possible that also other MathJax
4893|+ # settings have to be changed when switching between the different MathJax
4894|+ # versions.
4895|+ # Possible values are: MathJax_2 and MathJax_3.
4896|+ # The default value is: MathJax_2.
4897|+ # This tag requires that the tag USE_MATHJAX is set to YES.
4898|+
4899|+MATHJAX_VERSION          = MathJax_2
4900|+
4901|+ # When MathJax is enabled you need to specify the location relative to the HTML
4902|+ # the MathJax output. For more details about the output format see MathJax
4903|+ # version 2 (see:
4904|+ # http://docs.mathjax.org/en/v2.7-latest/output.html) and MathJax version 3
4905|+ # (see:
4906|+ # http://docs.mathjax.org/en/latest/web/components/output.html).
4907|+ # Possible values are: HTML-CSS (which is slower, but has the best
4908|+ # compatibility. This is the name for MathJax version 2, for MathJax version 3
4909|+ # this will be translated into chtml), NativeMML (i.e. MathML. Only supported
4910|+ # for MathJax 2. For MathJax version 3 chtml will be used instead.), chtml (This
4911|+ # is the name for MathJax version 3, for MathJax version 2 this will be
4912|+ # translated into HTML-CSS) and SVG.
4913|+ # The default value is: HTML-CSS.
4914|+ # This tag requires that the tag USE_MATHJAX is set to YES.
4915|+
4916|+MATHJAX_FORMAT           = HTML-CSS
4917|+
4918|+ # When MathJax is enabled you need to specify the location relative to the HTML
4919|+ # output directory using the MATHJAX_RELPATH option. The destination directory
4920|+ # should contain the MathJax.js script. For instance, if the mathjax directory
4921|+ # is located at the same level as the HTML output directory, then
4922|+ # MATHJAX_RELPATH should be ../mathjax. The default value points to the MathJax
4923|+ # Content Delivery Network so you can quickly see the result without installing
4924|+ # MathJax. However, it is strongly recommended to install a local copy of
4925|+ # MathJax from https://www.mathjax.org before deployment. The default value is:
4926|+ # - in case of MathJax version 2: https://cdn.jsdelivr.net/npm/mathjax@2
4927|+ # - in case of MathJax version 3: https://cdn.jsdelivr.net/npm/mathjax@3
4928|+ # This tag requires that the tag USE_MATHJAX is set to YES.
4929|+
4930|+MATHJAX_RELPATH          =
4931|+
4932|+ # The MATHJAX_EXTENSIONS tag can be used to specify one or more MathJax
4933|+ # extension names that should be enabled during MathJax rendering. For example
4934|+ # for MathJax version 2 (see
4935|+ # https://docs.mathjax.org/en/v2.7-latest/tex.html#tex-and-latex-extensions):
4936|+ # MATHJAX_EXTENSIONS = TeX/AMSTeX TeX/AMSsymbols
4937|+ # For example for MathJax version 3 (see
4938|+ # http://docs.mathjax.org/en/latest/input/tex/extensions/index.html):
4939|+ # MATHJAX_EXTENSIONS = ams
4940|+ # This tag requires that the tag USE_MATHJAX is set to YES.
4941|+
4942|+MATHJAX_EXTENSIONS       =
4943|+
4944|+ # The MATHJAX_CODEFILE tag can be used to specify a file with javascript pieces
4945|+ # of code that will be used on startup of the MathJax code. See the MathJax site
4946|+ # (see:
4947|+ # http://docs.mathjax.org/en/v2.7-latest/output.html) for more details. For an
4948|+ # example see the documentation.
4949|+ # This tag requires that the tag USE_MATHJAX is set to YES.
4950|+
4951|+MATHJAX_CODEFILE         =
4952|+
4953|+ # When the SEARCHENGINE tag is enabled doxygen will generate a search box for
4954|+ # the HTML output. The underlying search engine uses javascript and DHTML and
4955|+ # should work on any modern browser. Note that when using HTML help
4956|+ # (GENERATE_HTMLHELP), Qt help (GENERATE_QHP), or docsets (GENERATE_DOCSET)
4957|+ # there is already a search function so this one should typically be disabled.
4958|+ # For large projects the javascript based search engine can be slow, then
4959|+ # enabling SERVER_BASED_SEARCH may provide a better solution. It is possible to
4960|+ # search using the keyboard; to jump to the search box use <access key> + S
4961|+ # (what the <access key> is depends on the OS and browser, but it is typically
4962|+ # <CTRL>, <ALT> or <option>, or both). Inside the search box use the <cursor down
4963|+ # key> to jump into the search results window, the results can be navigated
4964|+ # using the <cursor keys>. Press <Enter> to select an item or <escape> to cancel
4965|+ # the search. The filter options can be selected when the cursor is inside the
4966|+ # search box by pressing <Shift>+<cursor down>. Also here use the <cursor keys>
4967|+ # to select a filter and <Enter> or <escape> to activate or cancel the filter
4968|+ # option.
4969|+ # The default value is: YES.
4970|+ # This tag requires that the tag GENERATE_HTML is set to YES.
4971|+
4972|+SEARCHENGINE             = YES
4973|+
4974|+ # When the SERVER_BASED_SEARCH tag is enabled the search engine will be
4975|+ # implemented using a web server instead of a web client using JavaScript. There
4976|+ # are two flavors of web server based searching depending on the EXTERNAL_SEARCH
4977|+ # setting. When disabled, doxygen will generate a PHP script for searching and
4978|+ # an index file used by the script. When EXTERNAL_SEARCH is enabled the indexing
4979|+ # and searching needs to be provided by external tools. See the section
4980|+ # "External Indexing and Searching" for details.
4981|+ # The default value is: NO.
4982|+ # This tag requires that the tag SEARCHENGINE is set to YES.

```

```

4983 | +
4984 | +SERVER_BASED_SEARCH      = NO
4985 | +
4986 | ## When EXTERNAL_SEARCH tag is enabled doxygen will no longer generate the PHP
4987 | ## script for searching. Instead the search results are written to an XML file
4988 | ## which needs to be processed by an external indexer. Doxygen will invoke an
4989 | ## external search engine pointed to by the SEARCHENGINE_URL option to obtain the
4990 | ## search results.
4991 | ##
4992 | ## Doxygen ships with an example indexer (doxyindexer) and search engine
4993 | ## (doxysearch.cgi) which are based on the open source search engine library
4994 | ## Xapian (see:
4995 | ## https://xapian.org/).
4996 | ##
4997 | ## See the section "External Indexing and Searching" for details.
4998 | ## The default value is: NO.
4999 | ## This tag requires that the tag SEARCHENGINE is set to YES.
5000 | +
5001 | +EXTERNAL_SEARCH          = NO
5002 | +
5003 | ## The SEARCHENGINE_URL should point to a search engine hosted by a web server
5004 | ## which will return the search results when EXTERNAL_SEARCH is enabled.
5005 | ##
5006 | ## Doxygen ships with an example indexer (doxyindexer) and search engine
5007 | ## (doxysearch.cgi) which are based on the open source search engine library
5008 | ## Xapian (see:
5009 | ## https://xapian.org/). See the section "External Indexing and Searching" for
5010 | ## details.
5011 | ## This tag requires that the tag SEARCHENGINE is set to YES.
5012 | +
5013 | +SEARCHENGINE_URL         =
5014 | +
5015 | ## When SERVER_BASED_SEARCH and EXTERNAL_SEARCH are both enabled the unindexed
5016 | ## search data is written to a file for indexing by an external tool. With the
5017 | ## SEARCHDATA_FILE tag the name of this file can be specified.
5018 | ## The default file is: searchdata.xml.
5019 | ## This tag requires that the tag SEARCHENGINE is set to YES.
5020 | +
5021 | +SEARCHDATA_FILE          = searchdata.xml
5022 | +
5023 | ## When SERVER_BASED_SEARCH and EXTERNAL_SEARCH are both enabled the
5024 | ## EXTERNAL_SEARCH_ID tag can be used as an identifier for the project. This is
5025 | ## useful in combination with EXTRA_SEARCH_MAPPINGS to search through multiple
5026 | ## projects and redirect the results back to the right project.
5027 | ## This tag requires that the tag SEARCHENGINE is set to YES.
5028 | +
5029 | +EXTERNAL_SEARCH_ID       =
5030 | +
5031 | ## The EXTRA_SEARCH_MAPPINGS tag can be used to enable searching through doxygen
5032 | ## projects other than the one defined by this configuration file, but that are
5033 | ## all added to the same external search index. Each project needs to have a
5034 | ## unique id set via EXTERNAL_SEARCH_ID. The search mapping then maps the id of
5035 | ## to a relative location where the documentation can be found. The format is:
5036 | ## EXTRA_SEARCH_MAPPINGS = tagname1=loc1 tagname2=loc2 ...
5037 | ## This tag requires that the tag SEARCHENGINE is set to YES.
5038 | +
5039 | +EXTRA_SEARCH_MAPPINGS    =
5040 | +
5041 | +-----+
5042 | ## Configuration options related to the LaTeX output
5043 | +-----+
5044 | +
5045 | ## If the GENERATE_LATEX tag is set to YES, doxygen will generate LaTeX output.
5046 | ## The default value is: YES.
5047 | +
5048 | +GENERATE_LATEX           = YES
5049 | +
5050 | ## The LATEX_OUTPUT tag is used to specify where the LaTeX docs will be put. If a
5051 | ## relative path is entered the value of OUTPUT_DIRECTORY will be put in front of
5052 | ## it.
5053 | ## The default directory is: latex.
5054 | ## This tag requires that the tag GENERATE_LATEX is set to YES.
5055 | +
5056 | +LATEX_OUTPUT             = latex
5057 | +
5058 | ## The LATEX_CMD_NAME tag can be used to specify the LaTeX command name to be
5059 | ## invoked.
5060 | ##
5061 | ## Note that when not enabling USE_PDFLATEX the default is latex when enabling
5062 | ## USE_PDFLATEX the default is pdflatex and when in the later case latex is
5063 | ## chosen this is overwritten by pdflatex. For specific output languages the
5064 | ## default can have been set differently, this depends on the implementation of
5065 | ## the output language.
5066 | ## This tag requires that the tag GENERATE_LATEX is set to YES.
5067 | +
5068 | +LATEX_CMD_NAME           =
5069 | +
5070 | ## The MAKEINDEX_CMD_NAME tag can be used to specify the command name to generate
5071 | ## index for LaTeX.
5072 | ## Note: This tag is used in the Makefile / make.bat.
5073 | ## See also: LATEX_MAKEINDEX_CMD for the part in the generated output file
5074 | ## (.tex).
5075 | ## The default file is: makeindex.
5076 | ## This tag requires that the tag GENERATE_LATEX is set to YES.
5077 | +
5078 | +MAKEINDEX_CMD_NAME       = makeindex
5079 | +
5080 | ## The LATEX_MAKEINDEX_CMD tag can be used to specify the command name to
5081 | ## generate index for LaTeX. In case there is no backslash (\) as first character
5082 | ## it will be automatically added in the LaTeX code.
5083 | ## Note: This tag is used in the generated output file (.tex).
5084 | ## See also: MAKEINDEX_CMD_NAME for the part in the Makefile / make.bat.
5085 | ## The default value is: makeindex.
5086 | ## This tag requires that the tag GENERATE_LATEX is set to YES.
5087 | +
5088 | +LATEX_MAKEINDEX_CMD      = makeindex
5089 | +
5090 | ## If the COMPACT_LATEX tag is set to YES, doxygen generates more compact LaTeX
5091 | ## documents. This may be useful for small projects and may help to save some
5092 | ## trees in general.
5093 | ## The default value is: NO.
5094 | ## This tag requires that the tag GENERATE_LATEX is set to YES.
5095 | +
5096 | +COMPACT_LATEX            = NO
5097 | +
5098 | ## The PAPER_TYPE tag can be used to set the paper type that is used by the

```



```

5099 ## printer.
5100 ## Possible values are: a4 (210 x 297 mm), letter (8.5 x 11 inches), legal (8.5 x
5101 ## 14 inches) and executive (7.25 x 10.5 inches).
5102 ## The default value is: a4.
5103 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5104 +
5105 +PAPER_TYPE = a4
5106 +
5107 ## The EXTRA_PACKAGES tag can be used to specify one or more LaTeX package names
5108 ## that should be included in the LaTeX output. The package can be specified just
5109 ## by its name or with the correct syntax as to be used with the LaTeX
5110 ## \usepackage command. To get the times font for instance you can specify :
5111 ## EXTRA_PACKAGES=times or EXTRA_PACKAGES={times}
5112 ## To use the option intlimits with the amsmath package you can specify:
5113 ## EXTRA_PACKAGES=[intlimits]{amsmath}
5114 ## If left blank no extra packages will be included.
5115 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5116 +
5117 +EXTRA_PACKAGES =
5118 +
5119 ## The LATEX_HEADER tag can be used to specify a user-defined LaTeX header for
5120 ## the generated LaTeX document. The header should contain everything until the
5121 ## first chapter. If it is left blank doxygen will generate a standard header. It
5122 ## is highly recommended to start with a default header using
5123 ## doxygen -w latex new_header.tex new_footer.tex new_stylesheet.sty
5124 ## and then modify the file new_header.tex. See also section "Doxygen usage" for
5125 ## information on how to generate the default header that doxygen normally uses.
5126 ##
5127 ## Note: Only use a user-defined header if you know what you are doing!
5128 ## Note: The header is subject to change so you typically have to regenerate the
5129 ## default header when upgrading to a newer version of doxygen. The following
5130 ## commands have a special meaning inside the header (and footer): For a
5131 ## description of the possible markers and block names see the documentation.
5132 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5133 +
5134 +LATEX_HEADER = header.tex
5135 +
5136 ## The LATEX_FOOTER tag can be used to specify a user-defined LaTeX footer for
5137 ## the generated LaTeX document. The footer should contain everything after the
5138 ## last chapter. If it is left blank doxygen will generate a standard footer. See
5139 ## LATEX_HEADER for more information on how to generate a default footer and what
5140 ## special commands can be used inside the footer. See also section "Doxygen
5141 ## usage" for information on how to generate the default footer that doxygen
5142 ## normally uses. Note: Only use a user-defined footer if you know what you are
5143 ## doing!
5144 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5145 +
5146 +LATEX_FOOTER =
5147 +
5148 ## The LATEX_EXTRA_STYLESHEET tag can be used to specify additional user-defined
5149 ## LaTeX style sheets that are included after the standard style sheets created
5150 ## by doxygen. Using this option one can overrule certain style aspects. Doxygen
5151 ## will copy the style sheet files to the output directory.
5152 ## Note: The order of the extra style sheet files is of importance (e.g. the last
5153 ## style sheet in the list overrules the setting of the previous ones in the
5154 ## list).
5155 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5156 +
5157 +LATEX_EXTRA_STYLESHEET =
5158 +
5159 ## The LATEX_EXTRA_FILES tag can be used to specify one or more extra images or
5160 ## other source files which should be copied to the LATEX_OUTPUT output
5161 ## directory. Note that the files will be copied as-is; there are no commands or
5162 ## markers available.
5163 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5164 +
5165 +LATEX_EXTRA_FILES =
5166 +
5167 ## If the PDF_HYPERLINKS tag is set to YES, the LaTeX that is generated is
5168 ## prepared for conversion to PDF (using ps2pdf or pdflatex). The PDF file will
5169 ## contain links (just like the HTML output) instead of page references. This
5170 ## makes the output suitable for online browsing using a PDF viewer.
5171 ## The default value is: YES.
5172 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5173 +
5174 +PDF_HYPERLINKS = YES
5175 +
5176 ## If the USE_PDFLATEX tag is set to YES, doxygen will use the engine as
5177 ## specified with LATEX_CMD_NAME to generate the PDF file directly from the LaTeX
5178 ## files. Set this option to YES, to get a higher quality PDF documentation.
5179 ##
5180 ## See also section LATEX_CMD_NAME for selecting the engine.
5181 ## The default value is: YES.
5182 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5183 +
5184 +USE_PDFLATEX = YES
5185 +
5186 ## If the LATEX_BATCHMODE tag is set to YES doxygen will add the \batchmode
5187 ## command to the generated LaTeX files. This will instruct LaTeX to keep running
5188 ## if errors occur instead of asking the user for help.
5189 ## The default value is: NO.
5190 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5191 +
5192 +LATEX_BATCHMODE = NO
5193 +
5194 ## If the LATEX_HIDE_INDICES tag is set to YES then doxygen will not include the
5195 ## index chapters (such as File Index, Compound Index, etc.) in the output.
5196 ## The default value is: NO.
5197 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5198 +
5199 +LATEX_HIDE_INDICES = NO
5200 +
5201 ## The LATEX_BIB_STYLE tag can be used to specify the style to use for the
5202 ## bibliography, e.g. plainnat, or leetr. See
5203 ## https://en.wikipedia.org/wiki/BibTeX and \cite for more info.
5204 ## The default value is: plain.
5205 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5206 +
5207 +LATEX_BIB_STYLE = plain
5208 +
5209 ## If the LATEX_TIMESTAMP tag is set to YES then the footer of each generated
5210 ## page will contain the date and time when the page was generated. Setting this
5211 ## to NO can help when comparing the output of multiple runs.
5212 ## The default value is: NO.
5213 ## This tag requires that the tag GENERATE_LATEX is set to YES.
5214 +

```

```

5215 +LATEX_TIMESTAMP           = NO
5216 +
5217 + # The LATEX_EMOJI_DIRECTORY tag is used to specify the (relative or absolute)
5218 + # path from which the emoji images will be read. If a relative path is entered,
5219 + # it will be relative to the LATEX_OUTPUT directory. If left blank the
5220 + # LATEX_OUTPUT directory will be used.
5221 + # This tag requires that the tag GENERATE_LATEX is set to YES.
5222 +
5223 +LATEX_EMOJI_DIRECTORY     =
5224 +
5225 +-----
5226 + # Configuration options related to the RTF output
5227 +-----
5228 +
5229 + # If the GENERATE_RTF tag is set to YES, doxygen will generate RTF output. The
5230 + # RTF output is optimized for Word 97 and may not look too pretty with other RTF
5231 + # readers/editors.
5232 + # The default value is: NO.
5233 +
5234 +GENERATE_RTF               = NO
5235 +
5236 + # The RTF_OUTPUT tag is used to specify where the RTF docs will be put. If a
5237 + # relative path is entered the value of OUTPUT_DIRECTORY will be put in front of
5238 + # it.
5239 + # The default directory is: rtf.
5240 + # This tag requires that the tag GENERATE_RTF is set to YES.
5241 +
5242 +RTF_OUTPUT                 = rtf
5243 +
5244 + # If the COMPACT_RTF tag is set to YES, doxygen generates more compact RTF
5245 + # documents. This may be useful for small projects and may help to save some
5246 + # trees in general.
5247 + # The default value is: NO.
5248 + # This tag requires that the tag GENERATE_RTF is set to YES.
5249 +
5250 +COMPACT_RTF                = NO
5251 +
5252 + # If the RTF_HYPERLINKS tag is set to YES, the RTF that is generated will
5253 + # contain hyperlink fields. The RTF file will contain links (just like the HTML
5254 + # output) instead of page references. This makes the output suitable for online
5255 + # browsing using Word or some other Word compatible readers that support those
5256 + # fields.
5257 +
5258 + # Note: WordPad (write) and others do not support links.
5259 + # The default value is: NO.
5260 + # This tag requires that the tag GENERATE_RTF is set to YES.
5261 +
5262 +RTF_HYPERLINKS             = NO
5263 +
5264 + # Load stylesheet definitions from file. Syntax is similar to doxygen's
5265 + # configuration file, i.e. a series of assignments. You only have to provide
5266 + # replacements, missing definitions are set to their default value.
5267 +
5268 + # See also section "Doxygen usage" for information on how to generate the
5269 + # default style sheet that doxygen normally uses.
5270 + # This tag requires that the tag GENERATE_RTF is set to YES.
5271 +
5272 +RTF_STYLESHEET_FILE        =
5273 +
5274 + # Set optional variables used in the generation of an RTF document. Syntax is
5275 + # similar to doxygen's configuration file. A template extensions file can be
5276 + # generated using doxygen -e rtf extensionFile.
5277 + # This tag requires that the tag GENERATE_RTF is set to YES.
5278 +
5279 +RTF_EXTENSIONS_FILE        =
5280 +
5281 +-----
5282 + # Configuration options related to the man page output
5283 +-----
5284 +
5285 + # If the GENERATE_MAN tag is set to YES, doxygen will generate man pages for
5286 + # classes and files.
5287 + # The default value is: NO.
5288 +
5289 +GENERATE_MAN                = NO
5290 +
5291 + # The MAN_OUTPUT tag is used to specify where the man pages will be put. If a
5292 + # relative path is entered the value of OUTPUT_DIRECTORY will be put in front of
5293 + # it. A directory man3 will be created inside the directory specified by
5294 + # MAN_OUTPUT.
5295 + # The default directory is: man.
5296 + # This tag requires that the tag GENERATE_MAN is set to YES.
5297 +
5298 +MAN_OUTPUT                  = man
5299 +
5300 + # The MAN_EXTENSION tag determines the extension that is added to the generated
5301 + # man pages. In case the manual section does not start with a number, the number
5302 + # 3 is prepended. The dot (.) at the beginning of the MAN_EXTENSION tag is
5303 + # optional.
5304 + # The default value is: .3.
5305 + # This tag requires that the tag GENERATE_MAN is set to YES.
5306 +
5307 +MAN_EXTENSION               = .3
5308 +
5309 + # The MAN_SUBDIR tag determines the name of the directory created within
5310 + # MAN_OUTPUT in which the man pages are placed. If defaults to man followed by
5311 + # MAN_EXTENSION with the initial . removed.
5312 + # This tag requires that the tag GENERATE_MAN is set to YES.
5313 +
5314 +MAN_SUBDIR                  =
5315 +
5316 + # If the MAN_LINKS tag is set to YES and doxygen generates man output, then it
5317 + # will generate one additional man file for each entity documented in the real
5318 + # man page(s). These additional files only source the real man page, but without
5319 + # them the man command would be unable to find the correct page.
5320 + # The default value is: NO.
5321 + # This tag requires that the tag GENERATE_MAN is set to YES.
5322 +
5323 +MAN_LINKS                   = NO
5324 +
5325 +-----
5326 + # Configuration options related to the XML output
5327 +-----
5328 +
5329 + # If the GENERATE_XML tag is set to YES, doxygen will generate an XML file that
5330 + # captures the structure of the code including all documentation.

```

```

5331|+# The default value is: NO.
5332|+
5333|+GENERATE_XML           = NO
5334|+
5335|+# The XML_OUTPUT tag is used to specify where the XML pages will be put. If a
5336|+# relative path is entered the value of OUTPUT_DIRECTORY will be put in front of
5337|+# it.
5338|+# The default directory is: xml.
5339|+# This tag requires that the tag GENERATE_XML is set to YES.
5340|+
5341|+XML_OUTPUT             = xml
5342|+
5343|+# If the XML_PROGRAMLISTING tag is set to YES, doxygen will dump the program
5344|+# listings (including syntax highlighting and cross-referencing information) to
5345|+# the XML output. Note that enabling this will significantly increase the size
5346|+# of the XML output.
5347|+# The default value is: YES.
5348|+# This tag requires that the tag GENERATE_XML is set to YES.
5349|+
5350|+XML_PROGRAMLISTING     = YES
5351|+
5352|+# If the XML_NS_MEMB_FILE_SCOPE tag is set to YES, doxygen will include
5353|+# namespace members in file scope as well, matching the HTML output.
5354|+# The default value is: NO.
5355|+# This tag requires that the tag GENERATE_XML is set to YES.
5356|+
5357|+XML_NS_MEMB_FILE_SCOPE = NO
5358|+
5359|+-----
5360|+# Configuration options related to the DOCBOOK output
5361|+-----
5362|+
5363|+# If the GENERATE_DOCBOOK tag is set to YES, doxygen will generate Docbook files
5364|+# that can be used to generate PDF.
5365|+# The default value is: NO.
5366|+
5367|+GENERATE_DOCBOOK       = NO
5368|+
5369|+# The DOCBOOK_OUTPUT tag is used to specify where the Docbook pages will be put.
5370|+# If a relative path is entered the value of OUTPUT_DIRECTORY will be put in
5371|+# front of it.
5372|+# The default directory is: docbook.
5373|+# This tag requires that the tag GENERATE_DOCBOOK is set to YES.
5374|+
5375|+DOCBOOK_OUTPUT         = docbook
5376|+
5377|+-----
5378|+# Configuration options for the AutoGen Definitions output
5379|+-----
5380|+
5381|+# If the GENERATE_AUTOGEN_DEF tag is set to YES, doxygen will generate an
5382|+# AutoGen Definitions (see http://autogen.sourceforge.net/) file that captures
5383|+# the structure of the code including all documentation. Note that this feature
5384|+# is still experimental and incomplete at the moment.
5385|+# The default value is: NO.
5386|+
5387|+GENERATE_AUTOGEN_DEF    = NO
5388|+
5389|+-----
5390|+# Configuration options related to Sqlite3 output
5391|+-----
5392|+
5393|+-----
5394|+# Configuration options related to the Perl module output
5395|+-----
5396|+
5397|+# If the GENERATE_PERLMOD tag is set to YES, doxygen will generate a Perl module
5398|+# file that captures the structure of the code including all documentation.
5399|+
5400|+# Note that this feature is still experimental and incomplete at the moment.
5401|+# The default value is: NO.
5402|+
5403|+GENERATE_PERLMOD        = NO
5404|+
5405|+# If the PERLMOD_LATEX tag is set to YES, doxygen will generate the necessary
5406|+# Makefile rules, Perl scripts and LaTeX code to be able to generate PDF and DVI
5407|+# output from the Perl module output.
5408|+# The default value is: NO.
5409|+# This tag requires that the tag GENERATE_PERLMOD is set to YES.
5410|+
5411|+PERLMOD_LATEX           = NO
5412|+
5413|+# If the PERLMOD_PRETTY tag is set to YES, the Perl module output will be nicely
5414|+# formatted so it can be parsed by a human reader. This is useful if you want to
5415|+# understand what is going on. On the other hand, if this tag is set to NO, the
5416|+# size of the Perl module output will be much smaller and Perl will parse it
5417|+# just the same.
5418|+# The default value is: YES.
5419|+# This tag requires that the tag GENERATE_PERLMOD is set to YES.
5420|+
5421|+PERLMOD_PRETTY          = YES
5422|+
5423|+# The names of the make variables in the generated doxyrules.make file are
5424|+# prefixed with the string contained in PERLMOD_MAKEVAR_PREFIX. This is useful
5425|+# so different doxyrules.make files included by the same Makefile don't
5426|+# overwrite each other's variables.
5427|+# This tag requires that the tag GENERATE_PERLMOD is set to YES.
5428|+
5429|+PERLMOD_MAKEVAR_PREFIX =
5430|+
5431|+-----
5432|+# Configuration options related to the preprocessor
5433|+-----
5434|+
5435|+# If the ENABLE_PREPROCESSING tag is set to YES, doxygen will evaluate all
5436|+# C-preprocessor directives found in the sources and include files.
5437|+# The default value is: YES.
5438|+
5439|+ENABLE_PREPROCESSING    = YES
5440|+
5441|+# If the MACRO_EXPANSION tag is set to YES, doxygen will expand all macro names
5442|+# in the source code. If set to NO, only conditional compilation will be
5443|+# performed. Macro expansion can be done in a controlled way by setting
5444|+# EXPAND_ONLY_PREDEF to YES.
5445|+# The default value is: NO.
5446|+# This tag requires that the tag ENABLE_PREPROCESSING is set to YES.

```



```

5447 | +
5448 | +MACRO_EXPANSION          = NO
5449 | +
5450 | ## If the EXPAND_ONLY_PREDEF and MACRO_EXPANSION tags are both set to YES then
5451 | ## the macro expansion is limited to the macros specified with the PREDEFINED and
5452 | ## EXPAND_AS_DEFINED tags.
5453 | ## The default value is: NO.
5454 | ## This tag requires that the tag ENABLE_PREPROCESSING is set to YES.
5455 | +
5456 | +EXPAND_ONLY_PREDEF      = NO
5457 | +
5458 | ## If the SEARCH_INCLUDES tag is set to YES, the include files in the
5459 | ## INCLUDE_PATH will be searched if a #include is found.
5460 | ## The default value is: YES.
5461 | ## This tag requires that the tag ENABLE_PREPROCESSING is set to YES.
5462 | +
5463 | +SEARCH_INCLUDES        = YES
5464 | +
5465 | ## The INCLUDE_PATH tag can be used to specify one or more directories that
5466 | ## contain include files that are not input files but should be processed by the
5467 | ## preprocessor.
5468 | ## This tag requires that the tag SEARCH_INCLUDES is set to YES.
5469 | +
5470 | +INCLUDE_PATH           =
5471 | +
5472 | ## You can use the INCLUDE_FILE_PATTERNS tag to specify one or more wildcard
5473 | ## patterns (like *.h and *.hpp) to filter out the header-files in the
5474 | ## directories. If left blank, the patterns specified with FILE_PATTERNS will be
5475 | ## used.
5476 | ## This tag requires that the tag ENABLE_PREPROCESSING is set to YES.
5477 | +
5478 | +INCLUDE_FILE_PATTERNS  =
5479 | +
5480 | ## The PREDEFINED tag can be used to specify one or more macro names that are
5481 | ## defined before the preprocessor is started (similar to the -D option of e.g.
5482 | ## gcc). The argument of the tag is a list of macros of the form: name or
5483 | ## name=definition (no spaces). If the definition and the "=" are omitted, "=1"
5484 | ## is assumed. To prevent a macro definition from being undefined via #undef or
5485 | ## recursively expanded use the := operator instead of the = operator.
5486 | ## This tag requires that the tag ENABLE_PREPROCESSING is set to YES.
5487 | +
5488 | +PREDEFINED             =
5489 | +
5490 | ## If the MACRO_EXPANSION and EXPAND_ONLY_PREDEF tags are set to YES then this
5491 | ## tag can be used to specify a list of macro names that should be expanded. The
5492 | ## macro definition that is found in the sources will be used. Use the PREDEFINED
5493 | ## tag if you want to use a different macro definition that overrules the
5494 | ## definition found in the source code.
5495 | ## This tag requires that the tag ENABLE_PREPROCESSING is set to YES.
5496 | +
5497 | +EXPAND_AS_DEFINED      =
5498 | +
5499 | ## If the SKIP_FUNCTION_MACROS tag is set to YES then doxygen's preprocessor will
5500 | ## remove all references to function-like macros that are alone on a line, have
5501 | ## an all uppercase name, and do not end with a semicolon. Such function macros
5502 | ## are typically used for boiler-plate code, and will confuse the parser if not
5503 | ## removed.
5504 | ## The default value is: YES.
5505 | ## This tag requires that the tag ENABLE_PREPROCESSING is set to YES.
5506 | +
5507 | +SKIP_FUNCTION_MACROS   = YES
5508 | +
5509 | +-----+
5510 | ## Configuration options related to external references
5511 | +-----+
5512 | +
5513 | ## The TAGFILES tag can be used to specify one or more tag files. For each tag
5514 | ## file the location of the external documentation should be added. The format of
5515 | ## a tag file without this location is as follows:
5516 | ## TAGFILES = file1 file2 ...
5517 | ## Adding location for the tag files is done as follows:
5518 | ## TAGFILES = file1=loc1 "file2 = loc2" ...
5519 | ## where loc1 and loc2 can be relative or absolute paths or URLs. See the
5520 | ## section "Linking to external documentation" for more information about the use
5521 | ## of tag files.
5522 | ## Note: Each tag file must have a unique name (where the name does NOT include
5523 | ## the path). If a tag file is not located in the directory in which doxygen is
5524 | ## run, you must also specify the path to the tagfile here.
5525 | +
5526 | +TAGFILES               =
5527 | +
5528 | ## When a file name is specified after GENERATE_TAGFILE, doxygen will create a
5529 | ## tag file that is based on the input files it reads. See section "Linking to
5530 | ## external documentation" for more information about the usage of tag files.
5531 | +
5532 | +GENERATE_TAGFILE       =
5533 | +
5534 | ## If the ALLEXTERNALS tag is set to YES, all external class will be listed in
5535 | ## the class index. If set to NO, only the inherited external classes will be
5536 | ## listed.
5537 | ## The default value is: NO.
5538 | +
5539 | +ALLEXTERNALS          = NO
5540 | +
5541 | ## If the EXTERNAL_GROUPS tag is set to YES, all external groups will be listed
5542 | ## in the modules index. If set to NO, only the current project's groups will be
5543 | ## listed.
5544 | ## The default value is: YES.
5545 | +
5546 | +EXTERNAL_GROUPS       = YES
5547 | +
5548 | ## If the EXTERNAL_PAGES tag is set to YES, all external pages will be listed in
5549 | ## the related pages index. If set to NO, only the current project's pages will
5550 | ## be listed.
5551 | ## The default value is: YES.
5552 | +
5553 | +EXTERNAL_PAGES        = YES
5554 | +
5555 | +-----+
5556 | ## Configuration options related to the dot tool
5557 | +-----+
5558 | +
5559 | ## If the CLASS_DIAGRAMS tag is set to YES, doxygen will generate a class diagram
5560 | ## (in HTML and LaTeX) for classes with base or super classes. Setting the tag to
5561 | ## NO turns the diagrams off. Note that this option also works with HAVE_DOT
5562 | ## disabled, but it is recommended to install and use dot, since it yields more

```

```

5563 | ## powerful graphs.
5564 | ## The default value is: YES.
5565 | +
5566 | +CLASS_DIAGRAMS      = YES
5567 | +
5568 | ## You can include diagrams made with dia in doxygen documentation. Doxygen will
5569 | ## then run dia to produce the diagram and insert it in the documentation. The
5570 | ## DIA_PATH tag allows you to specify the directory where the dia binary resides.
5571 | ## If left empty dia is assumed to be found in the default search path.
5572 | +
5573 | +DIA_PATH            =
5574 | +
5575 | ## If set to YES the inheritance and collaboration graphs will hide inheritance
5576 | ## and usage relations if the target is undocumented or is not a class.
5577 | ## The default value is: YES.
5578 | +
5579 | +HIDE_UNDOC_RELATIONS = YES
5580 | +
5581 | ## If you set the HAVE_DOT tag to YES then doxygen will assume the dot tool is
5582 | ## available from the path. This tool is part of Graphviz (see:
5583 | ## http://www.graphviz.org/), a graph visualization toolkit from AT&T and Lucent
5584 | ## Bell Labs. The other options in this section have no effect if this option is
5585 | ## set to NO
5586 | ## The default value is: NO.
5587 | +
5588 | +HAVE_DOT            = NO
5589 | +
5590 | ## The DOT_NUM_THREADS specifies the number of dot invocations doxygen is allowed
5591 | ## to run in parallel. When set to 0 doxygen will base this on the number of
5592 | ## processors available in the system. You can set it explicitly to a value
5593 | ## larger than 0 to get control over the balance between CPU load and processing
5594 | ## speed.
5595 | ## Minimum value: 0, maximum value: 32, default value: 0.
5596 | ## This tag requires that the tag HAVE_DOT is set to YES.
5597 | +
5598 | +DOT_NUM_THREADS     = 0
5599 | +
5600 | ## When you want a differently looking font in the dot files that doxygen
5601 | ## generates you can specify the font name using DOT_FONTNAME. You need to make
5602 | ## sure dot is able to find the font, which can be done by putting it in a
5603 | ## standard location or by setting the DOTFONTPATH environment variable or by
5604 | ## setting DOT_FONTPATH to the directory containing the font.
5605 | ## The default value is: Helvetica.
5606 | ## This tag requires that the tag HAVE_DOT is set to YES.
5607 | +
5608 | +DOT_FONTNAME        = Helvetica
5609 | +
5610 | ## The DOT_FONTSIZE tag can be used to set the size (in points) of the font of
5611 | ## dot graphs.
5612 | ## Minimum value: 4, maximum value: 24, default value: 10.
5613 | ## This tag requires that the tag HAVE_DOT is set to YES.
5614 | +
5615 | +DOT_FONTSIZE        = 10
5616 | +
5617 | ## By default doxygen will tell dot to use the default font as specified with
5618 | ## DOT_FONTNAME. If you specify a different font using DOT_FONTNAME you can set
5619 | ## the path where dot can find it using this tag.
5620 | ## This tag requires that the tag HAVE_DOT is set to YES.
5621 | +
5622 | +DOT_FONTPATH        =
5623 | +
5624 | ## If the CLASS_GRAPH tag is set to YES then doxygen will generate a graph for
5625 | ## each documented class showing the direct and indirect inheritance relations.
5626 | ## Setting this tag to YES will force the CLASS_DIAGRAMS tag to NO.
5627 | ## The default value is: YES.
5628 | ## This tag requires that the tag HAVE_DOT is set to YES.
5629 | +
5630 | +CLASS_GRAPH         = YES
5631 | +
5632 | ## If the COLLABORATION_GRAPH tag is set to YES then doxygen will generate a
5633 | ## graph for each documented class showing the direct and indirect implementation
5634 | ## dependencies (inheritance, containment, and class references variables) of the
5635 | ## class with other documented classes.
5636 | ## The default value is: YES.
5637 | ## This tag requires that the tag HAVE_DOT is set to YES.
5638 | +
5639 | +COLLABORATION_GRAPH = YES
5640 | +
5641 | ## If the GROUP_GRAPHS tag is set to YES then doxygen will generate a graph for
5642 | ## groups, showing the direct groups dependencies.
5643 | ## The default value is: YES.
5644 | ## This tag requires that the tag HAVE_DOT is set to YES.
5645 | +
5646 | +GROUP_GRAPHS        = YES
5647 | +
5648 | ## If the UML_LOOK tag is set to YES, doxygen will generate inheritance and
5649 | ## collaboration diagrams in a style similar to the OMG's Unified Modeling
5650 | ## Language.
5651 | ## The default value is: NO.
5652 | ## This tag requires that the tag HAVE_DOT is set to YES.
5653 | +
5654 | +UML_LOOK            = NO
5655 | +
5656 | ## If the UML_LOOK tag is enabled, the fields and methods are shown inside the
5657 | ## class node. If there are many fields or methods and many nodes the graph may
5658 | ## become too big to be useful. The UML_LIMIT_NUM_FIELDS threshold limits the
5659 | ## number of items for each type to make the size more manageable. Set this to 0
5660 | ## for no limit. Note that the threshold may be exceeded by 50% before the limit
5661 | ## is enforced. So when you set the threshold to 10, up to 15 fields may appear,
5662 | ## but if the number exceeds 15, the total amount of fields shown is limited to
5663 | ## 10.
5664 | ## Minimum value: 0, maximum value: 100, default value: 10.
5665 | ## This tag requires that the tag UML_LOOK is set to YES.
5666 | +
5667 | +UML_LIMIT_NUM_FIELDS = 10
5668 | +
5669 | ## If the DOT_UML_DETAILS tag is set to NO, doxygen will show attributes and
5670 | ## methods without types and arguments in the UML graphs. If the DOT_UML_DETAILS
5671 | ## tag is set to YES, doxygen will add type and arguments for attributes and
5672 | ## methods in the UML graphs. If the DOT_UML_DETAILS tag is set to NONE, doxygen
5673 | ## will not generate fields with class member information in the UML graphs. The
5674 | ## class diagrams will look similar to the default class diagrams but using UML
5675 | ## notation for the relationships.
5676 | ## Possible values are: NO, YES and NONE.
5677 | ## The default value is: NO.
5678 | ## This tag requires that the tag UML_LOOK is set to YES.

```

```

5679 +
5680 +DOT_UML_DETAILS          = NO
5681 +
5682 ## The DOT_WRAP_THRESHOLD tag can be used to set the maximum number of characters
5683 ## to display on a single line. If the actual line length exceeds this threshold
5684 ## significantly it will be wrapped across multiple lines. Some heuristics are apply
5685 ## to avoid ugly line breaks.
5686 ## Minimum value: 0, maximum value: 1000, default value: 17.
5687 ## This tag requires that the tag HAVE_DOT is set to YES.
5688 +
5689 +DOT_WRAP_THRESHOLD      = 17
5690 +
5691 ## If the TEMPLATE_RELATIONS tag is set to YES then the inheritance and
5692 ## collaboration graphs will show the relations between templates and their
5693 ## instances.
5694 ## The default value is: NO.
5695 ## This tag requires that the tag HAVE_DOT is set to YES.
5696 +
5697 +TEMPLATE_RELATIONS      = NO
5698 +
5699 ## If the INCLUDE_GRAPH, ENABLE_PREPROCESSING and SEARCH_INCLUDES tags are set to
5700 ## YES then doxygen will generate a graph for each documented file showing the
5701 ## direct and indirect include dependencies of the file with other documented
5702 ## files.
5703 ## The default value is: YES.
5704 ## This tag requires that the tag HAVE_DOT is set to YES.
5705 +
5706 +INCLUDE_GRAPH           = YES
5707 +
5708 ## If the INCLUDED_BY_GRAPH, ENABLE_PREPROCESSING and SEARCH_INCLUDES tags are
5709 ## set to YES then doxygen will generate a graph for each documented file showing
5710 ## the direct and indirect include dependencies of the file with other documented
5711 ## files.
5712 ## The default value is: YES.
5713 ## This tag requires that the tag HAVE_DOT is set to YES.
5714 +
5715 +INCLUDED_BY_GRAPH       = YES
5716 +
5717 ## If the CALL_GRAPH tag is set to YES then doxygen will generate a call
5718 ## dependency graph for every global function or class method.
5719 +
5720 ## Note that enabling this option will significantly increase the time of a run.
5721 ## So in most cases it will be better to enable call graphs for selected
5722 ## functions only using the \callgraph command. Disabling a call graph can be
5723 ## accomplished by means of the command \hidecallgraph.
5724 ## The default value is: NO.
5725 ## This tag requires that the tag HAVE_DOT is set to YES.
5726 +
5727 +CALL_GRAPH              = NO
5728 +
5729 ## If the CALLER_GRAPH tag is set to YES then doxygen will generate a caller
5730 ## dependency graph for every global function or class method.
5731 +
5732 ## Note that enabling this option will significantly increase the time of a run.
5733 ## So in most cases it will be better to enable caller graphs for selected
5734 ## functions only using the \callergraph command. Disabling a caller graph can be
5735 ## accomplished by means of the command \hidecallergraph.
5736 ## The default value is: NO.
5737 ## This tag requires that the tag HAVE_DOT is set to YES.
5738 +
5739 +CALLER_GRAPH            = NO
5740 +
5741 ## If the GRAPHICAL_HIERARCHY tag is set to YES then doxygen will graphical
5742 ## hierarchy of all classes instead of a textual one.
5743 ## The default value is: YES.
5744 ## This tag requires that the tag HAVE_DOT is set to YES.
5745 +
5746 +GRAPHICAL_HIERARCHY     = YES
5747 +
5748 ## If the DIRECTORY_GRAPH tag is set to YES then doxygen will show the
5749 ## dependencies a directory has on other directories in a graphical way. The
5750 ## dependency relations are determined by the #include relations between the
5751 ## files in the directories.
5752 ## The default value is: YES.
5753 ## This tag requires that the tag HAVE_DOT is set to YES.
5754 +
5755 +DIRECTORY_GRAPH         = YES
5756 +
5757 ## The DOT_IMAGE_FORMAT tag can be used to set the image format of the images
5758 ## generated by dot. For an explanation of the image formats see the section
5759 ## output formats in the documentation of the dot tool (Graphviz (see:
5760 ## http://www.graphviz.org/)).
5761 ## Note: If you choose svg you need to set HTML_FILE_EXTENSION to xhtml in order
5762 ## to make the SVG files visible in IE 9+ (other browsers do not have this
5763 ## requirement).
5764 ## Possible values are: png, jpg, gif, svg, png:gd, png:gd:gd, png:cairo,
5765 ## png:cairo:gd, png:cairo:cairo, png:cairo:gdiplus, png:gdiplus and
5766 ## png:gdiplus:gdiplus.
5767 ## The default value is: png.
5768 ## This tag requires that the tag HAVE_DOT is set to YES.
5769 +
5770 +DOT_IMAGE_FORMAT        = png
5771 +
5772 ## If DOT_IMAGE_FORMAT is set to svg, then this option can be set to YES to
5773 ## enable generation of interactive SVG images that allow zooming and panning.
5774 +
5775 ## Note that this requires a modern browser other than Internet Explorer. Tested
5776 ## and working are Firefox, Chrome, Safari, and Opera.
5777 ## Note: For IE 9+ you need to set HTML_FILE_EXTENSION to xhtml in order to make
5778 ## the SVG files visible. Older versions of IE do not have SVG support.
5779 ## The default value is: NO.
5780 ## This tag requires that the tag HAVE_DOT is set to YES.
5781 +
5782 +INTERACTIVE_SVG         = NO
5783 +
5784 ## The DOT_PATH tag can be used to specify the path where the dot tool can be
5785 ## found. If left blank, it is assumed the dot tool can be found in the path.
5786 ## This tag requires that the tag HAVE_DOT is set to YES.
5787 +
5788 +DOT_PATH                =
5789 +
5790 ## The DOTFILE_DIRS tag can be used to specify one or more directories that
5791 ## contain dot files that are included in the documentation (see the \dotfile
5792 ## command).
5793 ## This tag requires that the tag HAVE_DOT is set to YES.
5794 +

```

```

5795 +DOTFILE_DIRS           =
5796 +
5797 + # The MSCFILE_DIRS tag can be used to specify one or more directories that
5798 + # contain msc files that are included in the documentation (see the \mscfile
5799 + # command).
5800 +
5801 +MSCFILE_DIRS           =
5802 +
5803 + # The DIAFILE_DIRS tag can be used to specify one or more directories that
5804 + # contain dia files that are included in the documentation (see the \diafile
5805 + # command).
5806 +
5807 +DIAFILE_DIRS           =
5808 +
5809 + # When using plantuml, the PLANTUML_JAR_PATH tag should be used to specify the
5810 + # path where java can find the plantuml.jar file. If left blank, it is assumed
5811 + # PlantUML is not used or called during a preprocessing step. Doxygen will
5812 + # generate a warning when it encounters a \startuml command in this case and
5813 + # will not generate output for the diagram.
5814 +
5815 +PLANTUML_JAR_PATH      =
5816 +
5817 + # When using plantuml, the PLANTUML_CFG_FILE tag can be used to specify a
5818 + # configuration file for plantuml.
5819 +
5820 +PLANTUML_CFG_FILE      =
5821 +
5822 + # When using plantuml, the specified paths are searched for files specified by
5823 + # the !include statement in a plantuml block.
5824 +
5825 +PLANTUML_INCLUDE_PATH  =
5826 +
5827 + # The DOT_GRAPH_MAX_NODES tag can be used to set the maximum number of nodes
5828 + # that will be shown in the graph. If the number of nodes in a graph becomes
5829 + # larger than this value, doxygen will truncate the graph, which is visualized
5830 + # by representing a node as a red box. Note that doxygen if the number of direct
5831 + # children of the root node in a graph is already larger than
5832 + # DOT_GRAPH_MAX_NODES then the graph will not be shown at all. Also note that
5833 + # the size of a graph can be further restricted by MAX_DOT_GRAPH_DEPTH.
5834 + # Minimum value: 0, maximum value: 10000, default value: 50.
5835 + # This tag requires that the tag HAVE_DOT is set to YES.
5836 +
5837 +DOT_GRAPH_MAX_NODES    = 50
5838 +
5839 + # The MAX_DOT_GRAPH_DEPTH tag can be used to set the maximum depth of the graphs
5840 + # generated by dot. A depth value of 3 means that only nodes reachable from the
5841 + # root by following a path via at most 3 edges will be shown. Nodes that lay
5842 + # further from the root node will be omitted. Note that setting this option to 1
5843 + # or 2 may greatly reduce the computation time needed for large code bases. Also
5844 + # note that the size of a graph can be further restricted by
5845 + # DOT_GRAPH_MAX_NODES. Using a depth of 0 means no depth restriction.
5846 + # Minimum value: 0, maximum value: 1000, default value: 0.
5847 + # This tag requires that the tag HAVE_DOT is set to YES.
5848 +
5849 +MAX_DOT_GRAPH_DEPTH    = 0
5850 +
5851 + # Set the DOT_TRANSPARENT tag to YES to generate images with a transparent
5852 + # background. This is disabled by default, because dot on Windows does not seem
5853 + # to support this out of the box.
5854 +
5855 + # Warning: Depending on the platform used, enabling this option may lead to
5856 + # badly anti-aliased labels on the edges of a graph (i.e. they become hard to
5857 + # read).
5858 + # The default value is: NO.
5859 + # This tag requires that the tag HAVE_DOT is set to YES.
5860 +
5861 +DOT_TRANSPARENT        = NO
5862 +
5863 + # Set the DOT_MULTI_TARGETS tag to YES to allow dot to generate multiple output
5864 + # files in one run (i.e. multiple -o and -T options on the command line). This
5865 + # makes dot run faster, but since only newer versions of dot (>1.8.10) support
5866 + # this, this feature is disabled by default.
5867 + # The default value is: NO.
5868 + # This tag requires that the tag HAVE_DOT is set to YES.
5869 +
5870 +DOT_MULTI_TARGETS      = NO
5871 +
5872 + # If the GENERATE_LEGEND tag is set to YES doxygen will generate a legend page
5873 + # explaining the meaning of the various boxes and arrows in the dot generated
5874 + # graphs.
5875 + # The default value is: YES.
5876 + # This tag requires that the tag HAVE_DOT is set to YES.
5877 +
5878 +GENERATE_LEGEND        = YES
5879 +
5880 + # If the DOT_CLEANUP tag is set to YES, doxygen will remove the intermediate
5881 + # files that are used to generate the various graphs.
5882 +
5883 + # Note: This setting is not only used for dot files but also for msc temporary
5884 + # files.
5885 + # The default value is: YES.
5886 +
5887 +DOT_CLEANUP            = YES
5888 diff --git a/doxygen.sty b/doxygen.sty
5889 new file mode 100644
5890 index 0000000..326f830
5891 --- /dev/null
5892 +++ b/doxygen.sty
5893 @@ -0,0 +1,589 @@
5894 +% stylesheet for doxygen 1.9.2
5895 +\NeedsTeXFormat{LaTeX2e}
5896 +\ProvidesPackage{doxygen}
5897 +
5898 +% Packages used by this style file
5899 +\RequirePackage{alltt}
5900 +% \RequirePackage{array} %% moved to refman.tex due to workaround for LaTeX 2019 version and unmaintained tabu
5901 +\RequirePackage{calc}
5902 +\RequirePackage{float}
5903 +% \RequirePackage{ifthen} %% moved to refman.tex due to workaround for LaTeX 2019 version and unmaintained tabu
5904 +\RequirePackage{verbatim}
5905 +\RequirePackage[table]{xcolor}
5906 +\RequirePackage{longtable}
5907 +\RequirePackage{tabu}
5908 +\RequirePackage{fancyvrb}

```

```

5909 +\RequirePackage{tabularx}
5910 +\RequirePackage{multicol}
5911 +\RequirePackage{multirow}
5912 +\RequirePackage{hanging}
5913 +\RequirePackage{ifpdf}
5914 +\RequirePackage{adjustbox}
5915 +\RequirePackage{amssymb}
5916 +\RequirePackage{stackengine}
5917 +\RequirePackage{enumitem}
5918 +\RequirePackage{alphalph}
5919 +\RequirePackage[normalem]{ulem} % for strikeouts, but don't modify emphasis
5920 +
5921 +%----- Internal commands used in this style file -----
5922 +
5923 +\newcommand{\ensurespace}[1]{%
5924 +  \begingroup%
5925 +    \setlength{\dimen0}{#1}%
5926 +    \vskip\z@\@plus\dimen0%
5927 +    \penalty -100\vskip\z@\@plus -\dimen0%
5928 +    \vskip\dimen0%
5929 +    \penalty 9999%
5930 +    \vskip -\dimen0%
5931 +    \vskip\z@skip% hide the previous |\vskip| from |\addvspace|
5932 +  \endgroup%
5933 +}
5934 +
5935 +\newcommand{\DoxyHorRuler}[1]{%
5936 +  \setlength{\parskip}{0ex plus 0ex minus 0ex}%
5937 +  \ifthenelse{#1=0}%
5938 +  {%
5939 +    \hrule%
5940 +  }%
5941 +  {%
5942 +    \hrulefill%
5943 +  }%
5944 +}
5945 +\newcommand{\DoxyLabelFont}{}
5946 +\newcommand{\entrylabel}[1]{%
5947 +  {%
5948 +    \parbox[b]{\labelwidth-4pt}{%
5949 +      \makebox[0pt][l]{\DoxyLabelFont#1}%
5950 +      \vspace{1.5\baselineskip}%
5951 +    }%
5952 +  }%
5953 +}
5954 +
5955 +\newenvironment{DoxyDesc}[1]{%
5956 +  \ensurespace{4\baselineskip}%
5957 +  \begin{list}{}{%
5958 +    \settoheight{\labelwidth}{20pt}%
5959 +    \setlength{\parsep}{0pt}%
5960 +    \setlength{\itemsep}{0pt}%
5961 +    \setlength{\leftmargin}{\labelwidth+\labelsep}%
5962 +  }%
5963 +  \renewcommand{\makelabel}{\entrylabel}%
5964 +  \item[1]%
5965 +}{%
5966 +  \end{list}%
5967 +}
5968 +
5969 +\newsavebox{\xrefbox}
5970 +\newlength{\xreflength}
5971 +\newcommand{\xreflabel}[1]{%
5972 +  \sbox{\xrefbox}{#1}%
5973 +  \setlength{\xreflength}{\wd\xrefbox}%
5974 +  \ifthenelse{\xreflength>\labelwidth}{%
5975 +    \begin{minipage}{\textwidth}%
5976 +      \setlength{\parindent}{0pt}%
5977 +      \hangindent=15pt\bfseries #1\vspace{1.2\itemsep}%
5978 +    \end{minipage}%
5979 +  }{%
5980 +    \parbox[b]{\labelwidth}{\makebox[0pt][l]{\textbf{#1}}}%
5981 +  }%
5982 +}
5983 +
5984 +%----- Commands used by doxygen LaTeX output generator -----
5985 +
5986 +% Used by <pre> ... </pre>
5987 +\newenvironment{DoxyPre}{%
5988 +  \small%
5989 +  \begin{alltt}%
5990 +}{%
5991 +  \end{alltt}%
5992 +  \normalsize%
5993 +}
5994 +
5995 +% Necessary for redefining not defined characters, i.e. "Replacement Character" in tex output.
5996 +\newlength{\CodeWidthChar}
5997 +\newlength{\CodeHeightChar}
5998 +\settoheight{\CodeWidthChar}{?}
5999 +\settoheight{\CodeHeightChar}{?}
6000 +% Necessary for hanging indent
6001 +\newlength{\DoxyCodeWidth}
6002 +
6003 +\newcommand\DoxyCodeLine[1]{\hangpara{\DoxyCodeWidth}{1}{#1}\par}
6004 +
6005 +\newcommand\NiceSpace{%
6006 +  \discretionary{}{\kern\fontdimen2\font}{\kern\fontdimen2\font}%
6007 +}
6008 +
6009 +% Used by @code ... @endcode
6010 +\newenvironment{DoxyCode}[1]{%
6011 +  \par%
6012 +  \scriptsize%
6013 +  \normalfont\ttfamily%
6014 +  \rightskip0pt plus 1fil%
6015 +  \settoheight{\DoxyCodeWidth}{000000}%
6016 +  \settoheight{\CodeWidthChar}{?}%
6017 +  \settoheight{\CodeHeightChar}{?}%
6018 +  \setlength{\parskip}{0ex plus 0ex minus 0ex}%
6019 +  \ifthenelse{\equal{#1}{0}}{
6020 +    {\lccode'~32 \lowercase{\global\let~}\NiceSpace}\obeyspaces%
6021 +  }{
6022 +    {\lccode'~32 \lowercase{\global\let~}}\obeyspaces%
6023 +  }
6024 +}

```

```

6025| +
6026| +}{%
6027| + \normalfont%
6028| + \normalsize%
6029| + \settowidth{\CodeWidthChar}{?}%
6030| + \settoheight{\CodeHeightChar}{?}%
6031| +}
6032| +
6033| +% Redefining not defined characters, i.e. "Replacement Character" in tex output.
6034| +\def\ucr{\adjustbox{width=\CodeWidthChar,height=\CodeHeightChar}{\stackinset{c}{c}{-.2pt}{c}{%
6035| + \textcolor{white}{\sfamily\bfseries\small ?}}}%
6036| + \rotatebox{45}{\blacksquare$}}}%
6037| +
6038| +% Used by @example, @include, @includelineno and @dontinclude
6039| +\newenvironment{DoxyCodeInclude}[1]{%
6040| + \DoxyCode{#1}%
6041| +}{%
6042| + \endDoxyCode%
6043| +}
6044| +
6045| +% Used by @verbatim ... @endverbatim
6046| +\newenvironment{DoxyVerb}{%
6047| + \footnotesize%
6048| + \verbatim%
6049| +}{%
6050| + \endverbatim%
6051| + \normalsize%
6052| +}
6053| +
6054| +% Used by @verbinclude
6055| +\newenvironment{DoxyVerbInclude}{%
6056| + \DoxyVerb%
6057| +}{%
6058| + \endDoxyVerb%
6059| +}
6060| +
6061| +% Used by numbered lists (using '-#' or <ol> ... </ol>)
6062| +\newenvironment{DoxyEnumerate}{%
6063| + \enumerate%
6064| +}{%
6065| + \endenumerate%
6066| +}
6067| +
6068| +% Used by bullet lists (using '-', @li, @arg, or <ul> ... </ul>)
6069| +\newenvironment{DoxyItemize}{%
6070| + \itemize%
6071| +}{%
6072| + \enditemize%
6073| +}
6074| +
6075| +% Used by description lists (using <dl> ... </dl>)
6076| +\newenvironment{DoxyDescription}{%
6077| + \description%
6078| +}{%
6079| + \enddescription%
6080| +}
6081| +
6082| +% Used by @image, @dotfile, @dot ... @enddot, and @msc ... @endmsc
6083| +% (only if caption is specified)
6084| +\newenvironment{DoxyImage}{%
6085| + \begin{figure}[H]%
6086| + \centering%
6087| +}{%
6088| + \end{figure}%
6089| +}
6090| +
6091| +% Used by @image, @dotfile, @dot ... @enddot, and @msc ... @endmsc
6092| +% (only if no caption is specified)
6093| +\newenvironment{DoxyImageNoCaption}{%
6094| + \begin{center}%
6095| +}{%
6096| + \end{center}%
6097| +}
6098| +
6099| +% Used by @image
6100| +% (only if inline is specified)
6101| +\newenvironment{DoxyInlineImage}{%
6102| +}{%
6103| +}
6104| +
6105| +% Used by @attention
6106| +\newenvironment{DoxyAttention}[1]{%
6107| + \begin{DoxyDesc}{#1}%
6108| +}{%
6109| + \end{DoxyDesc}%
6110| +}
6111| +
6112| +% Used by @author and @authors
6113| +\newenvironment{DoxyAuthor}[1]{%
6114| + \begin{DoxyDesc}{#1}%
6115| +}{%
6116| + \end{DoxyDesc}%
6117| +}
6118| +
6119| +% Used by @date
6120| +\newenvironment{DoxyDate}[1]{%
6121| + \begin{DoxyDesc}{#1}%
6122| +}{%
6123| + \end{DoxyDesc}%
6124| +}
6125| +
6126| +% Used by @invariant
6127| +\newenvironment{DoxyInvariant}[1]{%
6128| + \begin{DoxyDesc}{#1}%
6129| +}{%
6130| + \end{DoxyDesc}%
6131| +}
6132| +
6133| +% Used by @note
6134| +\newenvironment{DoxyNote}[1]{%
6135| + \begin{DoxyDesc}{#1}%
6136| +}{%
6137| + \end{DoxyDesc}%
6138| +}
6139| +
6140| +% Used by @post

```

```

6141 +\newenvironment{DoxyPostcond}[1]{%
6142 + \begin{DoxyDesc}{#1}%
6143 +}%
6144 + \end{DoxyDesc}%
6145 +}
6146 +
6147 +% Used by @pre
6148 +\newenvironment{DoxyPrecond}[1]{%
6149 + \begin{DoxyDesc}{#1}%
6150 +}%
6151 + \end{DoxyDesc}%
6152 +}
6153 +
6154 +% Used by @copyright
6155 +\newenvironment{DoxyCopyright}[1]{%
6156 + \begin{DoxyDesc}{#1}%
6157 +}%
6158 + \end{DoxyDesc}%
6159 +}
6160 +
6161 +% Used by @remark
6162 +\newenvironment{DoxyRemark}[1]{%
6163 + \begin{DoxyDesc}{#1}%
6164 +}%
6165 + \end{DoxyDesc}%
6166 +}
6167 +
6168 +% Used by @return and @returns
6169 +\newenvironment{DoxyReturn}[1]{%
6170 + \begin{DoxyDesc}{#1}%
6171 +}%
6172 + \end{DoxyDesc}%
6173 +}
6174 +
6175 +% Used by @since
6176 +\newenvironment{DoxySince}[1]{%
6177 + \begin{DoxyDesc}{#1}%
6178 +}%
6179 + \end{DoxyDesc}%
6180 +}
6181 +
6182 +% Used by @see
6183 +\newenvironment{DoxySeeAlso}[1]{%
6184 + \begin{DoxyDesc}{#1}%
6185 +}%
6186 + \end{DoxyDesc}%
6187 +}
6188 +
6189 +% Used by @version
6190 +\newenvironment{DoxyVersion}[1]{%
6191 + \begin{DoxyDesc}{#1}%
6192 +}%
6193 + \end{DoxyDesc}%
6194 +}
6195 +
6196 +% Used by @warning
6197 +\newenvironment{DoxyWarning}[1]{%
6198 + \begin{DoxyDesc}{#1}%
6199 +}%
6200 + \end{DoxyDesc}%
6201 +}
6202 +
6203 +% Used by @internal
6204 +\newenvironment{DoxyInternal}[1]{%
6205 + \paragraph*{#1}%
6206 +}%
6207 +}
6208 +
6209 +% Used by @par and @paragraph
6210 +\newenvironment{DoxyParagraph}[1]{%
6211 + \begin{DoxyDesc}{#1}%
6212 +}%
6213 + \end{DoxyDesc}%
6214 +}
6215 +
6216 +% Used by parameter lists
6217 +\newenvironment{DoxyParams}[2][1]{%
6218 + \tabulinesep=1mm%
6219 + \par%
6220 + \ifthenelse{\equal{#1}{}}%
6221 + {\begin{longtabu*}spread Opt [1]{|X[-1,1]|X[-1,1]|}% name + description
6222 + {\ifthenelse{\equal{#1}{1}}%
6223 + {\begin{longtabu*}spread Opt [1]{|X[-1,1]|X[-1,1]|X[-1,1]|}% in/out + name + desc
6224 + {\begin{longtabu*}spread Opt [1]{|X[-1,1]|X[-1,1]|X[-1,1]|X[-1,1]|}% in/out + type + name + desc
6225 + }
6226 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #2}\[1ex]%
6227 + \hline%
6228 + \endfirsthead%
6229 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #2}\[1ex]%
6230 + \hline%
6231 + \endhead%
6232 +}%
6233 + \end{longtabu*}%
6234 + \vspace{6pt}%
6235 +}
6236 +
6237 +% Used for fields of simple structs
6238 +\newenvironment{DoxyFields}[1]{%
6239 + \tabulinesep=1mm%
6240 + \par%
6241 + \begin{longtabu*}spread Opt [1]{|X[-1,r]|X[-1,1]|X[-1,1]|}%
6242 + \multicolumn{3}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6243 + \hline%
6244 + \endfirsthead%
6245 + \multicolumn{3}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6246 + \hline%
6247 + \endhead%
6248 +}%
6249 + \end{longtabu*}%
6250 + \vspace{6pt}%
6251 +}
6252 +
6253 +% Used for fields simple class style enums
6254 +\newenvironment{DoxyEnumFields}[1]{%
6255 + \tabulinesep=1mm%
6256 + \par%

```



```

6257 + \begin{longtabu*}spread Opt [1]{|X[-1,r]|X[-1,1]|}%
6258 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6259 + \hline%
6260 + \endfirsthead%
6261 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6262 + \hline%
6263 + \endhead%
6264 +}%
6265 + \end{longtabu*}%
6266 + \vspace{6pt}%
6267 +
6268 +
6269 +% Used for parameters within a detailed function description
6270 +\newenvironment{DoxyParamCaption}{%
6271 + \renewcommand{\item}[2][{\hspace*{2.0cm} ##1 {\em ##2}}}%
6272 +}%
6273 +
6274 +
6275 +% Used by return value lists
6276 +\newenvironment{DoxyRetVal}[1]{%
6277 + \tabulinesep=1mm%
6278 + \par%
6279 + \begin{longtabu*}spread Opt [1]{|X[-1,r]|X[-1,1]|}%
6280 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6281 + \hline%
6282 + \endfirsthead%
6283 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6284 + \hline%
6285 + \endhead%
6286 +}%
6287 + \end{longtabu*}%
6288 + \vspace{6pt}%
6289 +
6290 +
6291 +% Used by exception lists
6292 +\newenvironment{DoxyExceptions}[1]{%
6293 + \tabulinesep=1mm%
6294 + \par%
6295 + \begin{longtabu*}spread Opt [1]{|X[-1,r]|X[-1,1]|}%
6296 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6297 + \hline%
6298 + \endfirsthead%
6299 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6300 + \hline%
6301 + \endhead%
6302 +}%
6303 + \end{longtabu*}%
6304 + \vspace{6pt}%
6305 +
6306 +
6307 +% Used by template parameter lists
6308 +\newenvironment{DoxyTemplParams}[1]{%
6309 + \tabulinesep=1mm%
6310 + \par%
6311 + \begin{longtabu*}spread Opt [1]{|X[-1,r]|X[-1,1]|}%
6312 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6313 + \hline%
6314 + \endfirsthead%
6315 + \multicolumn{2}{1}{\hspace{-6pt}\bfseries\fontseries{bc}\selectfont\color{darkgray} #1}\[1ex]%
6316 + \hline%
6317 + \endhead%
6318 +}%
6319 + \end{longtabu*}%
6320 + \vspace{6pt}%
6321 +
6322 +
6323 +% Used for member lists
6324 +\newenvironment{DoxyCompactItemize}{%
6325 + \begin{itemize}%
6326 + \setlength{\itemsep}{-3pt}%
6327 + \setlength{\parsep}{0pt}%
6328 + \setlength{\topsep}{0pt}%
6329 + \setlength{\partopsep}{0pt}%
6330 +}%
6331 + \end{itemize}%
6332 +
6333 +
6334 +% Used for member descriptions
6335 +\newenvironment{DoxyCompactList}{%
6336 + \begin{list}{}{%
6337 + \setlength{\leftmargin}{0.5cm}%
6338 + \setlength{\itemsep}{0pt}%
6339 + \setlength{\parsep}{0pt}%
6340 + \setlength{\topsep}{0pt}%
6341 + \renewcommand{\makelabel}{\hfill}%
6342 + }%
6343 +}%
6344 + \end{list}%
6345 +
6346 +
6347 +% Used for reference lists (@bug, @deprecated, @todo, etc.)
6348 +\newenvironment{DoxyRefList}{%
6349 + \begin{list}{}{%
6350 + \setlength{\labelwidth}{10pt}%
6351 + \setlength{\leftmargin}{\labelwidth}%
6352 + \addtolength{\leftmargin}{\labelsep}%
6353 + \renewcommand{\makelabel}{\xreflabel}%
6354 + }%
6355 +}%
6356 + \end{list}%
6357 +
6358 +
6359 +% Used by @bug, @deprecated, @todo, etc.
6360 +\newenvironment{DoxyRefDesc}[1]{%
6361 + \begin{list}{}{%
6362 + \renewcommand{\makelabel}[1]{\textbf{##1}}%
6363 + \settowidth{\labelwidth}{\makelabel{##1}}%
6364 + \setlength{\leftmargin}{\labelwidth+\labelsep}%
6365 + }%
6366 +}%
6367 + \end{list}%
6368 +
6369 +
6370 +% Used by parameter lists and simple sections
6371 +\newenvironment{Desc}{%
6372 + \begin{list}{}{%

```



```

6373 + \settowidth{\labelwidth}{20pt}%
6374 + \setlength{\parsep}{0pt}%
6375 + \setlength{\itemsep}{0pt}%
6376 + \setlength{\leftmargin}{\labelwidth+\labelsep}%
6377 + \renewcommand{\makelabel}{\entrylabel}%
6378 + }
6379 +}%
6380 + \end{list}%
6381 +}
6382 +
6383 +% Used by tables
6384 +\newcommand{\PBS}[1]{\let\temp=\#1\let\=\temp}%
6385 +\newenvironment{TabularC}[1]%
6386 +{\tabulinesep=1mm
6387 +\begin{longtabu*}spread Opt [c]{*#1{X[-1]}|}}%
6388 +{\end{longtabu*}\par}%
6389 +
6390 +\newenvironment{TabularNC}[1]%
6391 +{\begin{tabu}spread Opt [l]{*#1{X[-1]}|}}%
6392 +{\end{tabu}\par}%
6393 +
6394 +% Used for member group headers
6395 +\newenvironment{Indent}{%
6396 + \begin{list}{}{%
6397 + \setlength{\leftmargin}{0.5cm}%
6398 + }%
6399 + \item[]\ignorespaces%
6400 +}%
6401 + \unskip%
6402 + \end{list}%
6403 +}
6404 +
6405 +% Used when hyperlinks are turned off
6406 +\newcommand{\doxyref}[3]{%
6407 + \textbf{\#1} (\textnormal{\#2}\, \pageref{\#3})%
6408 +}
6409 +
6410 +% Used to link to a table when hyperlinks are turned on
6411 +\newcommand{\doxytablelink}[2]{%
6412 + \ref{\#1}%
6413 +}
6414 +
6415 +% Used to link to a table when hyperlinks are turned off
6416 +\newcommand{\doxytableref}[3]{%
6417 + \ref{\#3}%
6418 +}
6419 +
6420 +% Used by @addindex
6421 +\newcommand{\lcurly}{\{}
6422 +\newcommand{\rcurly}{\}}
6423 +
6424 +% Colors used for syntax highlighting
6425 +\definecolor{comment}{rgb}{0.5,0.0,0.0}
6426 +\definecolor{keyword}{rgb}{0.0,0.5,0.0}
6427 +\definecolor{keywordtype}{rgb}{0.38,0.25,0.125}
6428 +\definecolor{keywordflow}{rgb}{0.88,0.5,0.0}
6429 +\definecolor{preprocessor}{rgb}{0.5,0.38,0.125}
6430 +\definecolor{stringliteral}{rgb}{0.0,0.125,0.25}
6431 +\definecolor{charliteral}{rgb}{0.0,0.5,0.5}
6432 +\definecolor{vhldigit}{rgb}{1.0,0.0,1.0}
6433 +\definecolor{vhldkeyword}{rgb}{0.43,0.0,0.43}
6434 +\definecolor{vhdllogic}{rgb}{1.0,0.0,0.0}
6435 +\definecolor{vhdlchar}{rgb}{0.0,0.0,0.0}
6436 +
6437 +% Color used for table heading
6438 +\newcommand{\tableheadbgcolor}{lightgray}%
6439 +
6440 +% Version of hypertarget with correct landing location
6441 +\newcommand{\Hypertarget}[1]{\Hy@raisedlink{\hypertarget{\#1}}}%
6442 +
6443 +% possibility to have sections etc. be within the margins
6444 +% unfortunately had to copy part of book.cls and add \raggedright
6445 +\makeatletter
6446 +\newcommand{\doxysection}{\@startsection {section}{1}{\z@}%
6447 + {-3.5ex \@plus -1ex \@minus -.2ex}%
6448 + {2.3ex \@plus .2ex}%
6449 + {\raggedright\normalfont\Large\bfseries}}
6450 +\newcommand{\doxysubsection}{\@startsection{subsection}{2}{\z@}%
6451 + {-3.25ex\@plus -1ex \@minus -.2ex}%
6452 + {1.5ex \@plus .2ex}%
6453 + {\raggedright\normalfont\large\bfseries}}
6454 +\newcommand{\doxysubsubsection}{\@startsection{subsubsection}{3}{\z@}%
6455 + {-3.25ex\@plus -1ex \@minus -.2ex}%
6456 + {1.5ex \@plus .2ex}%
6457 + {\raggedright\normalfont\normalsize\bfseries}}
6458 +\newcommand{\doxyparagraph}{\@startsection{paragraph}{4}{\z@}%
6459 + {3.25ex \@plus 1ex \@minus .2ex}%
6460 + {-1em}%
6461 + {\raggedright\normalfont\normalsize\bfseries}}
6462 +\newcommand{\doxysubparagraph}{\@startsection{subparagraph}{5}{\parindent}%
6463 + {3.25ex \@plus 1ex \@minus .2ex}%
6464 + {-1em}%
6465 + {\raggedright\normalfont\normalsize\bfseries}}
6466 +\makeatother
6467 +% Define caption that is also suitable in a table
6468 +\makeatletter
6469 +\def\doxyfigcaption{%
6470 +\H@refstepcounter{figure}%
6471 +\@dblarg{\caption{figure}}}%
6472 +\makeatother
6473 +
6474 +% Define alpha enumerative names for counters > 26
6475 +\makeatletter
6476 +\def\enumalphalphcnt#1{\expandafter\@enumalphalphcnt\csname c@#1\endcsname}
6477 +\def\@enumalphalphcnt#1{\alphalph{#1}}
6478 +\def\enumAlphaAlphcnt#1{\expandafter\enumAlphaAlphcnt\csname c@#1\endcsname}
6479 +\def\@enumAlphaAlphcnt#1{\AlphAlph{#1}}
6480 +\makeatother
6481 +\AddEnumerateCounter{\enumalphalphcnt}{\@enumalphalphcnt}{aa}
6482 +\AddEnumerateCounter{\enumAlphaAlphcnt}{\@enumAlphaAlphcnt}{AA}
6483 diff --git a/footer.tex b/footer.tex
6484 new file mode 100644
6485 index 0000000..384988f
6486 --- /dev/null
6487 +++ b/footer.tex
6488 @@ -0,0 +1,51 @@

```

```

6489 +% Latex footer for doxygen 1.9.2
6490 +--- End generated contents ---
6491 +
6492 +%BEGIN CITATIONS_PRESENT
6493 + % Bibliography
6494 + \newpage
6495 + \phantomsection
6496 +
6497 +%BEGIN !PDF_HYPERLINKS
6498 + \clearemptydoublepage
6499 +%BEGIN COMPACT_LATEX
6500 + \addcontentsline{toc}{section}{\latexcitereference}
6501 +%END COMPACT_LATEX
6502 +%BEGIN !COMPACT_LATEX
6503 + \addcontentsline{toc}{chapter}{\latexcitereference}
6504 +%END !COMPACT_LATEX
6505 + \printindex
6506 +%END !PDF_HYPERLINKS
6507 +
6508 + \bibliographystyle{\latexbibstyle}
6509 + \bibliography{\latexbibfiles}
6510 +%BEGIN PDF_HYPERLINKS
6511 +%BEGIN COMPACT_LATEX
6512 + \addcontentsline{toc}{section}{\latexcitereference}
6513 +%END COMPACT_LATEX
6514 +%BEGIN !COMPACT_LATEX
6515 + \addcontentsline{toc}{chapter}{\latexcitereference}
6516 +%END !COMPACT_LATEX
6517 +%END PDF_HYPERLINKS
6518 +
6519 +%END CITATIONS_PRESENT
6520 +
6521 +% Index
6522 +%BEGIN !COMPACT_LATEX
6523 + \backmatter
6524 +%END !COMPACT_LATEX
6525 + \newpage
6526 + \phantomsection
6527 + \clearemptydoublepage
6528 +%BEGIN COMPACT_LATEX
6529 + \addcontentsline{toc}{section}{\indexname}
6530 +%END COMPACT_LATEX
6531 +%BEGIN !COMPACT_LATEX
6532 + \addcontentsline{toc}{chapter}{\indexname}
6533 +%END !COMPACT_LATEX
6534 + \printindex
6535 +
6536 +% Required for some languages (in combination with latexdocumentpre from the header)
6537 +$latexdocumentpost
6538 +\end{document}
6539 +
6540 diff --git a/header.tex b/header.tex
6541 new file mode 100644
6542 index 0000000..4a51e52
6543 --- /dev/null
6544 +++ b/header.tex
6545 @@ -0,0 +1,250 @@
6546 +% Latex header for doxygen 1.9.2
6547 + % Handle batch mode
6548 +%BEGIN LATEX_BATCHMODE
6549 + \batchmode
6550 +%END LATEX_BATCHMODE
6551 +
6552 + % to overcome problems with too many open files
6553 + \let\mypdfximage\pdfximage\def\pdfximage{\immediate\mypdfximage}
6554 +
6555 + % Set document class depending on configuration
6556 +%BEGIN COMPACT_LATEX
6557 + \documentclass[twoside]{article}
6558 +%END COMPACT_LATEX
6559 +%BEGIN !COMPACT_LATEX
6560 + \documentclass[twoside]{book}
6561 +%END !COMPACT_LATEX
6562 +
6563 + %% moved from doxygen.sty due to workaround for LaTeX 2019 version and unmaintained tabu package
6564 + \usepackage{ifthen}
6565 + \ifx\requestedLaTeXdate\undefined
6566 + \usepackage{array}
6567 + \else
6568 + \usepackage{array}[=2016-10-06]
6569 + \fi
6570 + %%
6571 +
6572 + % Packages required by doxygen
6573 + \usepackage{fixltx2e} % for \textsubscript
6574 + \usepackage{doxygen}
6575 +
6576 + $extralateststylesheet
6577 +
6578 + \usepackage{graphicx}
6579 + \usepackage[utf8]{inputenc}
6580 + \usepackage{makeidx}
6581 + \PassOptionsToPackage{warn}{textcomp}
6582 + \usepackage{textcomp}
6583 + \usepackage[nointegrals]{wasysym}
6584 + \usepackage{ifxetex}
6585 +
6586 + % NLS support packages
6587 + $languagesupport
6588 +
6589 + % Define default fonts
6590 + % Font selection
6591 +%BEGIN LATEX_FONTENC
6592 + \usepackage[$latexfontenc]{fontenc}
6593 +%END LATEX_FONTENC
6594 +
6595 + % set main and monospaced font
6596 + $latexfont
6597 +
6598 + \usepackage{sectsty}
6599 + \allsectionsfont{%
6600 + \fontseries{bc}\selectfont%
6601 + \color{darkgray}%
6602 + }
6603 + \renewcommand{\DoxyLabelFont}{%
6604 + \fontseries{bc}\selectfont%

```

```

6605 + \color{darkgray}%
6606 + }
6607 + \newcommand{\+}{\discretionary{\mbox{\scriptsize$\hookleftarrow$}}{\+}}
6608 +
6609 + % Arguments of doxygenemoji:
6610 + % 1) ':\text:>' form of the emoji, already LaTeX-escaped
6611 + % 2) file with the name of the emoji without the .png extension
6612 + % in case image exist use this otherwise use the ':\text:>' form
6613 + \newcommand{\doxygenemoji}[2]{%
6614 + \IfFileExists{$latexemojidirectory/#2.png}{\raisebox{-0.1em}{\includegraphics[height=0.9em]{$latexemojidirectory
6615 + /#2.png}}}{#1}%
6616 + }
6617 +
6618 + % Page & text layout
6619 + \usepackage{geometry}
6620 + \geometry{%
6621 + $papertype,%
6622 + top=2.5cm,%
6623 + bottom=2.5cm,%
6624 + left=2.5cm,%
6625 + right=2.5cm%
6626 + }
6627 +
6628 + % Allow a bit of overflow to go unnoticed by other means
6629 + \tolerance=750
6630 + \hfuzz=15pt
6631 + \hbadness=750
6632 + \setlength{\emergencystretch}{15pt}
6633 + \setlength{\parindent}{0cm}
6634 + \newcommand{\doxynormalparskip}{\setlength{\parskip}{3ex plus 2ex minus 2ex}}
6635 + \newcommand{\doxycparskip}{\setlength{\parskip}{1ex plus 0ex minus 0ex}}
6636 + \doxynormalparskip
6637 + % Redefine paragraph/subparagraph environments, using sectsty fonts
6638 + \makeatletter
6639 + \renewcommand{\paragraph}{%
6640 + \@startsection{paragraph}{4}{0ex}{-1.0ex}{1.0ex}{%
6641 + \normalfont\normalsize\bfseries\SS@parafont}%
6642 + }%
6643 + }
6644 + \renewcommand{\subparagraph}{%
6645 + \@startsection{subparagraph}{5}{0ex}{-1.0ex}{1.0ex}{%
6646 + \normalfont\normalsize\bfseries\SS@subparafont}%
6647 + }%
6648 + }
6649 + \makeatother
6650 +
6651 + \makeatletter
6652 + \newcommand{\hrulefilll}{\leavevmode\leaders\hrule\hskip 0pt plus 1filll\kern\z@}
6653 + \makeatother
6654 +
6655 + % Headers & footers
6656 + \usepackage{fancyhdr}
6657 + \pagestyle{fancyplain}
6658 + \renewcommand{\footrulewidth}{0.4pt}
6659 +
6660 + \fancypagestyle{fancyplain}{
6661 + \fancyhf{}
6662 + \fancyhead[LE, RO]{\bfseries\thepage}
6663 + \fancyhead[LO]{\bfseries\rightmark}
6664 + \fancyhead[RE]{\bfseries\leftmark}
6665 + % \fancyfoot[LO, RE]{\bfseries\scriptsize $generatedby Doxygen }
6666 + }
6667 +
6668 + \fancypagestyle{plain}{
6669 + \fancyhf{}
6670 + % \fancyfoot[LO, RE]{\bfseries\scriptsize $generatedby Doxygen }
6671 + \renewcommand{\headrulewidth}{0pt}
6672 + }
6673 +
6674 + \pagestyle{fancyplain}
6675 +
6676 + %%BEGIN !COMPACT_LATEX
6677 + \renewcommand{\chaptermark}[1]{%
6678 + \markboth{#1}{}%
6679 + }
6680 + %%END !COMPACT_LATEX
6681 + \renewcommand{\sectionmark}[1]{%
6682 + \markright{\thesection\ #1}%
6683 + }
6684 +
6685 + % ToC, LoF, LoT, bibliography, and index
6686 + % Indices & bibliography
6687 + \usepackage{natbib}
6688 + \usepackage[titles]{tocloft}
6689 + \setcounter{tocdepth}{3}
6690 + \setcounter{secnumdepth}{5}
6691 +
6692 + % creating indexes
6693 + $makeindex
6694 +
6695 + $extralatepackages
6696 +
6697 + $latexspecialformulachars
6698 +
6699 + %%BEGIN FORMULA_MACROFILE
6700 + \input{$formulamacrofile}
6701 + %%END FORMULA_MACROFILE
6702 +
6703 + % Hyperlinks
6704 + %%BEGIN PDF_HYPERLINKS
6705 + % Hyperlinks (required, but should be loaded last)
6706 + \ifpdf
6707 + \usepackage[pdftex,pagebackref=true]{hyperref}
6708 + \else
6709 + \ifxetex
6710 + \usepackage[pagebackref=true]{hyperref}
6711 + \else
6712 + \usepackage[ps2pdf,pagebackref=true]{hyperref}
6713 + \fi
6714 + \fi
6715 +
6716 + \hypersetup{%
6717 + colorlinks=true,%
6718 + linkcolor=blue,%
6719 + citecolor=blue,%

```

```

6720 +     unicode%
6721 + }
6722 +
6723 +%%END PDF_HYPERLINKS
6724 +
6725 + % Custom commands used by the header
6726 + % Custom commands
6727 + \newcommand{\clearemptydoublepage}{%
6728 + \newpage{\pagestyle{empty}\cleardoublepage}%
6729 + }
6730 +
6731 + % caption style definition
6732 + \usepackage{caption}
6733 + \captionsetup{labelsep=space,justification=centering,font={bf},singlelinecheck=off,skip=4pt,position=top}
6734 +
6735 +
6736 + % in page table of contents
6737 + \usepackage{etoc}
6738 + \etocsettocstyle{\doxytocparskip}{\doxynormalparskip}
6739 +
6740 + % prevent numbers overlap the titles in toc
6741 + \renewcommand{\numberline}[1]{#1-}
6742 +
6743 + % End of preamble, now comes the document contents
6744 + %===== C O N T E N T S =====
6745 +
6746 + \begin{document}
6747 + \raggedbottom
6748 +
6749 + $latexdocumentpre
6750 +
6751 + % Titlepage & ToC
6752 +%%BEGIN PDF_HYPERLINKS
6753 +%%BEGIN USE_PDFLATEX
6754 + % To avoid duplicate page anchors due to reuse of same numbers for
6755 + % the index (be it as roman numbers)
6756 + \hypersetup{pageanchor=false,
6757 +             bookmarksnumbered=true,
6758 +             pdfencoding=unicode
6759 + }
6760 +%%END USE_PDFLATEX
6761 +%%END PDF_HYPERLINKS
6762 + \pagenumbering{alph}
6763 + \begin{titlepage}
6764 + \vspace*{7cm}
6765 + \begin{center}%
6766 + {\Large $title}\\
6767 + \vspace*{1cm}
6768 + {\large $generatedby Doxygen $doxygenversion}\\
6769 +%%BEGIN LATEX_TIMESTAMP
6770 + \vspace*{0.5cm}
6771 + {\small $datetime}
6772 +%%END LATEX_TIMESTAMP
6773 + \end{center}
6774 + \end{titlepage}
6775 +
6776 +%%BEGIN !COMPACT_LATEX
6777 + \clearemptydoublepage
6778 +%%END !COMPACT_LATEX
6779 + \pagenumbering{roman}
6780 +
6781 + \tableofcontents
6782 +%%BEGIN !COMPACT_LATEX
6783 + \clearemptydoublepage
6784 +%%END !COMPACT_LATEX
6785 + \pagenumbering{arabic}
6786 +
6787 +%%BEGIN PDF_HYPERLINKS
6788 +%%BEGIN USE_PDFLATEX
6789 + % re-enable anchors again
6790 + \hypersetup{pageanchor=true}
6791 +%%END USE_PDFLATEX
6792 +%%END PDF_HYPERLINKS
6793 +
6794 + %--- Begin generated contents ---
6795 +
6796 + diff --git a/src/red_black.c b/src/red_black.c
6797 + index c658162..d8e0d40 100644
6798 + --- a/src/red_black.c
6799 + +++ b/src/red_black.c
6800 + @@ -13,6 +13,7 @@ rb_n* rb_cr(int key, void *dado)
6801 + {
6802 +     rb_n *a = (rb_n *) malloc(sizeof(rb_n));
6803 +     a->k = key;
6804 +     /** raiz sempre colorido com preto */
6805 +     a->c = 0;
6806 +     a->d = dado;
6807 +     a->p = a->l = a->r = NULL;
6808 + @@ -41,6 +42,8 @@ int rb_ins_elem(rb_n *a, int k, void *e)
6809 +     it = (rb_n *) malloc(sizeof(rb_n));
6810 +     it->d = e;
6811 +     it->k = k;
6812 +     /** cor vermelha por default na insercao */
6813 +     it->c = 1;
6814 +     it->l = it->r = NULL;
6815 +
6816 +     if (r1 == 0)
6817 + diff --git a/src/red_black.h b/src/red_black.h
6818 + index b16d42f..4db5559 100644
6819 + --- a/src/red_black.h
6820 + +++ b/src/red_black.h
6821 + @@ -22,12 +22,14 @@
6822 + * @c cor (c de color ou cor, 0-black, 1-red)
6823 + * @t tipo (t de tipo ou type)
6824 + * @d dado (de de data ou dados)
6825 + * @np numero de nohs pretos ate a folha
6826 + * @l @r @p l-efit filho a esquerda, r-right filho a direita e p-parent, noh pai
6827 + */
6828 + typedef struct rb_n {
6829 +     int k;
6830 +     char c:1;
6831 +     void *d;
6832 +     unsigned int np;
6833 +     struct rb_n *l, *r, *p;
6834 + } rb_n;
6835 +

```

```

6836
6837 commit b89cc58e414bedb599010b7e6b231a61a87c16eb
6838 Author: Joilnen <joilnen@gmail.com>
6839 Date: Sat Dec 18 14:21:28 2021 -0300
6840
6841 lidando com o bit da coloracao do noh na rb
6842
6843 diff --git a/src/red_black.c b/src/red_black.c
6844 index dbe0d40..75505bc 100644
6845 --- a/src/red_black.c
6846 +++ b/src/red_black.c
6847 @@ -28,11 +28,11 @@ int rb_ins_elem(rb_n *a, int k, void *e)
6848     while (it)
6849     {
6850         rb_n *tmp = it;
6851         char rl = 0;
6852         int rl = 0;
6853         if (k < it->k)
6854             it = it->l, rl = 0 << 0;
6855         else if (k > it->k)
6856             it = it->r, rl = 1 << 0;
6857         else return -1;
6858         printf("rl %d\n", rl);
6859         @@ -46,9 +46,9 @@ int rb_ins_elem(rb_n *a, int k, void *e)
6860             it->c = 1;
6861             it->l = it->r = NULL;
6862             if (rl == 0)
6863                 if (rl & 0)
6864                     tmp->l = it;
6865             else if (rl == 1)
6866                 tmp->r = it;
6867             return 0;
6868     }
6869 diff --git a/src/red_black.h b/src/red_black.h
6870 index 4db5559..cf4306a 100644
6871 --- a/src/red_black.h
6872 +++ b/src/red_black.h
6873 @@ -27,7 +27,7 @@
6874     typedef struct rb_n {
6875         int k;
6876         char c:1;
6877         unsigned char c:1;
6878         void *d;
6879         unsigned int np;
6880         struct rb_n *l, *r, *p;
6881     }
6882 commit c8b58499800984bac7f9912c9853ef92f5f19395
6883 Author: Joilnen <joilnen@gmail.com>
6884 Date: Sat Dec 18 15:05:28 2021 -0300
6885
6886 adicionando alguns comentarios para documentacao nos fontes
6887
6888 diff --git a/src/red_black.c b/src/red_black.c
6889 index 75505bc..3cbb909 100644
6890 --- a/src/red_black.c
6891 +++ b/src/red_black.c
6892 @@ -1,4 +1,4 @@
6893 /**
6894  * aluno: joilnen leite
6895  * arquivo separado do main, como especificado
6896  * de implementacao das funcoes da redblack
6897 diff --git a/src/red_black.h b/src/red_black.h
6898 index cf4306a..3cf9eca 100644
6899 --- a/src/red_black.h
6900 +++ b/src/red_black.h
6901 @@ -1,12 +1,12 @@
6902 #ifndef REDBLACK_H
6903 #define REDBLACK_H
6904 /**
6905  * aluno: joilnen leite
6906  */
6907 #include <stdlib.h>
6908 /**
6909  * rubro negra
6910  * 0 - todo noh eh vermelho ou preto
6911  * 1 - raiz sempre preta
6912 @@ -40,25 +40,91 @@ typedef struct rb_n {
6913     rb_n* rb_cr(int key, void *dado);
6914 /**
6915  * @a noh raiz da arvore
6916  * @k chave do elemento a ser inserido
6917  * @e tipo genperico de dado a ser guardado neste noh
6918  */
6919 int rb_ins_elem(rb_n *a, int k, void *e);
6920 /**
6921  * @a noh raiz da arvore
6922  * @k chave do elemento a ser excluido
6923  */
6924 int rb_exclui_elem(rb_n *a, int k);
6925 /**
6926  * @a noh raiz da arvore
6927  * @k chave do elemento a ser checado
6928  * retorna 0 se o lelemento nao existe na arvore
6929  * ou diferente de zero caso exista
6930  */

```

```

6952 int rb_exite_elem(rb_n *a, int k);
6953
6954 /**
6955  * @a noh raiz da arvore
6956  * @b noh raiz da arvore a ser operada
6957  * * retorna ponteiro para uma nova arvore fruto da interseccao
6958  * das duas arvores @a e @b
6959  */
6960 rb_n* rb_intersec(rb_n *a, rb_n *b);
6961 +
6962 /**
6963  * @a noh raiz da arvore
6964  * @b noh raiz da arvore a ser operada
6965  * * retorna ponteiro para uma nova arvore fruto da uniao
6966  * das duas arvores @a e @b
6967  */
6968 rb_n* rb_une(rb_n *a, rb_n *b);
6969 +
6970 /**
6971  * @a noh raiz da arvore
6972  * @b noh raiz da arvore a ser operada
6973  * * retorna ponteiro para uma nova arvore fruto da subtracao
6974  * das duas arvores @a e @b
6975  */
6976 rb_n* rb_subtrai(rb_n *a, rb_n *b);
6977
6978 /**
6979  * * deleta, e desaloca recursos de memoria
6980  * para a arvore apontada por @a
6981  */
6982 int rb_destroi(rb_n **a);
6983
6984 /**
6985  * * troca a cor de um noh apontado por @a
6986  */
6987 int rb_troca_cor(rb_n *a);
6988 +
6989 /**
6990  * * rotaciona subarvore a esquerda apartir do noh @a
6991  */
6992 int rb_rot_esq(rb_n *r, rb_n *w);
6993 +
6994 /**
6995  * * rotaciona subarvore a direita apartir do noh @a
6996  */
6997 int rb_rot_dir(rb_n *r, rb_n *w);
6998
6999 -extern void (*rb_impr_elem)(rb_n *);
7000
7001 /**
7002  * * imprime elemento como definido pelo o codigo cliente da biblioteca
7003  */
7004 xtern void (*rb_impr_elem)(rb_n *);
7005 +
7006 /**
7007  * * seta a rotina de impressao do dado criado pelo cliente
7008  */
7009 int rb_set_a_impr_elem(void (*i)(rb_n *n));
7010
7011 /**
7012  * * imprime arvore @a
7013  */
7014 void rb_pr(rb_n *a);
7015
7016 #endif
7017
7018 commit cfa71cb32fd85286fdc6b4fc7981c4776c3180a4
7019 Author: Joilnen <joilnen@gmail.com>
7020 Date: Sat Dec 18 16:05:59 2021 -0300
7021
7022 melhorando a documentacao nos comentarios
7023
7024 diff --git a/src/red_black.c b/src/red_black.c
7025 index 3cbb909..5e1efa9 100644
7026 --- a/src/red_black.c
7027 +++ b/src/red_black.c
7028 @@ -13,7 +13,7 @@ rb_n* rb_cr(int key, void *dado)
7029 {
7030     rb_n *a = (rb_n *) malloc(sizeof(rb_n));
7031     a->k = key;
7032     - /** raiz sempre colorido com preto */
7033     + /** raiz sempre colorida com preto */
7034     a->c = 0;
7035     a->d = dado;
7036     a->p = a->l = a->r = NULL;
7037 diff --git a/src/red_black.h b/src/red_black.h
7038 index 3cf9eca..a5048cf 100644
7039 --- a/src/red_black.h
7040 +++ b/src/red_black.h
7041 @@ -17,13 +17,14 @@
7042
7043 /**
7044  * composicao do noh da RB
7045  * @rb_n noh red black
7046  * @k chave (k de key)
7047  * @c cor (c de color ou cor, 0-black, 1-red)
7048  * @t tipo (t de tipo ou type)
7049  * @d dado (de de data ou dados)
7050  * @np numero de nohs pretos ateh a folha
7051  * @l @r @p l-left filho a esquerda, r-right filho a direita e p-parent, noh pai
7052  * @param rb_n noh red black
7053  * @param k chave (k de key)
7054  * @param c cor (c de color ou cor, 0-black, 1-red)
7055  * @param d dado (d de data ou dados)
7056  * @param np numero de nohs pretos ateh a folha
7057  * @param l left, ponteiro para filho a esquerda
7058  * @param r right, ponteiro para filho a direita
7059  * @param p parent, ponteiro para noh pai
7060  */
7061 typedef struct rb_n {
7062     int k;
7063     @@ -34,81 +35,80 @@ typedef struct rb_n {
7064 } rb_n;
7065
7066 /**
7067  * * cria o primeiro noh, o raiz da arvore rb
7068  * * serah o primeiro elemento criado na arvore

```

```

7068 - * e o primeiro do conjunto ordenado
7069 + * cria o primeiro noh, a raiz da arvore rb,
7070 */
7071 rb_n* rb_cr(int key, void *dado);
7072
7073 /**
7074 - * @a noh raiz da arvore
7075 + * @param a noh raiz da arvore
7076 * retorna numero de elementos da arvore
7077 */
7078 unsigned int rb_nelem(rb_n *a);
7079
7080 /**
7081 - * @a noh raiz da arvore
7082 - * @k chave do elemento a ser inserido
7083 - * @e tipo genperico de dado a ser guardado neste noh
7084 + * @param a noh raiz da arvore
7085 + * @param k chave do elemento a ser inserido
7086 + * @param e tipo generico de dado a ser guardado neste noh
7087 */
7088 int rb_ins_elem(rb_n *a, int k, void *e);
7089
7090 /**
7091 - * @a noh raiz da arvore
7092 - * @k chave do elemento a ser excluido
7093 + * @param a noh raiz da arvore
7094 + * @param k chave do elemento a ser excluido
7095 */
7096 int rb_exclui_elem(rb_n *a, int k);
7097
7098 /**
7099 - * @a noh raiz da arvore
7100 - * @k chave do elemento a ser checado
7101 + * @param a noh raiz da arvore
7102 + * @param k chave do elemento a ser checado
7103 * retorna 0 se o lelemento nao existe na arvore
7104 * ou diferente de zero caso exista
7105 */
7106 int rb_exite_elem(rb_n *a, int k);
7107
7108 /**
7109 - * @a noh raiz da arvore
7110 - * @b noh raiz da arvore a ser operada
7111 - * retorna ponteir para uma nova arvore fruto da interseccao
7112 - * das duas arvores @a e @b
7113 + * @param a noh raiz da arvore
7114 + * @param b noh raiz da arvore a ser operada
7115 + * @return retorna ponteiro para uma nova arvore fruto da interseccao
7116 + * das duas arvores @param a e @param b
7117 */
7118 rb_n* rb_intersec(rb_n *a, rb_n *b);
7119
7120 /**
7121 - * @a noh raiz da arvore
7122 - * @b noh raiz da arvore a ser operada
7123 + * @param a noh raiz da arvore
7124 + * @param b noh raiz da arvore a ser operada
7125 * retorna ponteir para uma nova arvore fruto da uniao
7126 - * das duas arvores @a e @b
7127 + * das duas arvores @param a e @param b
7128 */
7129 rb_n* rb_une(rb_n *a, rb_n *b);
7130
7131 /**
7132 - * @a noh raiz da arvore
7133 - * @b noh raiz da arvore a ser operada
7134 + * @param a noh raiz da arvore
7135 + * @param b noh raiz da arvore a ser operada
7136 * retorna ponteir para uma nova arvore fruto da subtracao
7137 - * das duas arvores @a e @b
7138 + * das duas arvores @param a e @param b
7139 */
7140 rb_n* rb_subtrai(rb_n *a, rb_n *b);
7141
7142 /**
7143 * deleta, e desaloca recursos de memoria
7144 - * para a arvore apontada por @a
7145 + * para a arvore apontada por @param a
7146 */
7147 int rb_destroi(rb_n **a);
7148
7149 /**
7150 - * troca a cor de um noh apontado por @a
7151 + * troca a cor de um noh apontado por @param a
7152 */
7153 int rb_troca_cor(rb_n *a);
7154
7155 /**
7156 - * rotaciona subarvore a esquerda apartir do noh @a
7157 + * rotaciona subarvore a esquerda apartir do noh @param a
7158 */
7159 int rb_rot_esq(rb_n *r, rb_n *w);
7160
7161 /**
7162 - * rotaciona subarvore a direita apartir do noh @a
7163 + * rotaciona subarvore a direita apartir do noh
7164 + * @param a noh
7165 */
7166 int rb_rot_dir(rb_n *r, rb_n *w);
7167
7168 @@ -123,7 +123,7 @@ xtern void (*rb_impr_elem)(rb_n *);
7169 int rb_sete_impr_elem(void (*i)(rb_n *n));
7170
7171 /**
7172 - * imprime arvore @a
7173 + * imprime arvore @param a
7174 */
7175 void rb_pr(rb_n *a);
7176
7177
7178 commit a71d7223b5a14740b069b1129d63ff593b7e3095
7179 Author: Joilnen <joilnen@gmail.com>
7180 Date: Sat Dec 18 16:12:33 2021 -0300
7181
7182 conserdanto um pequeno erro de digitacao
7183

```

```

7184 diff --git a/src/red_black.h b/src/red_black.h
7185 index a5048cf..c09ae56 100644
7186 --- a/src/red_black.h
7187 +++ b/src/red_black.h
7188 @@ -115,7 +115,7 @@ int rb_rot_dir(rb_n *r, rb_n *w);
7189 /**
7190  * imprime elemento como definido pelo o codigo cliente da biblioteca
7191  */
7192 -extern void (*rb_impr_elem)(rb_n *);
7193 +extern void (*rb_impr_elem)(rb_n *);
7194
7195 /**
7196  * seta a rotina de impressao do dado criado pelo cliente
7197  */
7198 commit 407fbad6094efb087f5ea41fcfb424c1699a4216
7199 Author: Joilnen <joilnen@gmail.com>
7200 Date: Sun Dec 19 00:36:30 2021 -0300
7201
7202     changing makefile
7203
7204 diff --git a/src/makefile b/src/makefile
7205 index 6fe7410..d201161 100644
7206 --- a/src/makefile
7207 +++ b/src/makefile
7208 @@ -1,17 +1,17 @@
7209 #
7210 -# Aluno: Joilnen Leite
7211 +# Aluno:
7212 #
7213
7214 C=gcc
7215 S=main.o conjunto_ordenado.o red_black.o
7216 O=main.o conjunto_ordenado.o red_black.o
7217 -B=rodeme
7218 +E=rodeme
7219 D=-g
7220
7221 .c.o:
7222 $(C) -c $(D) $<
7223 all: $(O)
7224 - $(C) -o $(B) $(O)
7225 + $(C) -o $(E) $(O)
7226 clean:
7227 - rm -f *.o $(B)
7228 + rm -f *.o $(E)
7229
7230 diff --git a/src/red_black.c b/src/red_black.c
7231 index 5e1efa9..8b13789 100644
7232 --- a/src/red_black.c
7233 +++ b/src/red_black.c
7234 @@ -1,144 +1 @@
7235 -/*
7236 - * aluno: joilnen leite
7237 - * arquivo separado do main, como especificado
7238 - * de implementacao das funcoes da redblack
7239 - * utilizada no conjunto ordenado
7240 - */
7241 -#include "red_black.h"
7242 -#include <stdio.h>
7243 -
7244 -void (*rb_impr_elem)(rb_n *);
7245 -
7246 -rb_n* rb_cr(int key, void *dado)
7247 -{
7248 -    rb_n *a = (rb_n *) malloc(sizeof(rb_n));
7249 -    a->k = key;
7250 -    /** raiz sempre colorida com preto */
7251 -    a->c = 0;
7252 -    a->d = dado;
7253 -    a->p = a->l = a->r = NULL;
7254 -
7255 -    return a;
7256 -}
7257 -
7258 -int rb_ins_elem(rb_n *a, int k, void *e)
7259 -{
7260 -    rb_n *it = a;
7261 -    printf("k %d it->k %d\n", k, it->k);
7262 -    while (it)
7263 -    {
7264 -        rb_n *tmp = it;
7265 -        int rl = 0;
7266 -        if (k < it->k)
7267 -            it = it->l, rl = !0;
7268 -        else if (k > it->k)
7269 -            it = it->r, rl = !0;
7270 -        else return -1;
7271 -
7272 -        printf("rl %d\n", rl);
7273 -
7274 -        if (!it)
7275 -        {
7276 -            it = (rb_n *) malloc(sizeof(rb_n));
7277 -            it->d = e;
7278 -            it->k = k;
7279 -            /** cor vermelha por default na insercao */
7280 -            it->c = 1;
7281 -            it->l = it->r = NULL;
7282 -
7283 -            if (rl & 0)
7284 -                tmp->l = it;
7285 -            else
7286 -                tmp->r = it;
7287 -            return 0;
7288 -        }
7289 -    }
7290 -
7291 -    return -1;
7292 -}
7293 -
7294 -int rb_exclui_elem(rb_n *a, int k)
7295 -{
7296 -    return 0;
7297 -}
7298 -
7299 -rb_n* rb_intersec(rb_n *a, rb_n *b)

```



```

7300 -{
7301 -     return NULL;
7302 -}
7303 -
7304 -rb_n* rb_une(rb_n *a, rb_n *b)
7305 -{
7306 -     return NULL;
7307 -}
7308 -
7309 -rb_n* rb_subtrai(rb_n *a, rb_n *b)
7310 -{
7311 -     return NULL;
7312 -}
7313 -
7314 -int rb_destroi(rb_n **a)
7315 -{
7316 -     return 0;
7317 -}
7318 -
7319 -int rb_rot_esq(rb_n *r, rb_n *w)
7320 -{
7321 -     rb_n *v = w->r;
7322 -     w->r = v->l;
7323 -     if (v->l)
7324 -         v->l->p = w;
7325 -     v->p = w->p;
7326 -     if (!w->p)
7327 -         r = v;
7328 -     else
7329 -     {
7330 -         if (w == w->p->l)
7331 -             w->p->l = v;
7332 -         else
7333 -             w->p->r = v;
7334 -     }
7335 -     v->l = w;
7336 -     w->p = v;
7337 -     return 0;
7338 -}
7339 -
7340 -int rb_rot_dir(rb_n *r, rb_n *w)
7341 -{
7342 -     rb_n *v = w->l;
7343 -     w->l = v->r;
7344 -     if (v->r)
7345 -         v->r->p = w;
7346 -     v->p = w->p;
7347 -     if (!w->p)
7348 -         r = v;
7349 -     else
7350 -     {
7351 -         if (w == w->p->r)
7352 -             w->p->r = v;
7353 -         else
7354 -             w->p->l = v;
7355 -     }
7356 -     v->r = w;
7357 -     w->p = v;
7358 -     return 0;
7359 -}
7360 -
7361 -int rb_sete_impr_elem(void (*i)(rb_n *n))
7362 -{
7363 -     rb_impr_elem = i;
7364 -}
7365 -
7366 -void rb_pr(rb_n *a)
7367 -{
7368 -     if (a)
7369 -     {
7370 -         printf("k:%dc:%d\n", a->k, a->c);
7371 -         rb_pr(a->l);
7372 -         rb_pr(a->r);
7373 -     }
7374 -}
7375 -
7376 -
7377 -
7378 -
7379 -commit 6f728d3673b2ac981c030adfd1fc87f88e4c0611
7380 -Author: Joilnen <joilnen@gmail.com>
7381 -Date: Sun Dec 19 00:46:07 2021 -0300
7382 -
7383 -recuperando red_black.c foi perdido por algum problema no nvim
7384 -
7385 -diff --git a/src/red_black.c b/src/red_black.c
7386 -index 8b13789..e0b04e8 100644
7387 ---- a/src/red_black.c
7388 -+++ b/src/red_black.c
7389 -@@ -1,138 @@
7390 -+/*
7391 -+ * aluno: joilnen leite
7392 -+ * arquivo separado do main, como especificado
7393 -+ * de implementacao das funcoes da redblack
7394 -+ * utilizada no conjunto ordenado
7395 -+ */
7396 -+#include "red_black.h"
7397 -+#include <stdio.h>
7398 -+
7399 -+void (*rb_impr_elem)(rb_n *);
7400 -+
7401 -+rb_n* rb_cr(int key, void *dado)
7402 -+{
7403 -+     rb_n *a = (rb_n *) malloc(sizeof(rb_n));
7404 -+     a->k = key;
7405 -+     /* raiz sempre colorida com preto */
7406 -+     a->c = 0;
7407 -+     a->d = dado;
7408 -+     a->p = a->l = a->r = NULL;
7409 -+     return a;
7410 -+}
7411 -+
7412 -+int rb_ins_elem(rb_n *a, int k, void *e)
7413 -+{

```

```

7416 +   rb_n *it = a;
7417 +   printf("k %d it->k %d\n", k, it->k);
7418 +   while (it)
7419 +   {
7420 +       rb_n *tmp = it;
7421 +       int r1 = 0;
7422 +       if (k < it->k)
7423 +           it = it->l, r1 = !0;
7424 +       else if (k > it->k)
7425 +           it = it->r, r1 = !0;
7426 +       else return -1;
7427 +
7428 +       printf("r1 %d\n", r1);
7429 +
7430 +       if (!it)
7431 +       {
7432 +           it = (rb_n *) malloc(sizeof(rb_n));
7433 +           it->d = e;
7434 +           it->k = k;
7435 +           /* cor vermelha por default na insercao */
7436 +           it->c = 1;
7437 +           it->l = it->r = NULL;
7438 +
7439 +           if (r1 & 0)
7440 +               tmp->l = it;
7441 +           else
7442 +               tmp->r = it;
7443 +           return 0;
7444 +       }
7445 +   }
7446 +
7447 +   return -1;
7448 +}
7449 +
7450 +int rb_exclui_elem(rb_n *a, int k)
7451 +{
7452 +   return 0;
7453 +}
7454 +
7455 +rb_n* rb_intersec(rb_n *a, rb_n *b)
7456 +{
7457 +   return NULL;
7458 +}
7459 +
7460 +rb_n* rb_une(rb_n *a, rb_n *b)
7461 +{
7462 +   return NULL;
7463 +}
7464 +
7465 +rb_n* rb_subtrai(rb_n *a, rb_n *b)
7466 +{
7467 +   return NULL;
7468 +}
7469 +
7470 +int rb_destroi(rb_n **a)
7471 +{
7472 +   return 0;
7473 +}
7474 +
7475 +int rb_rot_esq(rb_n *r, rb_n *w)
7476 +{
7477 +   rb_n *v = w->r;
7478 +   w->r = v->l;
7479 +   if (v->l)
7480 +       v->l->p = w;
7481 +   v->p = w->p;
7482 +   if (!w->p)
7483 +       r = v;
7484 +   else if (w == w->p->l)
7485 +       w->p->l = v;
7486 +   else
7487 +       w->p->r = v;
7488 +   v->l = w;
7489 +   w->p = v;
7490 +
7491 +   return 0;
7492 +}
7493 +
7494 +int rb_rot_dir(rb_n *r, rb_n *w)
7495 +{
7496 +   rb_n *v = w->l;
7497 +   w->l = v->r;
7498 +   if (v->r)
7499 +       v->r->p = w;
7500 +   v->p = w->p;
7501 +   if (!w->p)
7502 +       r = v;
7503 +   else if (w == w->p->r)
7504 +       w->p->r = v;
7505 +   else
7506 +       w->p->l = v;
7507 +   v->r = w;
7508 +   w->p = v;
7509 +
7510 +   return 0;
7511 +}
7512 +
7513 +int rb_sete_impr_elem(void (*i)(rb_n *n))
7514 +{
7515 +   rb_impr_elem = i;
7516 +}
7517 +
7518 +void rb_pr(rb_n *a)
7519 +{
7520 +   if (a)
7521 +   {
7522 +       printf("k:%dc:%d\n", a->k, a->c);
7523 +       rb_pr(a->l);
7524 +       rb_pr(a->r);
7525 +   }
7526 +}
7527 +
7528 +
7529 +
7530 +commit 9668c7ddd355dbd3e77431a1b4b6d7fd59e4dba5
7531 +Author: Joilnen <joilnen@gmail.com>

```

```

7532 Date: Sun Dec 19 08:01:31 2021 -0300
7533
7534 insersao baseado no algoritmo mostrado em aula que eh o mesmo do cormen
7535
7536 diff --git a/src/red_black.c b/src/red_black.c
7537 index e0b04e8..ed530c6 100644
7538 --- a/src/red_black.c
7539 +++ b/src/red_black.c
7540 @@ -7,6 +7,13 @@
7541 #include "red_black.h"
7542 #include <stdio.h>
7543
7544 /**
7545  * definicao do que eh considerado RED e BLACK
7546  * neste codigo para maior clareza
7547  */
7548 #define BLACK 0
7549 #define RED 1
7550 +
7551 void (*rb_impr_elem)(rb_n *);
7552
7553 rb_n* rb_cr(int key, void *dado)
7554 @@ -14,7 +21,7 @@ rb_n* rb_cr(int key, void *dado)
7555     rb_n *a = (rb_n *) malloc(sizeof(rb_n));
7556     a->k = key;
7557     /** raiz sempre colorida com preto */
7558     a->c = 0;
7559     a->c = BLACK;
7560     a->d = dado;
7561     a->p = a->l = a->r = NULL;
7562
7563 @@ -57,6 +64,78 @@ int rb_ins_elem(rb_n *a, int k, void *e)
7564     return -1;
7565 }
7566
7567 /**
7568  * inclusao baseado no livro do cormen
7569  */
7570 int rb_insert_cormen(rb_n *t, rb_n *z)
7571 {
7572     rb_n *y = NULL;
7573     rb_n *x = t;
7574     while (x)
7575     {
7576         y = x;
7577         if (z->k < x->k)
7578             x = x->l;
7579         else x = x->r;
7580     }
7581     z->p = y;
7582     if (!y)
7583         t = z;
7584     else if (z->k < y->k)
7585         y->l = z;
7586     else
7587         y->r = z;
7588     z->l = z->r = NULL;
7589     z->c = RED;
7590     /** ajuste */
7591     while (z->p->c == RED)
7592     {
7593         if (z->p == z->p->p->l)
7594         {
7595             y = z->p->p->r;
7596             if (y->c == RED)
7597             {
7598                 z->p->c = BLACK;
7599                 y->p->p->c = RED;
7600                 z = z->p->p;
7601             }
7602             else
7603             {
7604                 if (z == z->p->r)
7605                 {
7606                     z = z->p;
7607                     rb_rot_esq(t, z);
7608                 }
7609                 z->p->c = BLACK;
7610                 z->p->p->c = RED;
7611                 rb_rot_dir(t, z->p->p);
7612             }
7613         }
7614         else
7615         {
7616             y = z->p->p->l;
7617             if (y->c == RED)
7618             {
7619                 z->p->c = BLACK;
7620                 y->p->p->c = RED;
7621                 z = z->p->p;
7622             }
7623             else
7624             {
7625                 if (z == z->p->l)
7626                 {
7627                     z = z->p;
7628                     rb_rot_dir(t, z);
7629                 }
7630                 z->p->c = BLACK;
7631                 z->p->p->c = RED;
7632                 rb_rot_esq(t, z->p->p);
7633             }
7634         }
7635     }
7636     t->c = BLACK;
7637 }
7638
7639 int rb_exclui_elem(rb_n *a, int k)
7640 {
7641     return 0;
7642 }
7643 diff --git a/src/red_black.h b/src/red_black.h
7644 index c09ae56..42c905b 100644
7645 --- a/src/red_black.h
7646 +++ b/src/red_black.h
7647 @@ -52,6 +52,12 @@ unsigned int rb_nelem(rb_n *a);
7648 */

```

```

7648 int rb_ins_elem(rb_n *a, int k, void *e);
7649
7650 /**
7651  * @param a noh raiz da arvore
7652  * @param b elemento a ser inserido
7653  */
7654 +int rb_insert_cormen(rb_n *t, rb_n *z);
7655 +
7656 /**
7657  * @param a noh raiz da arvore
7658  * @param k chave do elemento a ser excluido
7659  */
7660 commit b11b5a14445c3a4e73b523c00740929d55d8dce6
7661 Author: Joilnen <joilnen@gmail.com>
7662 Date: Sun Dec 19 08:52:11 2021 -0300
7663
7664     funcoes baseadas no livro do cormen e nas notas de aula da prof. Luciana Lee
7665
7666 diff --git a/src/red_black.c b/src/red_black.c
7667 index ed530c6..4de5272 100644
7668 --- a/src/red_black.c
7669 +++ b/src/red_black.c
7670 @@ -67,7 +67,7 @@ int rb_ins_elem(rb_n *a, int k, void *e)
7671  /**
7672   * inclusao baseado no livro do cormen
7673   */
7674 -int rb_insert_cormen(rb_n *t, rb_n *z)
7675 +int rb_insert(rb_n *t, rb_n *z)
7676 {
7677     rb_n *y = NULL;
7678     rb_n *x = t;
7679 @@ -136,6 +136,140 @@ int rb_insert_cormen(rb_n *t, rb_n *z)
7680     t->c = BLACK;
7681 }
7682
7683 /**
7684  * fucnao que faz trocas auxiliares na delecao
7685  */
7686 +static int rb_transplant(rb_n *t, rb_n *u, rb_n *v)
7687 +{
7688 +    if (!u->p)
7689 +        t = v;
7690 +    else if (u == u->p->l)
7691 +        u->p->l = v;
7692 +    else
7693 +        u->p->r = v;
7694 +    v->p = u->p;
7695 +}
7696 +
7697 +static rb_n *tree_minimum(rb_n *x)
7698 +{
7699 +    rb_n *a = x;
7700 +    while (a->r)
7701 +        a = a->r;
7702 +    return a;
7703 +}
7704 +
7705 +int rb_delete(rb_n *t, rb_n *z)
7706 +{
7707 +    rb_n *x;
7708 +    rb_n *y = z;
7709 +    unsigned char yoc = y->c;
7710 +    if (!z->l)
7711 +    {
7712 +        x = z->r;
7713 +        rb_transplant(t, z, z->r);
7714 +    }
7715 +    else if (!z->r)
7716 +    {
7717 +        x = z->l;
7718 +        rb_transplant(t, z, z->l);
7719 +    }
7720 +    else
7721 +    {
7722 +        y = tree_minimum(z->r);
7723 +        yoc = y->c;
7724 +        x = y->r;
7725 +        if (y->p == z)
7726 +            x->p = y;
7727 +        else
7728 +        {
7729 +            rb_transplant(t, y, y->r);
7730 +            y->r = z->r;
7731 +            y->r->p = y;
7732 +        }
7733 +        rb_transplant(t, z, y);
7734 +        y->l = z->l;
7735 +        y->l->p = y;
7736 +        y->c = z->c;
7737 +    }
7738 +    if (yoc == BLACK)
7739 +    {
7740 +        rb_n *w;
7741 +        while (x != t && x->c == BLACK)
7742 +        {
7743 +            if (x == x->p->l)
7744 +            {
7745 +                w = x->p->r;
7746 +                if (w->c == RED)
7747 +                {
7748 +                    w->c = BLACK;
7749 +                    x->p->c = RED;
7750 +                    rb_rot_esq(t, x->p);
7751 +                    w = x->p->r;
7752 +                }
7753 +                if (w->l->c == BLACK && w->r->c == BLACK)
7754 +                {
7755 +                    w->c = RED;
7756 +                    x = x->p;
7757 +                }
7758 +            }
7759 +            else
7760 +            {
7761 +                if (w->r->c == BLACK)
7762 +                {
7763 +                    w->l->c = BLACK;

```

```

7764 +             w->c = RED;
7765 +             rb_rot_dir(t, w);
7766 +             w = x->p->r;
7767 +         }
7768 +
7769 +             w->c = x->p->c;
7770 +             x->p->c = BLACK;
7771 +             w->r->c = BLACK;
7772 +             rb_rot_esq(t, x->p);
7773 +             x = t;
7774 +         }
7775 +
7776 +
7777 +     }
7778 +     else
7779 +     {
7780 +         w = x->p->l;
7781 +         if (w->c == RED)
7782 +         {
7783 +             w->c = BLACK;
7784 +             x->p->c = RED;
7785 +             rb_rot_esq(t, x->p);
7786 +             w = x->p->l;
7787 +         }
7788 +
7789 +         if (w->r->c == BLACK && w->l->c == BLACK)
7790 +         {
7791 +             w->c = RED;
7792 +             x = x->p;
7793 +         }
7794 +         else
7795 +         {
7796 +             if (w->l->c == BLACK)
7797 +             {
7798 +                 w->r->c = BLACK;
7799 +                 w->c = RED;
7800 +                 rb_rot_dir(t, w);
7801 +                 w = x->p->l;
7802 +             }
7803 +
7804 +             w->c = x->p->c;
7805 +             x->p->c = BLACK;
7806 +             w->l->c = BLACK;
7807 +             rb_rot_esq(t, x->p);
7808 +             x = t;
7809 +         }
7810 +     }
7811 + }
7812 +
7813 +
7814 + return 0;
7815 +}
7816 +
7817 +int rb_exclui_elem(rb_n *a, int k)
7818 +{
7819 +    return 0;
7820 +}
7821 +diff --git a/src/red_black.h b/src/red_black.h
7822 +index 42c905b..1a97b13 100644
7823 +--- a/src/red_black.h
7824 +++ b/src/red_black.h
7825 +@@ -56,7 +56,13 @@ int rb_ins_elem(rb_n *a, int k, void *e);
7826 + * @param a noh raiz da arvore
7827 + * @param b elemento a ser inserido
7828 + */
7829 +-int rb_insert_cormen(rb_n *t, rb_n *z);
7830 ++int rb_insert(rb_n *t, rb_n *z);
7831 ++
7832 ++/**
7833 ++ * @param a noh raiz da arvore
7834 ++ * @param b elemento a ser deletado
7835 ++ */
7836 ++int rb_delete(rb_n *t, rb_n *z);
7837 ++
7838 ++/**
7839 ++ * @param a noh raiz da arvore
7840 ++ */
7841 +commit b31da3f1a05d652cd2d844713c8be12443e16c64
7842 +Author: Joilnen <joilnen@gmail.com>
7843 +Date: Sun Dec 19 17:52:40 2021 -0300
7844 +
7845 +rb aparentemente funcionando perfeitamente pra 3 nohs testados vou
7846 +por um tag nesse codigo como alfa
7847 +
7848 +diff --git a/src/main.c b/src/main.c
7849 +index 81fbc54..7782288 100644
7850 +--- a/src/main.c
7851 +++ b/src/main.c
7852 +@@ -57,15 +57,32 @@ int main()
7853 + {
7854 +     struct mt1 nome, "nome para teste", sizeof("nome para teste") + 1);
7855 +     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
7856 +     struct mt2 telefone, "999999999", sizeof("999999999") + 1);
7857 +     rb_ins_elem(conjunto_a, 9, &mt1);
7858 +     // rb_ins_elem(conjunto_a, 9, &mt1);
7859 +
7860 +     rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
7861 +     novo_noh->k = 9;
7862 +     novo_noh->c = 1;
7863 +     novo_noh->d = &mt1;
7864 +     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
7865 +
7866 +     rb_insert(conjunto_a, novo_noh);
7867 +
7868 +     meu_tipo mt2;
7869 +     mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
7870 +     struct mt3 nome, "nome para teste", sizeof("nome para teste") + 1);
7871 +     mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
7872 +     struct mt4 telefone, "999999999", sizeof("999999999") + 1);
7873 +     rb_ins_elem(conjunto_a, 11, &mt2);
7874 +     // rb_ins_elem(conjunto_a, 11, &mt2);
7875 +
7876 +     novo_noh = (rb_n *) malloc (sizeof(rb_n));
7877 +     novo_noh->k = 11;
7878 +     novo_noh->c = 1;
7879 +     novo_noh->d = &mt2;
7880 +     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
7881 +     rb_insert(conjunto_a, novo_noh);

```

```

7880 +
7881 +     rb_pr(conjunto_a);
7882 +     rb_delete(conjunto_a, novo_noh);
7883 +     rb_pr(conjunto_a);
7884
7885 /**
7886 diff --git a/src/red_black.c b/src/red_black.c
7887 index 4de5272..83a4095 100644
7888 --- a/src/red_black.c
7889 +++ b/src/red_black.c
7890 @@ -147,7 +147,8 @@ static int rb_transplant(rb_n *t, rb_n *u, rb_n *v)
7891     u->p->l = v;
7892     else
7893         u->p->r = v;
7894     v->p = u->p;
7895     if(v)
7896         v->p = u->p;
7897 }
7898
7899 static rb_n *tree_minimum(rb_n *x)
7900 @@ -228,8 +229,6 @@ int rb_delete(rb_n *t, rb_n *z)
7901     rb_rot_esq(t, x->p);
7902     x = t;
7903 }
7904 -
7905 -
7906     }
7907     else
7908     {
7909 @@ -265,8 +264,8 @@ int rb_delete(rb_n *t, rb_n *z)
7910     }
7911     }
7912 }
7913 -
7914 -
7915 +     } /* if (yoc == BLACK) */
7916 +
7917     return 0;
7918 }
7919
7920
7921 commit 6fb5208352001b15e465d02ad352403f22897e26
7922 Author: Joilnen <joilnen@gmail.com>
7923 Date: Sun Dec 19 20:28:41 2021 -0300
7924
7925     tentando consertar as totacoes, a insercao parece estar colorindo mas n tah rotacionando
7926
7927 diff --git a/src/main.c b/src/main.c
7928 index 7782288..808a8aa 100644
7929 --- a/src/main.c
7930 +++ b/src/main.c
7931 @@ -61,7 +61,7 @@ int main()
7932
7933     rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
7934     novo_noh->k = 9;
7935     novo_noh->c = 1;
7936     novo_noh->c = RED;
7937     novo_noh->d = &mt1;
7938     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
7939
7940 @@ -76,13 +76,56 @@ int main()
7941
7942     novo_noh = (rb_n *) malloc (sizeof(rb_n));
7943     novo_noh->k = 11;
7944     novo_noh->c = 1;
7945     novo_noh->c = RED;
7946     novo_noh->d = &mt2;
7947     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
7948     rb_insert(conjunto_a, novo_noh);
7949
7950 +     meu_tipo mt3;
7951 +     mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
7952 +     strcpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
7953 +     mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
7954 +     strcpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
7955 +     // rb_ins_elem(conjunto_a, 11, &mt2);
7956 +
7957 +     novo_noh = (rb_n *) malloc (sizeof(rb_n));
7958 +     novo_noh->k = 8;
7959 +     novo_noh->c = RED;
7960 +     novo_noh->d = &mt3;
7961 +     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
7962 +     rb_insert(conjunto_a, novo_noh);
7963 +
7964 +     meu_tipo mt4;
7965 +     mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
7966 +     strcpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
7967 +     mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
7968 +     strcpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
7969 +     // rb_ins_elem(conjunto_a, 11, &mt2);
7970 +
7971 +     novo_noh = (rb_n *) malloc (sizeof(rb_n));
7972 +     novo_noh->k = 6;
7973 +     novo_noh->c = RED;
7974 +     novo_noh->d = &mt4;
7975 +     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
7976 +     rb_insert(conjunto_a, novo_noh);
7977 +
7978 +     meu_tipo mt5;
7979 +     mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
7980 +     strcpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
7981 +     mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
7982 +     strcpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
7983 +     // rb_ins_elem(conjunto_a, 11, &mt2);
7984 +
7985 +     novo_noh = (rb_n *) malloc (sizeof(rb_n));
7986 +     novo_noh->k = 5;
7987 +     novo_noh->c = RED;
7988 +     novo_noh->d = &mt5;
7989 +     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
7990 +     rb_insert(conjunto_a, novo_noh);
7991 +
7992     rb_pr(conjunto_a);
7993     rb_delete(conjunto_a, novo_noh);
7994     printf("\n");
7995     rb_pr(conjunto_a);

```

```

7996 /**
7997
7998 diff --git a/src/red_black.c b/src/red_black.c
7999 index 83a4095..a1dd300 100644
8000 --- a/src/red_black.c
8001 +++ b/src/red_black.c
8002 @@ -7,13 +7,6 @@
8003 #include "red_black.h"
8004 #include <stdio.h>
8005
8006 /**
8007  * definicao do que eh considerado RED e BLACK
8008  * neste codigo para maior clareza
8009  */
8010 #define BLACK 0
8011 #define RED 1
8012
8013 void (*rb_impr_elem)(rb_n *);
8014
8015 rb_n* rb_cr(int key, void *dado)
8016 @@ -88,15 +81,15 @@ int rb_insert(rb_n *t, rb_n *z)
8017 z->l = z->r = NULL;
8018 z->c = RED;
8019 /** ajuste */
8020 while (z->p->c == RED)
8021 + while (z && z->p && z->p->c == RED)
8022 {
8023     if (z->p == z->p->p->l)
8024     {
8025         y = z->p->p->r;
8026         if (y->c == RED)
8027 +         if (y && y->c == RED)
8028         {
8029             z->p->c = BLACK;
8030             y->p->p->c = RED;
8031 +             y->c = RED;
8032             z = z->p->p;
8033         }
8034     }
8035     else
8036     @@ -104,20 +97,20 @@ int rb_insert(rb_n *t, rb_n *z)
8037     if (z == z->p->r)
8038     {
8039         z = z->p;
8040         rb_rot_esq(t, z);
8041         rb_rot_esq(&t, z);
8042     }
8043     z->p->c = BLACK;
8044     z->p->p->c = RED;
8045     rb_rot_dir(t, z->p->p);
8046     rb_rot_dir(&t, z->p->p);
8047 }
8048 }
8049 else
8050 {
8051     y = z->p->p->l;
8052     if (y->c == RED)
8053 +     if (y && y->c == RED)
8054     {
8055         z->p->c = BLACK;
8056         y->p->p->c = RED;
8057 +         y->c = RED;
8058         z = z->p->p;
8059     }
8060     else
8061     @@ -125,11 +118,11 @@ int rb_insert(rb_n *t, rb_n *z)
8062     if (z == z->p->l)
8063     {
8064         z = z->p;
8065         rb_rot_dir(t, z);
8066         rb_rot_dir(&t, z);
8067     }
8068     z->p->c = BLACK;
8069     z->p->p->c = RED;
8070     rb_rot_esq(t, z->p->p);
8071     rb_rot_esq(&t, z->p->p);
8072 }
8073 }
8074 @@ -204,7 +197,7 @@ int rb_delete(rb_n *t, rb_n *z)
8075 {
8076     w->c = BLACK;
8077     x->p->c = RED;
8078     rb_rot_esq(t, x->p);
8079     rb_rot_esq(&t, x->p);
8080     w = x->p->r;
8081 }
8082
8083 @@ -219,14 +212,14 @@ int rb_delete(rb_n *t, rb_n *z)
8084 {
8085     w->l->c = BLACK;
8086     w->c = RED;
8087     rb_rot_dir(t, w);
8088     rb_rot_dir(&t, w);
8089     w = x->p->r;
8090 }
8091
8092 w->c = x->p->c;
8093 x->p->c = BLACK;
8094 w->r->c = BLACK;
8095 rb_rot_esq(t, x->p);
8096 + rb_rot_esq(&t, x->p);
8097 x = t;
8098 }
8099 }
8100 @@ -237,7 +230,7 @@ int rb_delete(rb_n *t, rb_n *z)
8101 {
8102     w->c = BLACK;
8103     x->p->c = RED;
8104     rb_rot_esq(t, x->p);
8105 + rb_rot_esq(&t, x->p);
8106     w = x->p->l;
8107 }
8108
8109 @@ -252,14 +245,14 @@ int rb_delete(rb_n *t, rb_n *z)
8110 {
8111     w->r->c = BLACK;

```

```

8112 |             w->c = RED;
8113 | -             rb_rot_dir(t, w);
8114 | +             rb_rot_dir(&t, w);
8115 |             w = x->p->l;
8116 |         }
8117 |
8118 |             w->c = x->p->c;
8119 |             x->p->c = BLACK;
8120 |             w->l->c = BLACK;
8121 | -             rb_rot_esq(t, x->p);
8122 | +             rb_rot_esq(&t, x->p);
8123 |             x = t;
8124 |         }
8125 |     }
8126 | @@ -294,7 +287,7 @@ int rb_destroi(rb_n **a)
8127 |     return 0;
8128 | }
8129 |
8130 | -int rb_rot_esq(rb_n *r, rb_n *w)
8131 | +int rb_rot_esq(rb_n **t, rb_n *w)
8132 | {
8133 |     rb_n *v = w->r;
8134 |     w->r = v->l;
8135 | @@ -302,7 +295,7 @@ int rb_rot_esq(rb_n *r, rb_n *w)
8136 |     v->l->p = w;
8137 |     v->p = w->p;
8138 |     if (!w->p)
8139 | -         r = v;
8140 | +         *t = v;
8141 |     else if (w == w->p->l)
8142 |         w->p->l = v;
8143 |     else
8144 | @@ -313,7 +306,7 @@ int rb_rot_esq(rb_n *r, rb_n *w)
8145 |     return 0;
8146 | }
8147 |
8148 | -int rb_rot_dir(rb_n *r, rb_n *w)
8149 | +int rb_rot_dir(rb_n **t, rb_n *w)
8150 | {
8151 |     rb_n *v = w->l;
8152 |     w->l = v->r;
8153 | @@ -321,7 +314,7 @@ int rb_rot_dir(rb_n *r, rb_n *w)
8154 |     v->r->p = w;
8155 |     v->p = w->p;
8156 |     if (!w->p)
8157 | -         r = v;
8158 | +         *t = v;
8159 |     else if (w == w->p->r)
8160 |         w->p->r = v;
8161 |     else
8162 | diff --git a/src/red_black.h b/src/red_black.h
8163 | index 1a97b13..f874138 100644
8164 | --- a/src/red_black.h
8165 | +++ b/src/red_black.h
8166 | @@ -6,6 +6,13 @@
8167 |  */
8168 | #include <stdlib.h>
8169 |
8170 | /**
8171 |  * definicao do que eh considerado RED e BLACK
8172 |  * neste codigo para maior clareza
8173 |  */
8174 | #define BLACK 0
8175 | #define RED 1
8176 |
8177 | /*
8178 |  * rubro negra
8179 |  * 0 - todo noh eh vermelho ou preto
8180 | @@ -116,13 +123,13 @@ int rb_troca_cor(rb_n *a);
8181 |  */
8182 |  * rotaciona subarvore a esquerda apartir do noh @param a
8183 |  */
8184 | -int rb_rot_esq(rb_n *r, rb_n *w);
8185 | +int rb_rot_esq(rb_n **t, rb_n *w);
8186 |
8187 | /**
8188 |  * rotaciona subarvore a direita apartir do noh
8189 |  * @param a noh
8190 |  */
8191 | -int rb_rot_dir(rb_n *r, rb_n *w);
8192 | +int rb_rot_dir(rb_n **t, rb_n *w);
8193 |
8194 | /**
8195 |  * imprime elemento como definido pelo o codigo cliente da biblioteca
8196 |
8197 | commit 314a4c50f8bda0bb4d97c44353ccc5423abf95fb
8198 | Author: Joilnen <joilnen@gmail.com>
8199 | Date: Mon Dec 20 07:17:50 2021 -0300
8200 |
8201 |     esmiucando a insercao e as rotacoes pra ver onde tah o erro
8202 |
8203 | diff --git a/src/red_black.c b/src/red_black.c
8204 | index a1dd300..f3a2e2a 100644
8205 | --- a/src/red_black.c
8206 | +++ b/src/red_black.c
8207 | @@ -62,16 +62,39 @@ int rb_ins_elem(rb_n *a, int k, void *e)
8208 |  */
8209 | int rb_insert(rb_n *t, rb_n *z)
8210 | {
8211 |     /**
8212 |     * y receberah noh pai imediato do noh adicionado
8213 |     * x eh um ponteiro temporario para iteracao feita
8214 |     * no loop while a seguir para chegar ao noh folha
8215 |     */
8216 |     rb_n *y = NULL;
8217 |     rb_n *x = t;
8218 |
8219 |     /**
8220 |     * itera para chegar no noh folha
8221 |     * da posicao a ser inserido o novo noh
8222 |     */
8223 |     while (x)
8224 |     {
8225 |         y = x;
8226 |         if (z->k < x->k)
8227 |             x = x->l;

```



```

8228 | -         else x = x->r;
8229 | +         else
8230 | +             x = x->r;
8231 | +     }
8232 | +
8233 | +     /**
8234 | +      * seta noh pai para o noh adicionado
8235 | +      */
8236 | +     z->p = y;
8237 | +
8238 | +     /**
8239 | +      * se esse noh adicionado eh unico y permance
8240 | +      * nulo que neste caso apontaria para o pai que eh nulo
8241 | +      * neste cenario torna o noh adicionado raiz
8242 | +      * caso n adiciona na esquerda ou direita de acordo
8243 | +      * com o valor da chave, noh recebe RED como padrao
8244 | +      */
8245 | +     if (!y)
8246 | +         t = z;
8247 | +     else if (z->k < y->k)
8248 | +         @@ -80,6 +103,7 @@ int rb_insert(rb_n *t, rb_n *z)
8249 | +             y->r = z;
8250 | +             z->l = z->r = NULL;
8251 | +             z->c = RED;
8252 | +
8253 | +     /** ajuste */
8254 | +     while (z && z->p && z->p->c == RED)
8255 | +     {
8256 | +         @@ -130,7 +154,7 @@ int rb_insert(rb_n *t, rb_n *z)
8257 | +     }
8258 | +
8259 | +     /**
8260 | +      * fucnao que faz trocas auxiliares na delecao
8261 | +      * funcao que faz trocas auxiliares na delecao
8262 | +      */
8263 | +     static int rb_transplant(rb_n *t, rb_n *u, rb_n *v)
8264 | +     {
8265 | +         @@ -147,8 +171,8 @@ static int rb_transplant(rb_n *t, rb_n *u, rb_n *v)
8266 | +         static rb_n *tree_minimum(rb_n *x)
8267 | +         {
8268 | +             rb_n *a = x;
8269 | +             while (a->r)
8270 | +                 a = a->r;
8271 | +             while (a->l)
8272 | +                 a = a->l;
8273 | +             return a;
8274 | +         }
8275 | +
8276 | +         @@ -157,6 +181,7 @@ int rb_delete(rb_n *t, rb_n *z)
8277 | +             rb_n *x;
8278 | +             rb_n *y = z;
8279 | +             unsigned char yoc = y->c;
8280 | +
8281 | +             if (!z->l)
8282 | +             {
8283 | +                 x = z->r;
8284 | +                 @@ -287,40 +312,40 @@ int rb_destroi(rb_n **a)
8285 | +                 return 0;
8286 | +             }
8287 | +
8288 | +             -int rb_rot_esq(rb_n **t, rb_n *w)
8289 | +             +int rb_rot_esq(rb_n **t, rb_n *x)
8290 | +             {
8291 | +                 rb_n *v = w->r;
8292 | +                 w->r = v->l;
8293 | +                 if (v->l)
8294 | +                     v->l->p = w;
8295 | +                 v->p = w->p;
8296 | +                 if (!w->p)
8297 | +                     *t = v;
8298 | +                 else if (w == w->p->l)
8299 | +                     w->p->l = v;
8300 | +                 rb_n *y = x->r;
8301 | +                 x->r = y->l;
8302 | +                 if (y->l)
8303 | +                     y->l->p = x;
8304 | +                 y->p = x->p;
8305 | +                 if (!x->p)
8306 | +                     *t = y;
8307 | +                 else if (x == x->p->l)
8308 | +                     x->p->l = y;
8309 | +                 else
8310 | +                     w->p->r = v;
8311 | +                 v->l = w;
8312 | +                 w->p = v;
8313 | +                 x->p->r = y;
8314 | +                 y->l = x;
8315 | +                 x->p = y;
8316 | +
8317 | +                 return 0;
8318 | +             }
8319 | +
8320 | +             -int rb_rot_dir(rb_n **t, rb_n *w)
8321 | +             +int rb_rot_dir(rb_n **t, rb_n *x)
8322 | +             {
8323 | +                 rb_n *v = w->l;
8324 | +                 w->l = v->r;
8325 | +                 if (v->r)
8326 | +                     v->r->p = w;
8327 | +                 v->p = w->p;
8328 | +                 if (!w->p)
8329 | +                     *t = v;
8330 | +                 else if (w == w->p->r)
8331 | +                     w->p->r = v;
8332 | +                 rb_n *y = x->l;
8333 | +                 x->l = y->r;
8334 | +                 if (y->r)
8335 | +                     y->r->p = x;
8336 | +                 y->p = x->p;
8337 | +                 if (!x->p)
8338 | +                     *t = y;
8339 | +                 else if (x == x->p->r)
8340 | +                     x->p->r = y;
8341 | +                 else
8342 | +                     w->p->l = v;
8343 | +                 v->r = w;

```

```

8344 | - w->p = v;
8345 | + x->p->l = y;
8346 | + y->r = x;
8347 | + x->p = y;
8348 |
8349 |     return 0;
8350 | }
8351 | @@ -334,7 +359,10 @@ void rb_pr(rb_n *a)
8352 | {
8353 |     if (a)
8354 |     {
8355 |         printf("k:%dc:%d\n", a->k, a->c);
8356 |         if(a->c == BLACK)
8357 |             printf("\033[44m\033[37mk:%dc:%d\033[0m\n", a->k, a->c);
8358 |         else
8359 |             printf("\033[41m\033[37mk:%dc:%d\033[0m\n", a->k, a->c);
8360 |         rb_pr(a->l);
8361 |         rb_pr(a->r);
8362 |     }
8363 | diff --git a/src/red_black.h b/src/red_black.h
8364 | index f874138..404346b 100644
8365 | --- a/src/red_black.h
8366 | +++ b/src/red_black.h
8367 | @@ -123,13 +123,13 @@ int rb_troca_cor(rb_n *a);
8368 | /**
8369 |  * rotaciona subarvore a esquerda apartir do noh @param a
8370 |  */
8371 | -int rb_rot_esq(rb_n **t, rb_n *w);
8372 | +int rb_rot_esq(rb_n **t, rb_n *x);
8373 |
8374 | /**
8375 |  * rotaciona subarvore a direita apartir do noh
8376 |  * @param a noh
8377 |  */
8378 | -int rb_rot_dir(rb_n **t, rb_n *w);
8379 | +int rb_rot_dir(rb_n **t, rb_n *x);
8380 |
8381 | /**
8382 |  * imprime elemento como definido pelo o codigo cliente da biblioteca
8383 |  */
8384 | commit 672008b3e8ddc58842129fdc4572efadd7358295
8385 | Author: Joilnen <joilnen@gmail.com>
8386 | Date: Mon Dec 20 11:39:40 2021 -0300
8387 |
8388 |     consertando rotacoes
8389 |
8390 | diff --git a/src/main.c b/src/main.c
8391 | index 808a8aa..4edeal8 100644
8392 | --- a/src/main.c
8393 | +++ b/src/main.c
8394 | @@ -61,9 +61,7 @@ int main()
8395 |
8396 |     rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
8397 |     novo_noh->k = 9;
8398 |     novo_noh->c = RED;
8399 |     novo_noh->d = &mt1;
8400 |     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
8401 |
8402 |     rb_insert(conjunto_a, novo_noh);
8403 |
8404 | @@ -76,9 +74,7 @@ int main()
8405 |
8406 |     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8407 |     novo_noh->k = 11;
8408 |     novo_noh->c = RED;
8409 |     novo_noh->d = &mt2;
8410 |     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
8411 |     rb_insert(conjunto_a, novo_noh);
8412 |
8413 |     meu_tipo mt3;
8414 | @@ -90,9 +86,7 @@ int main()
8415 |
8416 |     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8417 |     novo_noh->k = 8;
8418 |     novo_noh->c = RED;
8419 |     novo_noh->d = &mt3;
8420 |     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
8421 |     rb_insert(conjunto_a, novo_noh);
8422 |
8423 |     meu_tipo mt4;
8424 | @@ -104,9 +98,7 @@ int main()
8425 |
8426 |     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8427 |     novo_noh->k = 6;
8428 |     novo_noh->c = RED;
8429 |     novo_noh->d = &mt4;
8430 |     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
8431 |     rb_insert(conjunto_a, novo_noh);
8432 |
8433 |     meu_tipo mt5;
8434 | @@ -118,9 +110,7 @@ int main()
8435 |
8436 |     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8437 |     novo_noh->k = 5;
8438 |     novo_noh->c = RED;
8439 |     novo_noh->d = &mt5;
8440 |     novo_noh->l = novo_noh->r = novo_noh->p = NULL;
8441 |     rb_insert(conjunto_a, novo_noh);
8442 |
8443 |     rb_pr(conjunto_a);
8444 | diff --git a/src/red_black.c b/src/red_black.c
8445 | index f3a2e2a..8f155e0 100644
8446 | --- a/src/red_black.c
8447 | +++ b/src/red_black.c
8448 | @@ -113,7 +113,8 @@ int rb_insert(rb_n *t, rb_n *z)
8449 |     if (y && y->c == RED)
8450 |     {
8451 |         z->p->c = BLACK;
8452 |         y->c = RED;
8453 |         y->c = BLACK;
8454 |         z->p->p->c = RED;
8455 |         z = z->p->p;
8456 |     }
8457 |     else
8458 | @@ -134,7 +135,8 @@ int rb_insert(rb_n *t, rb_n *z)
8459 |         if (y && y->c == RED)

```

```

8460         {
8461             z->p->c = BLACK;
8462         -       y->c = RED;
8463         +       y->c = BLACK;
8464         +       z->p->p->c = RED;
8465             z = z->p->p;
8466         }
8467     else
8468
8469 commit e201eca5b7f67035f56a61791abccccee40f5c3cc
8470 Author: Joilinen <joilinen@gmail.com>
8471 Date: Mon Dec 20 15:25:24 2021 -0300
8472
8473     foi consertado um ponteiro para funcao de insercao pra ser por referencia
8474     pra possibilitar sua alteracao dentro da funcao e adicionado
8475     exemplos que forcassem a rotacao da raiz, o corre com 7 nohs
8476     adicionados
8477
8478 diff --git a/src/main.c b/src/main.c
8479 index 4edeal8..8d92b95 100644
8480 --- a/src/main.c
8481 +++ b/src/main.c
8482 @@ -57,61 +57,78 @@ int main()
8483     strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
8484     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
8485     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
8486 -    // rb_ins_elem(conjunto_a, 9, &mt1);
8487
8488     rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
8489     novo_noh->k = 9;
8490     novo_noh->d = &mt1;
8491
8492 -    rb_insert(conjunto_a, novo_noh);
8493 +    rb_insert(&conjunto_a, novo_noh);
8494
8495     meu_tipo mt2;
8496     mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
8497     strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
8498     mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
8499     strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
8500 -    // rb_ins_elem(conjunto_a, 11, &mt2);
8501
8502     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8503     novo_noh->k = 11;
8504     novo_noh->d = &mt2;
8505 +    rb_insert(conjunto_a, novo_noh);
8506 +    rb_insert(&conjunto_a, novo_noh);
8507
8508     meu_tipo mt3;
8509     mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
8510     strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
8511     mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
8512     strncpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
8513 -    // rb_ins_elem(conjunto_a, 11, &mt2);
8514
8515     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8516     novo_noh->k = 8;
8517     novo_noh->d = &mt3;
8518 -    rb_insert(conjunto_a, novo_noh);
8519 +    rb_insert(&conjunto_a, novo_noh);
8520
8521     meu_tipo mt4;
8522     mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
8523     strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
8524     mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
8525     strncpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
8526 -    // rb_ins_elem(conjunto_a, 11, &mt2);
8527
8528     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8529     novo_noh->k = 6;
8530     novo_noh->d = &mt4;
8531 -    rb_insert(conjunto_a, novo_noh);
8532 +    rb_insert(&conjunto_a, novo_noh);
8533
8534     meu_tipo mt5;
8535     mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
8536     strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
8537     mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
8538     strncpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
8539 -    // rb_ins_elem(conjunto_a, 11, &mt2);
8540
8541     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8542     novo_noh->k = 5;
8543     novo_noh->d = &mt5;
8544 -    rb_insert(conjunto_a, novo_noh);
8545 +    rb_insert(&conjunto_a, novo_noh);
8546
8547     meu_tipo mt6;
8548     mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
8549     strncpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
8550     mt6.telefone = (char *) malloc(sizeof("999999999") + 1);
8551     strncpy(mt6.telefone, "999999999", sizeof("999999999") + 1);
8552 +
8553     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8554     novo_noh->k = 4;
8555     novo_noh->d = &mt6;
8556     rb_insert(&conjunto_a, novo_noh);
8557
8558     meu_tipo mt7;
8559     mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
8560     strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
8561     mt7.telefone = (char *) malloc(sizeof("999999999") + 1);
8562     strncpy(mt7.telefone, "999999999", sizeof("999999999") + 1);
8563 +
8564     novo_noh = (rb_n *) malloc (sizeof(rb_n));
8565     novo_noh->k = 3;
8566     novo_noh->d = &mt7;
8567     rb_insert(&conjunto_a, novo_noh);
8568
8569     rb_pr(conjunto_a);
8570     rb_delete(conjunto_a, novo_noh);
8571 diff --git a/src/red_black.c b/src/red_black.c
8572 index 8f155e0..0cdcd3a 100644
8573 --- a/src/red_black.c
8574 +++ b/src/red_black.c
8575 @@ -60,7 +60,7 @@ int rb_ins_elem(rb_n *a, int k, void *e)

```

```

8576 /**
8577  * inclusao baseado no livro do cormen
8578  */
8579 -int rb_insert(rb_n *t, rb_n *z)
8580 +int rb_insert(rb_n **t, rb_n *z)
8581 {
8582     /**
8583      * y receberah noh pai imediato do noh adicionado
8584      @@ -68,7 +68,7 @@ int rb_insert(rb_n *t, rb_n *z)
8585      * no loop while a seguir para chegar ao noh folha
8586      */
8587     rb_n *y = NULL;
8588     rb_n *x = t;
8589     + rb_n *x = *t;
8590
8591     /**
8592      * itera para chegar no noh folha
8593      @@ -122,11 +122,11 @@ int rb_insert(rb_n *t, rb_n *z)
8594      if (z == z->p->r)
8595      {
8596          z = z->p;
8597          rb_rot_esq(&t, z);
8598          + rb_rot_esq(t, z);
8599      }
8600      z->p->c = BLACK;
8601      z->p->p->c = RED;
8602      - rb_rot_dir(&t, z->p->p);
8603      + rb_rot_dir(t, z->p->p);
8604      }
8605     }
8606     else
8607     @@ -144,15 +144,15 @@ int rb_insert(rb_n *t, rb_n *z)
8608     if (z == z->p->l)
8609     {
8610         z = z->p;
8611         - rb_rot_dir(&t, z);
8612         + rb_rot_dir(t, z);
8613     }
8614     z->p->c = BLACK;
8615     z->p->p->c = RED;
8616     - rb_rot_esq(&t, z->p->p);
8617     + rb_rot_esq(t, z->p->p);
8618     }
8619     }
8620     - t->c = BLACK;
8621     + (*t)->c = BLACK;
8622 }
8623
8624 /**
8625 diff --git a/src/red_black.h b/src/red_black.h
8626 index 404346b..efe1615 100644
8627 --- a/src/red_black.h
8628 +++ b/src/red_black.h
8629 @@ -63,7 +63,7 @@ int rb_ins_elem(rb_n *a, int k, void *e);
8630  * @param a noh raiz da arvore
8631  * @param b elemento a ser inserido
8632  */
8633 -int rb_insert(rb_n *t, rb_n *z);
8634 +int rb_insert(rb_n **t, rb_n *z);
8635
8636 /**
8637  * @param a noh raiz da arvore
8638
8639 commit 0ad91c93a5ba9c26e870931e600ae5be11d76c26
8640 Author: Joilnen <joilnen@gmail.com>
8641 Date: Mon Dec 20 15:46:33 2021 -0300
8642
8643 delecao com problema caso em q o noh deletado eh o raiz, isso quebra
8644
8645 diff --git a/src/main.c b/src/main.c
8646 index 8d92b95..07a1566 100644
8647 --- a/src/main.c
8648 +++ b/src/main.c
8649 @@ -131,7 +131,7 @@ int main()
8650     rb_insert(&conjunto_a, novo_noh);
8651     rb_pr(conjunto_a);
8652     - rb_delete(conjunto_a, novo_noh);
8653     + rb_delete(&conjunto_a, novo_noh);
8654     printf("\n");
8655     rb_pr(conjunto_a);
8656
8657 diff --git a/src/red_black.c b/src/red_black.c
8658 index 0cdcd3a..01349fe 100644
8659 --- a/src/red_black.c
8660 +++ b/src/red_black.c
8661 @@ -96,7 +96,7 @@ int rb_insert(rb_n **t, rb_n *z)
8662     * com o valor da chave, noh recebe RED como padrao
8663     */
8664     if (!y)
8665     - t = z;
8666     + *t = z;
8667     else if (z->k < y->k)
8668     y->l = z;
8669     else
8670     @@ -158,10 +158,10 @@ int rb_insert(rb_n **t, rb_n *z)
8671     /**
8672      * funcao que faz trocas auxiliares na delecao
8673      */
8674     -static int rb_transplant(rb_n *t, rb_n *u, rb_n *v)
8675     +static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
8676     {
8677         if (!u->p)
8678         - t = v;
8679         + *t = v;
8680         else if (u == u->p->l)
8681         u->p->l = v;
8682         else
8683     @@ -178,7 +178,7 @@ static rb_n *tree_minimum(rb_n *x)
8684     return a;
8685     }
8686
8687 -int rb_delete(rb_n *t, rb_n *z)
8688 +int rb_delete(rb_n **t, rb_n *z)
8689 {

```

```

8692     rb_n *x;
8693     rb_n *y = z;
8694 @@ -215,7 +215,7 @@ int rb_delete(rb_n *t, rb_n *z)
8695     if (yoc == BLACK)
8696     {
8697         rb_n *w;
8698         while (x != t && x->c == BLACK)
8699 +         while (x != *t && x->c == BLACK)
8700         {
8701             if (x == x->p->l)
8702             {
8703 @@ -224,7 +224,7 @@ int rb_delete(rb_n *t, rb_n *z)
8704             {
8705                 w->c = BLACK;
8706                 x->p->c = RED;
8707 -                 rb_rot_esq(&t, x->p);
8708 +                 rb_rot_esq(t, x->p);
8709                 w = x->p->r;
8710             }
8711
8712 @@ -239,15 +239,15 @@ int rb_delete(rb_n *t, rb_n *z)
8713             {
8714                 w->l->c = BLACK;
8715                 w->c = RED;
8716 -                 rb_rot_dir(&t, w);
8717 +                 rb_rot_dir(t, w);
8718                 w = x->p->r;
8719             }
8720
8721                 w->c = x->p->c;
8722                 x->p->c = BLACK;
8723                 w->r->c = BLACK;
8724 -                 rb_rot_esq(&t, x->p);
8725 -                 x = t;
8726 +                 rb_rot_esq(t, x->p);
8727 +                 x = *t;
8728             }
8729         }
8730     }
8731 @@ -257,7 +257,7 @@ int rb_delete(rb_n *t, rb_n *z)
8732     {
8733         w->c = BLACK;
8734         x->p->c = RED;
8735 -         rb_rot_esq(&t, x->p);
8736 +         rb_rot_esq(t, x->p);
8737         w = x->p->l;
8738     }
8739
8740 @@ -272,15 +272,15 @@ int rb_delete(rb_n *t, rb_n *z)
8741     {
8742         w->r->c = BLACK;
8743         w->c = RED;
8744 -         rb_rot_dir(&t, w);
8745 +         rb_rot_dir(t, w);
8746         w = x->p->l;
8747     }
8748
8749         w->c = x->p->c;
8750         x->p->c = BLACK;
8751         w->l->c = BLACK;
8752 -         rb_rot_esq(&t, x->p);
8753 -         x = t;
8754 +         rb_rot_esq(t, x->p);
8755 +         x = *t;
8756     }
8757 }
8758 }
8759 diff --git a/src/red_black.h b/src/red_black.h
8760 index efe1615..80f39db 100644
8761 --- a/src/red_black.h
8762 +++ b/src/red_black.h
8763 @@ -69,7 +69,7 @@ int rb_insert(rb_n **t, rb_n *z);
8764 * @param a noh raiz da arvore
8765 * @param b elemento a ser deletado
8766 */
8767 -int rb_delete(rb_n *t, rb_n *z);
8768 +int rb_delete(rb_n **t, rb_n *z);
8769
8770 /**
8771 * @param a noh raiz da arvore
8772
8773 commit 587184fd58bd2a3919801b5ddfb9d8f6b5f72af0
8774 Author: Joilnen <joilnen@gmail.com>
8775 Date: Mon Dec 20 18:35:17 2021 -0300
8776
8777     delete cheio de problemas por enquanto soh insersao funciona bem
8778
8779 diff --git a/src/main.c b/src/main.c
8780 index 07a1566..c278464 100644
8781 --- a/src/main.c
8782 +++ b/src/main.c
8783 @@ -131,7 +131,11 @@ int main()
8784     rb_insert(&conjunto_a, novo_noh);
8785
8786     rb_pr(conjunto_a);
8787 -     rb_delete(&conjunto_a, novo_noh);
8788 +     // rb_delete(&conjunto_a, tt);
8789 +     rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
8790 +     printf("\n");
8791 +     rb_pr(conjunto_a);
8792 +     rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
8793 +     printf("\n");
8794     rb_pr(conjunto_a);
8795
8796 diff --git a/src/red_black.c b/src/red_black.c
8797 index 01349fe..9a1395a 100644
8798 --- a/src/red_black.c
8799 +++ b/src/red_black.c
8800 @@ -371,3 +371,21 @@ void rb_pr(rb_n *a)
8801 }
8802
8803
8804 +rb_n *rb_search(rb_n *t, int k)
8805 +{
8806 +     rb_n *y = NULL;
8807 +     rb_n *x = t;

```

```

8808 +
8809 +     while (x)
8810 +     {
8811 +         y = x;
8812 +         if (k < x->k)
8813 +             x = x->l;
8814 +         else if (k > x->k)
8815 +             x = x->r;
8816 +         else
8817 +             return x;
8818 +     }
8819 +
8820 +     return NULL;
8821 +}
8822 diff --git a/src/red_black.h b/src/red_black.h
8823 index 80f39db..2f93388 100644
8824 --- a/src/red_black.h
8825 +++ b/src/red_black.h
8826 @@ -60,12 +60,21 @@ unsigned int rb_nelem(rb_n *a);
8827 int rb_ins_elem(rb_n *a, int k, void *e);
8828
8829 /**
8830 * tc
8831 * @param a noh raiz da arvore
8832 * @param b elemento a ser inserido
8833 */
8834 int rb_insert(rb_n **t, rb_n *z);
8835
8836 /**
8837 * tch
8838 * @param a noh raiz da arvore
8839 * @param b elemento a ser inserido
8840 */
8841 +rb_n *rb_search(rb_n *t, int k);
8842 +
8843 /**
8844 * tc
8845 * @param a noh raiz da arvore
8846 * @param b elemento a ser deletado
8847 */
8848
8849 commit 9d8058765125f656a322aa28591432e204025a57
8850 Author: Joilnen <joilnen@gmail.com>
8851 Date: Mon Dec 20 21:03:54 2021 -0300
8852
8853     consertado o delete era chegar validade de um ponteiro
8854
8855 diff --git a/src/main.c b/src/main.c
8856 index c278464..17a7261 100644
8857 --- a/src/main.c
8858 +++ b/src/main.c
8859 @@ -138,6 +138,12 @@ int main()
8860     rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
8861     printf("\n");
8862     rb_pr(conjunto_a);
8863 +    rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
8864 +    printf("\n");
8865 +    rb_pr(conjunto_a);
8866 +    rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
8867 +    printf("\n");
8868 +    rb_pr(conjunto_a);
8869
8870     /**
8871      * testando apenas rb no momento
8872 diff --git a/src/red_black.c b/src/red_black.c
8873 index 9a1395a..62f968b 100644
8874 --- a/src/red_black.c
8875 +++ b/src/red_black.c
8876 @@ -215,7 +215,7 @@ int rb_delete(rb_n **t, rb_n *z)
8877     if (yoc == BLACK)
8878     {
8879         rb_n *w;
8880         while (x != *t && x->c == BLACK)
8881 +         while (x && x != *t && x->c == BLACK)
8882         {
8883             if (x == x->p->l)
8884             {
8885
8886 commit 79ff5bb796251b0ee9ce70d5cf0e55cbb23b7d04
8887 Author: Joilnen <joilnen@gmail.com>
8888 Date: Mon Dec 20 21:12:56 2021 -0300
8889
8890     checagem se passa NULL como noh para ser deletado, tava quebrando
8891
8892 diff --git a/src/main.c b/src/main.c
8893 index 17a7261..7217926 100644
8894 --- a/src/main.c
8895 +++ b/src/main.c
8896 @@ -144,7 +144,7 @@ int main()
8897     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
8898     printf("\n");
8899     rb_pr(conjunto_a);
8900 -
8901 +    rb_delete(&conjunto_a, NULL);
8902     /**
8903      * testando apenas rb no momento
8904     conjord *conj_a = conjord_cria();
8905 diff --git a/src/red_black.c b/src/red_black.c
8906 index 62f968b..e8325b2 100644
8907 --- a/src/red_black.c
8908 +++ b/src/red_black.c
8909 @@ -182,7 +182,11 @@ int rb_delete(rb_n **t, rb_n *z)
8910     {
8911         rb_n *x;
8912         rb_n *y = z;
8913         unsigned char yoc = y->c;
8914 +         unsigned char yoc;
8915 +
8916 +         if (!z)
8917 +             return -1;
8918 +         yoc = y->c;
8919 +
8920         if (!z->l)
8921         {
8922
8923 commit a2dedb7770c26d9d9ad06336a7ae9f03ac169af3

```

```

8924 Author: Joilnen <joilnen@gmail.com>
8925 Date: Mon Dec 20 21:44:44 2021 -0300
8926
8927     consertando mais um ponteiro invalido dentro do delete
8928
8929 diff --git a/src/main.c b/src/main.c
8930 index 7217926..876383a 100644
8931 --- a/src/main.c
8932 +++ b/src/main.c
8933 @@ -144,7 +144,16 @@ int main()
8934     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
8935     printf("\n");
8936     rb_pr(conjunto_a);
8937     rb_delete(&conjunto_a, NULL);
8938     rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
8939     printf("\n");
8940     rb_pr(conjunto_a);
8941     rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
8942     printf("\n");
8943     rb_pr(conjunto_a);
8944     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
8945     printf("\n nada");
8946     rb_pr(conjunto_a);
8947     printf("\n");
8948     /**
8949      * testando apenas rb no momento
8950      conjord *conj_a = conjord_cria();
8951 diff --git a/src/red_black.c b/src/red_black.c
8952 index e8325b2..f32dbe7 100644
8953 --- a/src/red_black.c
8954 +++ b/src/red_black.c
8955 @@ -204,7 +204,7 @@ int rb_delete(rb_n **t, rb_n *z)
8956     yoc = y->c;
8957     x = y->r;
8958     if (y->p == z)
8959         x->p = y;
8960     +     if(x) x->p = y;
8961     else
8962     {
8963         rb_transplant(t, y, y->r);
8964
8965 commit 80b0701d8f57960614ebb8df46ac71282e93669c
8966 Author: Joilnen <joilnen@gmail.com>
8967 Date: Mon Dec 20 21:50:58 2021 -0300
8968
8969     aparentemente o delete da arvore vermelha e preta esta deletando todos
8970     os casos inclusive todos os nohs sem crashes
8971
8972 diff --git a/src/main.c b/src/main.c
8973 index 876383a..efbadf8 100644
8974 --- a/src/main.c
8975 +++ b/src/main.c
8976 @@ -131,7 +131,6 @@ int main()
8977     rb_insert(&conjunto_a, novo_noh);
8978
8979     rb_pr(conjunto_a);
8980     // rb_delete(&conjunto_a, tt);
8981     rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
8982     printf("\n");
8983     rb_pr(conjunto_a);
8984 @@ -161,34 +160,6 @@ int main()
8985     */
8986 }
8987
8988 -int imprime_arvore(rb_n *a, short h)
8989 -{
8990     rb_n *it = a;
8991     int count = h;
8992     if(it)
8993     {
8994         for(; count; --count) printf(" ");
8995         printf("%d%d", it->k, it->c);
8996         if(it->l)
8997         {
8998             for(count = h - 2; h; --h) printf(" ");
8999             printf("%d%d", it->l->k, it->l->c);
9000
9001         }
9002         if(it->r)
9003         {
9004             for(count = h + 2; h; --h) printf(" ");
9005             printf("%d%d", it->r->k, it->r->c);
9006
9007         }
9008         printf("\n");
9009         imprime_arvore(it->r->r, h - 1);
9010         imprime_arvore(it->r->l, h - 1);
9011         imprime_arvore(it->l->r, h - 1);
9012         imprime_arvore(it->l->l, h - 1);
9013     }
9014 -}
9015
9016 /**
9017  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
9018  * exemplo
9019
9020 commit f9036f3c01bfdab050acf00fb6944b4d23917b6f
9021 Author: Joilnen <joilnen@gmail.com>
9022 Date: Tue Dec 21 15:00:32 2021 -0300
9023
9024     troquei minimum por sucessor aparentemente nao faz diferenca
9025     se a pesquisa for sempre feita pela a raiz
9026
9027 diff --git a/src/red_black.c b/src/red_black.c
9028 index f32dbe7..508282e 100644
9029 --- a/src/red_black.c
9030 +++ b/src/red_black.c
9031 @@ -178,6 +178,20 @@ static rb_n *tree_minimum(rb_n *x)
9032     return a;
9033 }
9034
9035 +static rb_n* sucessor(rb_n *x)
9036 +{
9037     +     rb_n *y = NULL;
9038     +     if (x->r)
9039     +         return tree_minimum(x->r);
9039

```

```

9040 +     y = x->p;
9041 +     while (y && x == y->r)
9042 +     {
9043 +         x = y;
9044 +         y = y->p;
9045 +     }
9046 +     return y;
9047 +}
9048 +
9049 +int rb_delete(rb_n **t, rb_n *z)
9050 +{
9051 +    rb_n **x;
9052 +    @@ -200,7 +214,8 @@ int rb_delete(rb_n **t, rb_n *z)
9053 +    }
9054 +    else
9055 +    {
9056 +        y = tree_minimum(z->r);
9057 +        // y = tree_minimum(z->r);
9058 +        y = successor(z);
9059 +        yoc = y->c;
9060 +        x = y->r;
9061 +        if (y->p == z)
9062 +        @@ -216,6 +231,7 @@ int rb_delete(rb_n **t, rb_n *z)
9063 +        y->l->p = y;
9064 +        y->c = z->c;
9065 +    }
9066 +
9067 +    if (yoc == BLACK)
9068 +    {
9069 +        rb_n *w;
9070 +
9071 +commit 4d24ed67efd370457da02dd43d353d6e8959f8d8
9072 Author: Joilnen <joilnen@gmail.com>
9073 Date: Tue Dec 21 19:02:03 2021 -0300
9074
9075     consertando a indentcao da funcao sucessor
9076
9077 diff --git a/src/red_black.c b/src/red_black.c
9078 index 508282e..b2e3e26 100644
9079 --- a/src/red_black.c
9080 +++ b/src/red_black.c
9081 @@ -182,12 +182,12 @@ static rb_n* sucessor(rb_n *x)
9082 {
9083     rb_n *y = NULL;
9084     if (x->r)
9085         return tree_minimum(x->r);
9086 +    return tree_minimum(x->r);
9087     y = x->p;
9088     while (y && x == y->r)
9089     {
9090         x = y;
9091         y = y->p;
9092 +        x = y;
9093 +        y = y->p;
9094     }
9095     return y;
9096 }
9097
9098 commit a21a35ac39cfed6c3d1b65e7ff57c7f421e9c095
9099 Author: Joilnen <joilnen@gmail.com>
9100 Date: Tue Dec 21 19:35:49 2021 -0300
9101
9102     melhorando comentario
9103
9104 diff --git a/src/red_black.c b/src/red_black.c
9105 index b2e3e26..3a41fbf 100644
9106 --- a/src/red_black.c
9107 +++ b/src/red_black.c
9108 @@ -157,6 +157,8 @@ int rb_insert(rb_n **t, rb_n *z)
9109
9110 /**
9111  * funcao que faz trocas auxiliares na delecao
9112  * @param u noh que eh trocado por outro em seu lugar
9113  * @param v noh que eh colocado no lugar do noh anterior
9114  */
9115 static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
9116 {
9117
9118 commit a195b68a2e66f953315fa07042b09e1396294ed6
9119 Author: Joilnen <joilnen@gmail.com>
9120 Date: Wed Dec 22 07:00:24 2021 -0300
9121
9122     atualizando interfaces
9123
9124 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
9125 index 8e3abe4..81d2e70 100644
9126 --- a/src/conjunto_ordenado.c
9127 +++ b/src/conjunto_ordenado.c
9128 @@ -18,3 +18,5 @@ int conjord_esvazia(conjord *a)
9129 {
9130
9131 }
9132 +
9133 +
9134
9135 commit c891837cbe985fdb319ed2dcbbf203c05c4d9557
9136 Author: Joilnen <joilnen@gmail.com>
9137 Date: Wed Dec 22 12:41:31 2021 -0300
9138
9139     escrevendo TAD conjunto ordenado conjord
9140
9141 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
9142 index 81d2e70..14f03ab 100644
9143 --- a/src/conjunto_ordenado.c
9144 +++ b/src/conjunto_ordenado.c
9145 @@ -1,5 +1,8 @@
9146 #include "conjunto_ordenado.h"
9147
9148 /**
9149  * cria conjunto ordenado e retorna o ponteiro para ele
9150  */
9151 conjord *conjord_cria()
9152 {
9153     conjord *c = (conjord *) malloc (sizeof(conjord));
9154     @@ -9,7 +12,19 @@ conjord *conjord_cria()
9155     return c;

```



```

9156 }
9157
9158 -int conjord_eh_vazio(conjord *a)
9159 /**
9160  * imprime elementos do conjunto subitem 1 do item 2
9161  * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
9162  */
9163 void conjord_imprime()
9164 {
9165     +
9166 }
9167 +
9168 /**
9169  * testa se o conjunto estah vazio
9170  */
9171 int conjord_eh_vazio(const conjord *a)
9172 {
9173     return !a->raiz;
9174 }
9175 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
9176 index d8ac556..aa94a83 100644
9177 --- a/src/conjunto_ordenado.h
9178 +++ b/src/conjunto_ordenado.h
9179 @@ -7,27 +7,91 @@
9180  * estrutura da TAD explicitamente
9181  * esta estrutura oculta o uso da red black
9182  * suas funcoes e funcionamento interno do cliente
9183  * expondo apenas funcoes e operacoes com semantica
9184  * relativa ao uso do cliente aqui nesta avaliacao
9185  * conjunto ordenado as operacoes e funcoes
9186  * exigidas no documento da avaliacao para este TAD
9187  */
9188 typedef struct conjord {
9189     rb_n *raiz;
9190     int cardinalidade;
9191 } conjord;
9192
9193 /**
9194  * cria conjunto ordenado e retorna o ponteiro para ele
9195  */
9196 conjord *conjord_cria();
9197 +
9198 /**
9199  * imprime elementos do conjunto subitem 1 do item 2
9200  * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
9201  */
9202 void conjord_imprime();
9203 +
9204 /**
9205  * retorna a cardinalidade do conjunto
9206  */
9207 unsigned int conjord_cardinalidade();
9208 -int conjord_eh_vazio();
9209 +
9210 /**
9211  * testa se o conjunto estah vazio
9212  */
9213 int conjord_eh_vazio(const conjord *a);
9214 +
9215 /**
9216  * torna o conjunto vazio
9217  */
9218 int conjord_esvazia();
9219 +
9220 /**
9221  * destroi e desaloca uma instancia de conjunto ordenado
9222  */
9223 void conjord_destroi(conjord **a);
9224 +
9225 /**
9226  * insere elemento no conjunto ordenado
9227  */
9228 void conjord_insere_elemento(conjord *a, int chave, void *elemento);
9229 +
9230 /**
9231  * exclui elemento do conjunto ordenado através da chave
9232  * é o identificador de acordo com especificacao
9233  */
9234 void conjord_exclui_elemento(conjord *a, int chave);
9235 +
9236 /**
9237  * testa se existe elemento com a chave especificado
9238  * no conjunto ordenado
9239  */
9240 int conjord_existe_elemento(conjord *a, int chave);
9241 +
9242 /**
9243  * intersecciona dois conjuntos formando um terceiro
9244  * como resultado desta operacao retornando seu ponteiro
9245  * a seguir
9246  */
9247 conjord *conjord_intersecciona(conjord *a, conjord *b);
9248 +
9249 /**
9250  * une dois conjuntos formando um terceiro
9251  * como resultado desta operacao retornando seu ponteiro
9252  * a seguir
9253  */
9254 conjord *conjord_une(conjord *a, conjord *b);
9255 +
9256 /**
9257  * subtrai dois conjuntos formando um terceiro
9258  * como resultado desta operacao retornando seu ponteiro
9259  * a seguir
9260  */
9261 conjord *conjord_subtrai(conjord *a, conjord *b);
9262 +
9263 /**
9264  * item 3 da especificacao
9265  * "Imprime a arvore rubro negra
9266  * nesta operacao, cada noh da arvore serah
9267  * representado pela sua chave de identificacao seguida da sua cor"
9268  */
9269 void conjord_imprime_rb();
9270
9271 #endif

```

```

9272 diff --git a/src/red_black.c b/src/red_black.c
9273 index 3a41fbf..77e9e98 100644
9274 --- a/src/red_black.c
9275 +++ b/src/red_black.c
9276 @@ -180,20 +180,27 @@ static rb_n *tree_minimum(rb_n *x)
9277     return a;
9278 }
9279
9280 /**
9281  * * sucessor baseado nas notas e no livro do cormen, reescrito
9282  * * e consegui diminuir em duas linhas do algoritmo original
9283  */
9284 static rb_n* sucessor(rb_n *x)
9285 {
9286     rb_n *y = NULL;
9287     if (x->r)
9288         return tree_minimum(x->r);
9289     y = x->p;
9290     while (y && x == y->r)
9291     {
9292         while ((y = x->p) && x == y->r)
9293             x = y;
9294         y = y->p;
9295     }
9296     return y;
9297 }
9298
9299 /**
9300  * * deleta noh da arvore red black
9301  * * @param t ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao
9302  * * e manter a alteracao depois do retorno da funcao
9303  * * @param z ponteiro para o noh a ser deletado, para deletar por chave o TAD deve
9304  * * passar a saida da funcao pesquisar, num arranjo funcional na chamada
9305  */
9306 int rb_delete(rb_n **t, rb_n *z)
9307 {
9308     rb_n *x;
9309     @@ -411,3 +418,5 @@ rb_n *rb_search(rb_n *t, int k)
9310     return NULL;
9311 }
9312 +
9313 +
9314 +
9315
9316 commit ff3a1dc2514de4e3a236485f6e6260f61266e1f0
9317 Author: Joilnen <joilnen@gmail.com>
9318 Date: Wed Dec 22 14:05:06 2021 -0300
9319
9320     testando impressao da arvore
9321
9322 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
9323 index 14f03ab..e79ff10 100644
9324 --- a/src/conjunto_ordenado.c
9325 +++ b/src/conjunto_ordenado.c
9326 @@ -1,3 +1,4 @@
9327  #include <stdio.h>
9328  #include "conjunto_ordenado.h"
9329
9330 /**
9331  @@ -34,4 +35,33 @@ int conjord_esvazia(conjord *a)
9332  }
9333  */
9334
9335 /**
9336  * * imprime arvore
9337  */
9338 #define COUNT 10
9339 void conjord_imprime_rb(rb_n *t, int s) {
9340     // Base case
9341     if (!t)
9342         return;
9343     // Increase distance between levels
9344     s += COUNT;
9345     // Process l child
9346     conjord_imprime_rb(t->l, s);
9347     // Print current node after space
9348     // count
9349     printf("\n");
9350     for (int i = COUNT; i < s; i++)
9351         printf(" ");
9352     if (t->c == BLACK)
9353         printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
9354     else
9355         printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
9356     // Process right child first
9357     conjord_imprime_rb(t->r, s);
9358 }
9359 +
9360 +
9361 +
9362 +
9363 +
9364
9365 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
9366 index aa94a83..5061946 100644
9367 --- a/src/conjunto_ordenado.h
9368 +++ b/src/conjunto_ordenado.h
9369 @@ -92,7 +92,7 @@ conjord *conjord_subtrai(conjord *a, conjord *b);
9370  * nesta operacao, cada noh da arvore serah
9371  * representado pela sua chave de identificacao seguida da sua cor"
9372  */
9373 -void conjord_imprime_rb();
9374 +void conjord_imprime_rb(rb_n *t, int s);
9375
9376 #endif
9377
9378 diff --git a/src/main.c b/src/main.c
9379 index efbadf8..1daea2a 100644
9380 --- a/src/main.c
9381 +++ b/src/main.c
9382 @@ -52,6 +52,11 @@ int main()
9383     /*
9384     rb_n *conjunto_a = rb_cr(10, &mt0);
9385
9386     rb_pr(conjunto_a);
9387     printf("= arvore =====\n");

```

```

9388 + conjord_imprime_rb(conjunto_a, 1);
9389 + printf("= =====\n");
9390 +
9391 meu_tipo mt1;
9392 mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
9393 strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
9394 @@ -64,6 +69,11 @@ int main()
9395
9396 rb_insert(&conjunto_a, novo_noh);
9397
9398 + rb_pr(conjunto_a);
9399 + printf("= arvore =====\n");
9400 + conjord_imprime_rb(conjunto_a, 1);
9401 + printf("= =====\n");
9402 +
9403 meu_tipo mt2;
9404 mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
9405 strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
9406 @@ -75,6 +85,11 @@ int main()
9407 novo_noh->d = &mt2;
9408 rb_insert(&conjunto_a, novo_noh);
9409
9410 + rb_pr(conjunto_a);
9411 + printf("= arvore =====\n");
9412 + conjord_imprime_rb(conjunto_a, 1);
9413 + printf("= =====\n");
9414 +
9415 meu_tipo mt3;
9416 mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
9417 strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
9418 @@ -86,6 +101,11 @@ int main()
9419 novo_noh->d = &mt3;
9420 rb_insert(&conjunto_a, novo_noh);
9421
9422 + rb_pr(conjunto_a);
9423 + printf("= arvore =====\n");
9424 + conjord_imprime_rb(conjunto_a, 1);
9425 + printf("= =====\n");
9426 +
9427 meu_tipo mt4;
9428 mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
9429 strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
9430 @@ -97,6 +117,11 @@ int main()
9431 novo_noh->d = &mt4;
9432 rb_insert(&conjunto_a, novo_noh);
9433
9434 + rb_pr(conjunto_a);
9435 + printf("= arvore =====\n");
9436 + conjord_imprime_rb(conjunto_a, 1);
9437 + printf("= =====\n");
9438 +
9439 meu_tipo mt5;
9440 mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
9441 strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
9442 @@ -119,6 +144,11 @@ int main()
9443 novo_noh->d = &mt6;
9444 rb_insert(&conjunto_a, novo_noh);
9445
9446 + rb_pr(conjunto_a);
9447 + printf("= arvore =====\n");
9448 + conjord_imprime_rb(conjunto_a, 1);
9449 + printf("= =====\n");
9450 +
9451 meu_tipo mt7;
9452 mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
9453 strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
9454 @@ -130,6 +160,10 @@ int main()
9455 novo_noh->d = &mt7;
9456 rb_insert(&conjunto_a, novo_noh);
9457
9458 + printf("= arvore =====\n");
9459 + conjord_imprime_rb(conjunto_a, 1);
9460 + printf("= =====\n");
9461 +
9462 rb_pr(conjunto_a);
9463 rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
9464 printf("\n");
9465 @@ -138,6 +172,9 @@ int main()
9466 printf("\n");
9467 rb_pr(conjunto_a);
9468 rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
9469 + printf("= arvore =====\n");
9470 + conjord_imprime_rb(conjunto_a, 1);
9471 + printf("= =====\n");
9472 + printf("\n");
9473 rb_pr(conjunto_a);
9474 rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
9475 @@ -145,12 +182,21 @@ int main()
9476 rb_pr(conjunto_a);
9477 rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
9478 printf("\n");
9479 + printf("= arvore =====\n");
9480 + conjord_imprime_rb(conjunto_a, 1);
9481 + printf("= =====\n");
9482 rb_pr(conjunto_a);
9483 rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
9484 printf("\n");
9485 rb_pr(conjunto_a);
9486 + printf("= arvore =====\n");
9487 + conjord_imprime_rb(conjunto_a, 1);
9488 + printf("= =====\n");
9489 rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
9490 - printf("\n nada");
9491 + printf("\n nada\n");
9492 + printf("= arvore =====\n");
9493 + conjord_imprime_rb(conjunto_a, 1);
9494 + printf("= =====\n");
9495 rb_pr(conjunto_a);
9496 printf("\n");
9497 /**
9498
9499 commit 5ec58093d577a71de87d125f0c9485c35b4665af
9500 Author: Joilnen <joilnen@gmail.com>
9501 Date: Wed Dec 22 14:41:05 2021 -0300
9502
9503 pequena demonstracao da rb

```

```

9504 diff --git a/src/main.c b/src/main.c
9505 index 1daea2a..b5968aa 100644
9506 --- a/src/main.c
9507 +++ b/src/main.c
9508 @@ -56,6 +56,8 @@ int main()
9509     printf("= arvore =====\n");
9510     conjord_imprime_rb(conjunto_a, 1);
9511     printf("= =====\n");
9512 +     printf(" * tecla enter *");
9513 +     getc(stdin);
9514
9515     meu_tipo mt1;
9516     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
9517 @@ -73,6 +75,8 @@ int main()
9518     printf("= arvore =====\n");
9519     conjord_imprime_rb(conjunto_a, 1);
9520     printf("= =====\n");
9521 +     printf(" * tecla enter *");
9522 +     getc(stdin);
9523
9524     meu_tipo mt2;
9525     mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
9526 @@ -89,6 +93,8 @@ int main()
9527     printf("= arvore =====\n");
9528     conjord_imprime_rb(conjunto_a, 1);
9529     printf("= =====\n");
9530 +     printf(" * tecla enter *");
9531 +     getc(stdin);
9532
9533     meu_tipo mt3;
9534     mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
9535 @@ -105,6 +111,8 @@ int main()
9536     printf("= arvore =====\n");
9537     conjord_imprime_rb(conjunto_a, 1);
9538     printf("= =====\n");
9539 +     printf(" * tecla enter *");
9540 +     getc(stdin);
9541
9542     meu_tipo mt4;
9543     mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
9544 @@ -121,6 +129,8 @@ int main()
9545     printf("= arvore =====\n");
9546     conjord_imprime_rb(conjunto_a, 1);
9547     printf("= =====\n");
9548 +     printf(" * tecla enter *");
9549 +     getc(stdin);
9550
9551     meu_tipo mt5;
9552     mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
9553 @@ -148,6 +158,8 @@ int main()
9554     printf("= arvore =====\n");
9555     conjord_imprime_rb(conjunto_a, 1);
9556     printf("= =====\n");
9557 +     printf(" * tecla enter *");
9558 +     getc(stdin);
9559
9560     meu_tipo mt7;
9561     mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
9562 @@ -163,6 +175,8 @@ int main()
9563     printf("= arvore =====\n");
9564     conjord_imprime_rb(conjunto_a, 1);
9565     printf("= =====\n");
9566 +     printf(" * tecla enter *");
9567 +     getc(stdin);
9568
9569     rb_pr(conjunto_a);
9570     rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
9571 @@ -176,6 +190,8 @@ int main()
9572     conjord_imprime_rb(conjunto_a, 1);
9573     printf("= =====\n");
9574     printf("\n");
9575 +     printf(" * tecla enter *");
9576 +     getc(stdin);
9577     rb_pr(conjunto_a);
9578     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
9579     printf("\n");
9580 @@ -185,6 +201,8 @@ int main()
9581     printf("= arvore =====\n");
9582     conjord_imprime_rb(conjunto_a, 1);
9583     printf("= =====\n");
9584 +     printf(" * tecla enter *");
9585 +     getc(stdin);
9586     rb_pr(conjunto_a);
9587     rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
9588     printf("\n");
9589 @@ -192,6 +210,8 @@ int main()
9590     printf("= arvore =====\n");
9591     conjord_imprime_rb(conjunto_a, 1);
9592     printf("= =====\n");
9593 +     printf(" * tecla enter *");
9594 +     getc(stdin);
9595     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
9596     printf("\n nada\n");
9597     printf("= arvore =====\n");
9598
9599
9600 commit 5cd18bc478e1492c25c4468738ae5b40add977
9601 Author: Joilnen <joilnen@gmail.com>
9602 Date: Wed Dec 22 17:37:42 2021 -0300
9603
9604     demonstracao da arvore rb
9605
9606 diff --git a/src/main.c b/src/main.c
9607 index b5968aa..2da400d 100644
9608 --- a/src/main.c
9609 +++ b/src/main.c
9610 @@ -56,7 +56,7 @@ int main()
9611     printf("= arvore =====\n");
9612     conjord_imprime_rb(conjunto_a, 1);
9613     printf("= =====\n");
9614 -     printf(" * tecla enter *");
9615 +     printf(" * tecla enter *\n");
9616 +     getc(stdin);
9617
9618     meu_tipo mt1;
9619 @@ -75,7 +75,7 @@ int main()

```

```

9620     printf("= arvore =====\n");
9621     conjord_imprime_rb(conjunto_a, 1);
9622     printf("= =====\n");
9623 -     printf("* tecle enter *");
9624 +     printf("* tecle enter *\n");
9625     getc(stdin);
9626
9627     meu_tipo mt2;
9628 @@ -93,7 +93,7 @@ int main()
9629     printf("= arvore =====\n");
9630     conjord_imprime_rb(conjunto_a, 1);
9631     printf("= =====\n");
9632 -     printf("* tecle enter *");
9633 +     printf("* tecle enter *\n");
9634     getc(stdin);
9635
9636     meu_tipo mt3;
9637 @@ -111,7 +111,7 @@ int main()
9638     printf("= arvore =====\n");
9639     conjord_imprime_rb(conjunto_a, 1);
9640     printf("= =====\n");
9641 -     printf("* tecle enter *");
9642 +     printf("* tecle enter *\n");
9643     getc(stdin);
9644
9645     meu_tipo mt4;
9646 @@ -129,7 +129,7 @@ int main()
9647     printf("= arvore =====\n");
9648     conjord_imprime_rb(conjunto_a, 1);
9649     printf("= =====\n");
9650 -     printf("* tecle enter *");
9651 +     printf("* tecle enter *\n");
9652     getc(stdin);
9653
9654     meu_tipo mt5;
9655 @@ -158,7 +158,7 @@ int main()
9656     printf("= arvore =====\n");
9657     conjord_imprime_rb(conjunto_a, 1);
9658     printf("= =====\n");
9659 -     printf("* tecle enter *");
9660 +     printf("* tecle enter *\n");
9661     getc(stdin);
9662
9663     meu_tipo mt7;
9664 @@ -175,7 +175,7 @@ int main()
9665     printf("= arvore =====\n");
9666     conjord_imprime_rb(conjunto_a, 1);
9667     printf("= =====\n");
9668 -     printf("* tecle enter *");
9669 +     printf("* tecle enter *\n");
9670     getc(stdin);
9671
9672     rb_pr(conjunto_a);
9673 @@ -189,8 +189,7 @@ int main()
9674     printf("= arvore =====\n");
9675     conjord_imprime_rb(conjunto_a, 1);
9676     printf("= =====\n");
9677 -     printf("\n");
9678 -     printf("* tecle enter *");
9679 +     printf("* tecle enter *\n");
9680     getc(stdin);
9681     rb_pr(conjunto_a);
9682     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
9683 @@ -201,7 +200,7 @@ int main()
9684     printf("= arvore =====\n");
9685     conjord_imprime_rb(conjunto_a, 1);
9686     printf("= =====\n");
9687 -     printf("* tecle enter *");
9688 +     printf("* tecle enter *\n");
9689     getc(stdin);
9690     rb_pr(conjunto_a);
9691     rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
9692 @@ -210,7 +209,7 @@ int main()
9693     printf("= arvore =====\n");
9694     conjord_imprime_rb(conjunto_a, 1);
9695     printf("= =====\n");
9696 -     printf("* tecle enter *");
9697 +     printf("* tecle enter *\n");
9698     getc(stdin);
9699     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
9700     printf("\n nada\n");
9701
9702 commit 98d87c66232c4fe8c1f700d6cffe3b3ea766b4d5
9703 Author: Joilnen <joilnen@gmail.com>
9704 Date: Wed Dec 22 17:41:22 2021 -0300
9705
9706     ajustando demonstracao de arvore rb
9707
9708 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
9709 index e79ff10..fbf8c1f 100644
9710 --- a/src/conjunto_ordenado.c
9711 +++ b/src/conjunto_ordenado.c
9712 @@ -48,7 +48,7 @@ void conjord_imprime_rb(rb_n *t, int s) {
9713     s += COUNT;
9714
9715     // Process l child
9716 -     conjord_imprime_rb(t->l, s);
9717 +     conjord_imprime_rb(t->r, s);
9718
9719     // Print current node after space
9720     // count
9721 @@ -61,7 +61,7 @@ void conjord_imprime_rb(rb_n *t, int s) {
9722     printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
9723
9724     // Process right child first
9725 -     conjord_imprime_rb(t->r, s);
9726 +     conjord_imprime_rb(t->l, s);
9727 }
9728
9729
9730
9731 commit 87f35c13d0197242d6b5e7fa87785cdcd8f9e81a
9732 Author: Joilnen <joilnen@gmail.com>
9733 Date: Wed Dec 22 19:44:38 2021 -0300
9734
9735     demonstracao da insercao e delecao na arvore rb no main.c

```

```

9736 diff --git a/src/main.c b/src/main.c
9737 index 2da400d..df889e7 100644
9738 --- a/src/main.c
9739 +++ b/src/main.c
9740 @@ -9,6 +9,12 @@
9741 #include <stdio.h>
9742 #include <string.h>
9743
9744 +void pula_tela()
9745 +{
9746 +    int i;
9747 +    for (i = 0; i < 21; i++)
9748 +        printf("\n");
9749 +}
9750 /**
9751  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
9752  * @noh rb
9753  @@ -52,7 +58,9 @@ int main()
9754     */
9755     rb_n *conjunto_a = rb_cr(10, &mt0);
9756
9757     pula_tela();
9758     rb_pr(conjunto_a);
9759 +    printf("\n");
9760 +    printf("= arvore =====\n");
9761 +    conjord_imprime_rb(conjunto_a, 1);
9762 +    printf("= =====\n");
9763 @@ -71,7 +79,9 @@ int main()
9764     rb_insert(&conjunto_a, novo_noh);
9765
9766     pula_tela();
9767     rb_pr(conjunto_a);
9768 +    printf("\n");
9769 +    printf("= arvore =====\n");
9770 +    conjord_imprime_rb(conjunto_a, 1);
9771 +    printf("= =====\n");
9772 @@ -89,7 +99,9 @@ int main()
9773     novo_noh->d = &mt2;
9774     rb_insert(&conjunto_a, novo_noh);
9775
9776     pula_tela();
9777     rb_pr(conjunto_a);
9778 +    printf("\n");
9779 +    printf("= arvore =====\n");
9780 +    conjord_imprime_rb(conjunto_a, 1);
9781 +    printf("= =====\n");
9782 @@ -107,7 +119,9 @@ int main()
9783     novo_noh->d = &mt3;
9784     rb_insert(&conjunto_a, novo_noh);
9785
9786     pula_tela();
9787     rb_pr(conjunto_a);
9788 +    printf("\n");
9789 +    printf("= arvore =====\n");
9790 +    conjord_imprime_rb(conjunto_a, 1);
9791 +    printf("= =====\n");
9792 @@ -125,7 +139,9 @@ int main()
9793     novo_noh->d = &mt4;
9794     rb_insert(&conjunto_a, novo_noh);
9795
9796     pula_tela();
9797     rb_pr(conjunto_a);
9798 +    printf("\n");
9799 +    printf("= arvore =====\n");
9800 +    conjord_imprime_rb(conjunto_a, 1);
9801 +    printf("= =====\n");
9802 @@ -143,7 +159,14 @@ int main()
9803     novo_noh->d = &mt5;
9804     rb_insert(&conjunto_a, novo_noh);
9805
9806     pula_tela();
9807     rb_pr(conjunto_a);
9808 +    printf("\n");
9809 +    printf("= arvore =====\n");
9810 +    conjord_imprime_rb(conjunto_a, 1);
9811 +    printf("= =====\n");
9812 +    printf("* tecla enter *\n");
9813
9814     meu_tipo mt6;
9815     mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
9816     strcpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
9817 @@ -154,7 +178,9 @@ int main()
9818     novo_noh->d = &mt6;
9819     rb_insert(&conjunto_a, novo_noh);
9820
9821     pula_tela();
9822     rb_pr(conjunto_a);
9823 +    printf("\n");
9824 +    printf("= arvore =====\n");
9825 +    conjord_imprime_rb(conjunto_a, 1);
9826 +    printf("= =====\n");
9827 @@ -172,30 +198,61 @@ int main()
9828     novo_noh->d = &mt7;
9829     rb_insert(&conjunto_a, novo_noh);
9830
9831     pula_tela();
9832     rb_pr(conjunto_a);
9833 +    printf("\n");
9834 +    printf("= arvore =====\n");
9835 +    conjord_imprime_rb(conjunto_a, 1);
9836 +    printf("= =====\n");
9837 +    printf("* tecla enter *\n");
9838     getch(stdin);
9839
9840     rb_pr(conjunto_a);
9841     rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
9842     printf("\n");
9843     pula_tela();
9844     rb_pr(conjunto_a);
9845 +    printf("\n");
9846 +    printf("= arvore =====\n");
9847 +    conjord_imprime_rb(conjunto_a, 1);
9848 +    printf("= =====\n");
9849 +    printf("* tecla enter *\n");
9850
9851

```

```

9852 +     getc(stdin);
9853 +
9854     rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
9855     printf("\n");
9856     pula_tela();
9857     rb_pr(conjunto_a);
9858     printf("= arvore =====\n");
9859     conjord_imprime_rb(conjunto_a, 1);
9860     printf("= =====\n");
9861     printf("* tecla enter *\n");
9862     getc(stdin);
9863 +
9864     rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
9865     printf("\n");
9866     pula_tela();
9867     rb_pr(conjunto_a);
9868     printf("\n");
9869     printf("= arvore =====\n");
9870     conjord_imprime_rb(conjunto_a, 1);
9871     printf("= =====\n");
9872     printf("* tecla enter *\n");
9873     getc(stdin);
9874 +
9875     pula_tela();
9876     rb_pr(conjunto_a);
9877 -     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
9878     printf("\n");
9879 +     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
9880     rb_pr(conjunto_a);
9881     printf("= arvore =====\n");
9882     conjord_imprime_rb(conjunto_a, 1);
9883     printf("= =====\n");
9884     printf("* tecla enter *\n");
9885     getc(stdin);
9886 +
9887     rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
9888     pula_tela();
9889     rb_pr(conjunto_a);
9890     printf("\n");
9891     printf("= arvore =====\n");
9892     conjord_imprime_rb(conjunto_a, 1);
9893 @@ -203,15 +260,20 @@ int main()
9894     printf("* tecla enter *\n");
9895     getc(stdin);
9896     rb_pr(conjunto_a);
9897 +     printf("\n");
9898     rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
9899     printf("\n");
9900 +
9901     pula_tela();
9902     rb_pr(conjunto_a);
9903 +     printf("\n");
9904     printf("= arvore =====\n");
9905     conjord_imprime_rb(conjunto_a, 1);
9906     printf("= =====\n");
9907     printf("* tecla enter *\n");
9908     getc(stdin);
9909     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
9910 +     pula_tela();
9911     printf("\n nada\n");
9912     printf("= arvore =====\n");
9913     conjord_imprime_rb(conjunto_a, 1);
9914 diff --git a/src/red_black.c b/src/red_black.c
9915 index 77e9e98..4223138 100644
9916 --- a/src/red_black.c
9917 +++ b/src/red_black.c
9918 @@ -391,9 +391,9 @@ void rb_pr(rb_n *a)
9919     if (a)
9920     {
9921         if(a->c == BLACK)
9922             printf("\033[44m\033[37mk:%dc:%d\033[0m\n", a->k, a->c);
9923 +         printf("\033[44m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
9924         else
9925             printf("\033[41m\033[37mk:%dc:%d\033[0m\n", a->k, a->c);
9926 +         printf("\033[41m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
9927         rb_pr(a->l);
9928         rb_pr(a->r);
9929     }
9930
9931 commit 7ed984a1847f5984682c35be109ee002f6d6bd75
9932 Author: Joilnen <joilnen@gmail.com>
9933 Date: Thu Dec 23 09:15:10 2021 -0300
9934
9935     melhorando a documentacao relatorio principal
9936
9937 diff --git a/doc/tex/figs/ADT.jpg b/doc/tex/figs/ADT.jpg
9938 new file mode 100644
9939 index 0000000..8752fad
9940 Binary files /dev/null and b/doc/tex/figs/ADT.jpg differ
9941 diff --git a/doc/tex/fig/golpatterns.png b/doc/tex/figs/golpatterns.png
9942 similarity index 100%
9943 rename from doc/tex/fig/golpatterns.png
9944 rename to doc/tex/figs/golpatterns.png
9945 diff --git a/doc/tex/fig/lista.png b/doc/tex/figs/lista.png
9946 similarity index 100%
9947 rename from doc/tex/fig/lista.png
9948 rename to doc/tex/figs/lista.png
9949 diff --git a/doc/tex/fig/ontologia.pdf b/doc/tex/figs/ontologia.pdf
9950 similarity index 100%
9951 rename from doc/tex/fig/ontologia.pdf
9952 rename to doc/tex/figs/ontologia.pdf
9953 diff --git a/doc/tex/fig/ontologiai.eps b/doc/tex/figs/ontologiai.eps
9954 similarity index 100%
9955 rename from doc/tex/fig/ontologiai.eps
9956 rename to doc/tex/figs/ontologiai.eps
9957 diff --git a/doc/tex/fig/t.jpg b/doc/tex/figs/t.jpg
9958 similarity index 100%
9959 rename from doc/tex/fig/t.jpg
9960 rename to doc/tex/figs/t.jpg
9961 diff --git a/doc/tex/packages.tex b/doc/tex/packages.tex
9962 index 1351cce..5e4756a 100644
9963 --- a/doc/tex/packages.tex
9964 +++ b/doc/tex/packages.tex
9965 @@ -79,7 +79,7 @@
9966     title=Listing \thetcbcounter: #2, #1}
9967

```

```

9968 % Automata packages
9969 -\usepackage{tikz, graphics}
9970 +\usepackage{tikz, graphicx}
9971
9972 \usepackage{pgfplots}
9973 \pgfplotsset{compat=1.16}
9974 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
9975 index aec8bba..ef705d1 100644
9976 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
9977 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
9978 index 829d600..9c07db1 100644
9979 --- a/doc/tex/relatorio.tex
9980 +++ b/doc/tex/relatorio.tex
9981 @@ -84,12 +84,22 @@ da atividade conjunto ordenado \\
9982 \indent \textbf{Palavras-chave: } fontes, C, Red Black, Conjunto\\
9983 \small
9984 \section{Introdução}
9985 -Esta biblioteca é composta por dois arquivos, conjunto\_ordenado.c e seu cabeçalho,
9986 -conjunto\_ordenado.h, sendo necessário apenas para seu uso a inclusão via, \\
9987 -\enfn[#include "conjunto\_ordenado.h"] e compilação conjunta com conjunto\_ordenado.c
9988 +\noindent Esta biblioteca é composta por quatro arquivos, \\
9989 +\enfn{conjunto\_ordenado.h}\\
9990 +\enfn{conjunto\_ordenado.c}\\
9991 +\enfn{red\_black.h}\\
9992 +\enfn{red\_black.c}\\
9993 +sendo necessário apenas para seu uso a inclusão via, \\
9994 +\enfn[#include "conjunto\_ordenado.h"] onde se encontra declarado, sendo definido em \\
9995 +\enfn{"conjunto\_ordenado.c"}, o TAD \footnote{Tipo Abstrato de Dados}
9996 +\enfn{conjord} que oculta do código cliente e manipula internamente a estrutura de dados
9997 +\red black \enfn{"struct rb\_n *"} aqui neste trabalho, declarada em \enfn{red\_black.h} e definida em
9998 +\enfn{red\_black.c}
9999
10000 Além da convencional inclusão e compilação, o usuário da biblioteca, pessoa
10001 -que escreve o programa cliente, aqui neste trabalho exemplificado pelo o arquivo \enfn{main.c}
10002 +que escreve o programa cliente, aqui neste trabalho exemplificado pelo os arquivos \\
10003 +\enfn{main.c}\\
10004 +\enfn{test\_tree.c}\\
10005 +deverá escolher definir seu tipo dado, através de uma struct ou utilizar um tipo de dado nativo da
10006 +linguagem.
10007 +\renewcommand{\lstlistingname}{Fonte}
10008 @@ -192,10 +202,18 @@ void (*imprime_elemento)(rb\_n *);
10009 int seta_imprime_elemento(void (*i)(rb\_n *n));
10010 \end{lstlisting}
10011
10012 -% \section{Referências Bibliográficas}
10013 +\begin{figure}
10014 + \begin{center}
10015 + \includegraphics [scale = .4] {figs/ADT}
10016 + \caption{Diagrama TAD}
10017 + \end{center}
10018 +\end{figure}
10019 +
10020 +\section{Referências Bibliográficas}
10021 +\nocite{*}
10022 +\bibliography{mybib.bib}{ }
10023 +\bibliographystyle{plain}
10024 +\href{https://www.geeksforgeeks.org/abstract-data-types/} {Diagrama TAD - \color{blue} https://www.geeksforgeeks.org/
10025 + abstract-data-types/}
10026
10027 \end {document}
10028
10029 diff --git a/src/main.c b/src/main.c
10030 index df889e7..bb9a986 100644
10031 --- a/src/main.c
10032 +++ b/src/main.c
10033 @@ -9,12 +9,20 @@
10034 #include <stdio.h>
10035 #include <string.h>
10036
10037 -void pula_tela()
10038 -{
10039 - int i;
10040 - for (i = 0; i < 21; i++)
10041 - printf("\n");
10042 +/**
10043 + * foi utilizado macro em vez de funcao
10044 + * pra evitar o footprint da chanada
10045 + * pois essa funcao limpa a tela entao
10046 + * varias a vezes eh preferivel q seja inline
10047 + * mas como n tem funcao inline usei a macro
10048 + */
10049 +\define pula_tela() \
10050 +{ \
10051 + int i; \
10052 + for (i = 0; i < 21; i++) \
10053 + printf("\n"); \
10054 +}
10055 +/**
10056 + * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
10057 + * @noh rb
10058 @@ -219,9 +227,9 @@ int main()
10059 getc(stdin);
10060
10061 rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
10062 - printf("\n");
10063 pula_tela();
10064 rb_pr(conjunto_a);
10065 + printf("\n");
10066 printf("= arvore =====\n");
10067 conjord_imprime_rb(conjunto_a, 1);
10068 printf("= =====\n");
10069 @@ -229,7 +237,6 @@ int main()
10070 getc(stdin);
10071
10072 rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
10073 - printf("\n");
10074 pula_tela();
10075 rb_pr(conjunto_a);
10076 - printf("\n");
10077 @@ -239,11 +246,10 @@ int main()
10078 printf("= tecla enter *\n");
10079 getc(stdin);
10080
10081 + rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
10082 pula_tela();

```



```

10083     rb_pr(conjunto_a);
10084     printf("\n");
10085     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
10086     rb_pr(conjunto_a);
10087     printf("= arvore =====\n");
10088     conjord_imprime_rb(conjunto_a, 1);
10089     printf("= =====\n");
10090
10091     commit f479dcc6b016d0977efcc69d87a3fb76b4b15306
10092     Author: Joilnen <joilnen@gmail.com>
10093     Date: Thu Dec 23 11:22:57 2021 -0300
10094
10095     lidando com a limpeza de tela
10096
10097     diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
10098     index fbf8c1f..006050e 100644
10099     --- a/src/conjunto_ordenado.c
10100     +++ b/src/conjunto_ordenado.c
10101     @@ -4,10 +4,11 @@
10102     /**
10103      * cria conjunto ordenado e retorna o ponteiro para ele
10104      */
10105     -conjord *conjord_cria()
10106     +conjord *conjord_cria(char *nome)
10107     {
10108         conjord *c = (conjord *) malloc (sizeof(conjord));
10109         c->raiz = NULL;
10110         + c->nome = nome;
10111         c->cardinalidade = 0;
10112
10113         return c;
10114     diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
10115     index 5061946..2b291a6 100644
10116     --- a/src/conjunto_ordenado.h
10117     +++ b/src/conjunto_ordenado.h
10118     @@ -14,13 +14,14 @@
10119     /**
10120     typedef struct conjord {
10121         rb_n *raiz;
10122         + char *nome;
10123         int cardinalidade;
10124     } conjord;
10125
10126     /**
10127      * cria conjunto ordenado e retorna o ponteiro para ele
10128      */
10129     -conjord *conjord_cria();
10130     +conjord *conjord_cria(char *nome);
10131
10132     /**
10133      * imprime elementos do conjunto subitem 1 do item 2
10134     diff --git a/src/main.c b/src/main.c
10135     index bb9a986..6a2f197 100644
10136     --- a/src/main.c
10137     +++ b/src/main.c
10138     @@ -10,19 +10,17 @@
10139     #include <string.h>
10140
10141     /**
10142     - * foi utilizado macro em vez de funcao
10143     - * pra evitar o footprint da chamada
10144     - * pois essa funcao limpa a tela entao
10145     + * foi utilizado macro em vez de funcao
10146     + * pra evitar o footprint da chamada
10147     + * pois essa funcao limpa a tela entao chamada
10148     + * varias a vezes eh preferivel q seja inline
10149     + * mas como n tem funcao inline usei a macro
10150     + * foi primeiro implementado com um loop
10151     + * mas pensando melhor dessa forma mais rapido
10152     */
10153     +static char aux_pula_tela[22];
10154     #define pula_tela() \
10155     -{ \
10156         int i; \
10157         for (i = 0; i < 21; i++) \
10158             printf("\n"); \
10159     -}
10160     + printf("%s", aux_pula_tela);
10161     /**
10162      * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
10163      * @noh rb
10164     @@ -34,6 +32,8 @@ void imprime_meu_tipo(rb_n *);
10165     */
10166     int main()
10167     {
10168         /** seta scroll da tela */
10169         + memset(aux_pula_tela, '\n', 21);
10170         /**
10171          * definicao de um tipo exemplo pelo o usuario
10172          */
10173
10174     commit 61e26de8f07d32e6b0fd6f50e21fd984ea598e08
10175     Author: Joilnen <joilnen@gmail.com>
10176     Date: Thu Dec 23 18:43:08 2021 -0300
10177
10178     adicionado parametro pra suportar terminais sem cor
10179
10180     diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
10181     index 006050e..93520af 100644
10182     --- a/src/conjunto_ordenado.c
10183     +++ b/src/conjunto_ordenado.c
10184     @@ -38,31 +38,41 @@ int conjord_esvazia(conjord *a)
10185
10186     /**
10187      * imprime arvore
10188      * @param t noh raiz
10189      * @param s espaco entre os nohs na impressao
10190      * @param e tipo do terminal
10191      */
10192     #define COUNT 10
10193     -void conjord_imprime_rb(rb_n *t, int s) {
10194     -    // Base case
10195     +void conjord_imprime_rb(rb_n *t, int s, int e) {
10196     +    const int dist = 8;
10197     +    if (!t)
10198

```

```

10199     return;
10200
10201 - // Increase distance between levels
10202 - s += COUNT;
10203 + s += dist;
10204
10205 - // Process l child
10206 - conjord_imprime_rb(t->r, s);
10207 + conjord_imprime_rb(t->r, s, e);
10208
10209 - // Print current node after space
10210 - // count
10211 - printf("\n");
10212 - for (int i = COUNT; i < s; i++)
10213 -     printf(" ");
10214 + // for (int i = dist; i < s; i++)
10215 + //     printf(" ");
10216 + // printf("%*s", s, "");
10217 + printf("%*c", s - dist, ' ');
10218 if(t->c == BLACK)
10219     printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
10220 + {
10221     if(e == TCOLOR)
10222 +     printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
10223 +     else
10224 +     printf("k:%dc:%d\n", t->k, t->c);
10225 + }
10226 + }
10227 - else
10228 -     printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
10229 + {
10230 +     if(e == TCOLOR)
10231 +     printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
10232 +     else
10233 +     printf("k:%dc:%d\n", t->k, t->c);
10234 + }
10235
10236 - // Process right child first
10237 - conjord_imprime_rb(t->l, s);
10238 + conjord_imprime_rb(t->l, s, e);
10239 }
10240
10241 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
10242 index 2b291a6..872dd50 100644
10243 --- a/src/conjunto_ordenado.h
10244 +++ b/src/conjunto_ordenado.h
10245 @@ -93,7 +93,7 @@ conjord *conjord_subtrai(conjord *a, conjord *b);
10246 * nesta operacao, cada noh da arvore serah
10247 * representado pela sua chave de identificacao seguida da sua cor"
10248 */
10249
10250 -void conjord_imprime_rb(rb_n *t, int s);
10251 +void conjord_imprime_rb(rb_n *t, int s, int e);
10252
10253 #endif
10254
10255 diff --git a/src/main.c b/src/main.c
10256 index 6a2f197..89ce9d7 100644
10257 --- a/src/main.c
10258 +++ b/src/main.c
10259 @@ -12,20 +12,21 @@
10260 /**
10261  * foi utilizado macro em vez de funcao
10262  * pra evitar o footprint da chamada
10263  * pois essa funcao limpa a tela entao chama
10264  * pois essa funcao limpa a tela e eh chamada
10265  * varias a vezes eh preferivel q seja inline
10266  * mas como n tem funcao inline usei a macro
10267  * foi primeiro implementado com um loop
10268  * mas pensando melhor dessa forma mais rapido
10269  */
10270 -static char aux_pula_tela[22];
10271 +static char aux_pula_tela[TELA_PADRAO_FULL + 1];
10272 #define TELA_PADRAO_FULL 45
10273 #define pula_tela() \
10274     printf("%s", aux_pula_tela);
10275 /**
10276  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
10277  * @noh rb
10278  */
10279 -void imprime_meu_tipo(rb_n *);
10280 +void imprime_meu_tipo(void *);
10281
10282 /**
10283  * funcao main de teste separada do codigo da biblioteca como especificado
10284  * @ -33,7 +34,8 @@ void imprime_meu_tipo(rb_n *);
10285  int main()
10286  {
10287     /** seta scroll da tela */
10288     memset(aux_pula_tela, '\n', 21);
10289     memset(aux_pula_tela, '\n', TELA_PADRAO_FULL);
10290 +
10291     /** definicao de um tipo exemplo pelo o usuario
10292     */
10293 @@ -67,10 +69,10 @@ int main()
10294     rb_n *conjunto_a = rb_cr(10, &mt0);
10295
10296     pula_tela();
10297     rb_pr(conjunto_a);
10298 + rb_pr(conjunto_a, TCOLOR);
10299     printf("\n");
10300     printf("= arvore =====\n");
10301     conjord_imprime_rb(conjunto_a, 1);
10302 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10303     printf("= =====\n");
10304     printf("= * tecla enter *\n");
10305     getc(stdin);
10306 @@ -88,10 +90,10 @@ int main()
10307     rb_insert(&conjunto_a, novo_noh);
10308
10309     pula_tela();
10310     rb_pr(conjunto_a);
10311 + rb_pr(conjunto_a, TCOLOR);
10312     printf("\n");
10313     printf("= arvore =====\n");
10314

```

```

10315 - conjord_imprime_rb(conjunto_a, 1);
10316 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10317 printf("= =====\n");
10318 printf("* tecle enter *\n");
10319 getc(stdin);
10320 @@ -108,10 +110,10 @@ int main()
10321 rb_insert(&conjunto_a, novo_noh);
10322
10323 pula_tela();
10324 - rb_pr(conjunto_a);
10325 + rb_pr(conjunto_a, TCOLOR);
10326 printf("\n");
10327 printf("= arvore =====\n");
10328 - conjord_imprime_rb(conjunto_a, 1);
10329 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10330 printf("= =====\n");
10331 printf("* tecle enter *\n");
10332 getc(stdin);
10333 @@ -128,10 +130,10 @@ int main()
10334 rb_insert(&conjunto_a, novo_noh);
10335
10336 pula_tela();
10337 - rb_pr(conjunto_a);
10338 + rb_pr(conjunto_a, TCOLOR);
10339 printf("\n");
10340 printf("= arvore =====\n");
10341 - conjord_imprime_rb(conjunto_a, 1);
10342 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10343 printf("= =====\n");
10344 printf("* tecle enter *\n");
10345 getc(stdin);
10346 @@ -148,10 +150,10 @@ int main()
10347 rb_insert(&conjunto_a, novo_noh);
10348
10349 pula_tela();
10350 - rb_pr(conjunto_a);
10351 + rb_pr(conjunto_a, TCOLOR);
10352 printf("\n");
10353 printf("= arvore =====\n");
10354 - conjord_imprime_rb(conjunto_a, 1);
10355 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10356 printf("= =====\n");
10357 printf("* tecle enter *\n");
10358 getc(stdin);
10359 @@ -168,10 +170,10 @@ int main()
10360 rb_insert(&conjunto_a, novo_noh);
10361
10362 pula_tela();
10363 - rb_pr(conjunto_a);
10364 + rb_pr(conjunto_a, TCOLOR);
10365 printf("\n");
10366 printf("= arvore =====\n");
10367 - conjord_imprime_rb(conjunto_a, 1);
10368 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10369 printf("= =====\n");
10370 printf("* tecle enter *\n");
10371
10372 @@ -187,10 +189,10 @@ int main()
10373 rb_insert(&conjunto_a, novo_noh);
10374
10375 pula_tela();
10376 - rb_pr(conjunto_a);
10377 + rb_pr(conjunto_a, TCOLOR);
10378 printf("\n");
10379 printf("= arvore =====\n");
10380 - conjord_imprime_rb(conjunto_a, 1);
10381 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10382 printf("= =====\n");
10383 printf("* tecle enter *\n");
10384 getc(stdin);
10385 @@ -207,10 +209,10 @@ int main()
10386 rb_insert(&conjunto_a, novo_noh);
10387
10388 pula_tela();
10389 - rb_pr(conjunto_a);
10390 + rb_pr(conjunto_a, TCOLOR);
10391 printf("\n");
10392 printf("= arvore =====\n");
10393 - conjord_imprime_rb(conjunto_a, 1);
10394 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10395 printf("= =====\n");
10396 printf("* tecle enter *\n");
10397 getc(stdin);
10398 @@ -218,63 +220,63 @@ int main()
10399 rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
10400 printf("\n");
10401 pula_tela();
10402 - rb_pr(conjunto_a);
10403 + rb_pr(conjunto_a, TCOLOR);
10404 printf("\n");
10405 printf("= arvore =====\n");
10406 - conjord_imprime_rb(conjunto_a, 1);
10407 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10408 printf("= =====\n");
10409 printf("* tecle enter *\n");
10410 getc(stdin);
10411
10412 rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
10413 pula_tela();
10414 - rb_pr(conjunto_a);
10415 + rb_pr(conjunto_a, TCOLOR);
10416 printf("\n");
10417 printf("= arvore =====\n");
10418 - conjord_imprime_rb(conjunto_a, 1);
10419 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10420 printf("= =====\n");
10421 printf("* tecle enter *\n");
10422 getc(stdin);
10423
10424 rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
10425 pula_tela();
10426 - rb_pr(conjunto_a);
10427 + rb_pr(conjunto_a, TCOLOR);
10428 printf("\n");
10429 printf("= arvore =====\n");
10430 - conjord_imprime_rb(conjunto_a, 1);

```

```

10431 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10432 printf("= =====\n");
10433 printf("* tecla enter *\n");
10434 getc(stdin);
10435
10436 rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
10437 pula_tela();
10438 - rb_pr(conjunto_a);
10439 + rb_pr(conjunto_a, TCOLOR);
10440 printf("\n");
10441 printf("= arvore =====\n");
10442 - conjord_imprime_rb(conjunto_a, 1);
10443 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10444 printf("= =====\n");
10445 printf("* tecla enter *\n");
10446 getc(stdin);
10447
10448 rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
10449 pula_tela();
10450 - rb_pr(conjunto_a);
10451 + rb_pr(conjunto_a, TCOLOR);
10452 printf("\n");
10453 printf("= arvore =====\n");
10454 - conjord_imprime_rb(conjunto_a, 1);
10455 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10456 printf("= =====\n");
10457 printf("* tecla enter *\n");
10458 getc(stdin);
10459 - rb_pr(conjunto_a);
10460 + rb_pr(conjunto_a, TCOLOR);
10461 printf("\n");
10462 rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
10463 printf("\n");
10464
10465 pula_tela();
10466 - rb_pr(conjunto_a);
10467 + rb_pr(conjunto_a, TCOLOR);
10468 printf("\n");
10469 printf("= arvore =====\n");
10470 - conjord_imprime_rb(conjunto_a, 1);
10471 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10472 printf("= =====\n");
10473 printf("* tecla enter *\n");
10474 getc(stdin);
10475 @@ -282,9 +284,9 @@ int main()
10476 pula_tela();
10477 printf("\n nada\n");
10478 printf("= arvore =====\n");
10479 - conjord_imprime_rb(conjunto_a, 1);
10480 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10481 printf("= =====\n");
10482 - rb_pr(conjunto_a);
10483 + rb_pr(conjunto_a, TCOLOR);
10484 printf("\n");
10485 /**
10486  * testando apenas rb no momento
10487 @@ -297,7 +299,7 @@ int main()
10488  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
10489  * exemplo
10490  */
10491 -void imprime_meu_tipo(rb_n *n)
10492 +void imprime_meu_tipo(void *n)
10493 {
10494 }
10495 }
10496 diff --git a/src/red_black.c b/src/red_black.c
10497 index 4223138..4e7dbbb 100644
10498 --- a/src/red_black.c
10499 +++ b/src/red_black.c
10500 @@ -7,7 +7,7 @@
10501 #include "red_black.h"
10502 #include <stdio.h>
10503
10504 -void (*rb_impr_elem)(rb_n *);
10505 +void (*rb_impr_elem)(void *);
10506
10507 rb_n* rb_cr(int key, void *dado)
10508 {
10509 @@ -381,21 +381,31 @@ int rb_rot_dir(rb_n **t, rb_n *x)
10510     return 0;
10511 }
10512
10513 -int rb_set_impr_elem(void (*i)(rb_n *))
10514 +int rb_set_impr_elem(void (*i)(void *))
10515 {
10516     rb_impr_elem = i;
10517 }
10518
10519 -void rb_pr(rb_n *a)
10520 +void rb_pr(rb_n *a, int t)
10521 {
10522     if (a)
10523     {
10524         if (a->c == BLACK)
10525             printf("\033[44m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
10526         if (a->c == BLACK)
10527         {
10528             if (t == TCOLOR)
10529                 printf("\033[44m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
10530             else
10531                 printf("k:%dc:%d ", a->k, a->c);
10532         }
10533         else
10534             printf("\033[41m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
10535         rb_pr(a->l);
10536         rb_pr(a->r);
10537         if (t == TCOLOR)
10538             printf("\033[41m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
10539         else
10540             printf("k:%dc:%d ", a->k, a->c);
10541         rb_pr(a->l, t);
10542         rb_pr(a->r, t);
10543     }
10544 }
10545 }
10546 }

```

```

10547 diff --git a/src/red_black.h b/src/red_black.h
10548 index 2f93388..e1dc33d 100644
10549 --- a/src/red_black.h
10550 +++ b/src/red_black.h
10551 @@ -8,10 +8,18 @@
10552
10553 /**
10554  * definicao do que eh considerado RED e BLACK
10555  * neste codigo para maior clareza
10556  */
10557 + * neste codigo para maior clareza, resolvi usar
10558 + * enum no lugar de defines por ser mais seguro
10559 #define BLACK 0
10560 #define RED 1
10561 + */
10562 +enum {BLACK = 0, RED};
10563 +
10564 +/**
10565  * tipo de terminal se suporta formatacao e cor
10566  * ou se eh monocromatico simples
10567  */
10568 +enum {TMONO, TCOLOR};
10569
10570 /*
10571  * rubro negra
10572  @@ -143,16 +151,18 @@ int rb_rot_dir(rb_n **t, rb_n *x);
10573  */
10574  * imprime elemento como definido pelo o codigo cliente da biblioteca
10575  */
10576 -extern void (*rb_impr_elem)(rb_n *);
10577 +extern void (*rb_impr_elem)(void *);
10578
10579 /**
10580  * seta a rotina de impressao do dado criado pelo cliente
10581  */
10582 -int rb_set_a_impr_elem(void (*)(rb_n *n));
10583 +int rb_set_a_impr_elem(void (*)(void *n));
10584
10585 /**
10586  * imprime arvore @param a
10587  * imprime arvore
10588  * @param a raiz
10589  * @param t tipo de terminal
10590  */
10591 -void rb_pr(rb_n *a);
10592 +void rb_pr(rb_n *a, int t);
10593
10594 #endif
10595
10596 commit 80948706d3ba9f67ac4b72ece8c7037b768ece50
10597 Author: Joilnen <joilnen@gmail.com>
10598 Date: Thu Dec 23 21:11:25 2021 -0300
10599
10600 pequenos ajustes
10601
10602 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
10603 index 93520af..921b7cd 100644
10604 --- a/src/conjunto_ordenado.c
10605 +++ b/src/conjunto_ordenado.c
10606 @@ -68,10 +68,9 @@ void conjord_imprime_rb(rb_n *t, int s, int e) {
10607     if(e == TCOLOR)
10608         printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
10609     else
10610         printf("k:%dc:%d\n", t->k, t->c);
10611 +    printf("\033[47m\033[30mk:%dc:%d\033[0m\n", t->k, t->c);
10612 }
10613
10614 // Process right child first
10615 conjord_imprime_rb(t->l, s, e);
10616 }
10617
10618 diff --git a/src/red_black.c b/src/red_black.c
10619 index 4e7dbbb..af583fe 100644
10620 --- a/src/red_black.c
10621 +++ b/src/red_black.c
10622 @@ -402,7 +402,7 @@ void rb_pr(rb_n *a, int t)
10623     if(t == TCOLOR)
10624         printf("\033[41m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
10625     else
10626         printf("k:%dc:%d ", a->k, a->c);
10627 +    printf("\033[47m\033[30mk:%dc:%d\033[0m ", a->k, a->c);
10628 }
10629 rb_pr(a->l, t);
10630 rb_pr(a->r, t);
10631
10632 commit 8596eda2cceb4d4d8d1a7e10e202e4f181be050e
10633 Author: Joilnen <joilnen@gmail.com>
10634 Date: Fri Dec 24 05:35:16 2021 -0300
10635
10636 formatacao
10637
10638 diff --git a/src/red_black.c b/src/red_black.c
10639 index 4e7dbbb..2eeb88a 100644
10640 --- a/src/red_black.c
10641 +++ b/src/red_black.c
10642 @@ -7,8 +7,19 @@
10643 #include "red_black.h"
10644 #include <stdio.h>
10645
10646 /**
10647  * ponteiro para funcao de impressao do tipo
10648  * dado especificado pelo codigo cliente
10649  * @param void * ponteiro dado especificado pelo
10650  * o codigo cliente
10651  */
10652 void (*rb_impr_elem)(void *);
10653
10654 /**
10655  * criacao do noh raiz
10656  * @param k chave escolhida
10657  * @param dado especificado pelo o codigo cliente
10658  */
10659 rb_n* rb_cr(int key, void *dado)
10660 {

```

```

10663     rb_n *a = (rb_n *) malloc(sizeof(rb_n));
10664 @@ -21,10 +32,16 @@ rb_n* rb_cr(int key, void *dado)
10665     return a;
10666 }
10667
10668 /**
10669  * @param a noh raiz
10670  * @param k chave escolhida
10671  * @param e dado especificado pelo o codigo cliente
10672  */
10673 int rb_ins_elem(rb_n *a, int k, void *e)
10674 {
10675     rb_n *it = a;
10676     printf("k %d it->k %d\n", k, it->k);
10677 +
10678     while (it)
10679     {
10680         rb_n *tmp = it;
10681 @@ -172,6 +189,11 @@ static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
10682         v->p = u->p;
10683     }
10684
10685 /**
10686  * @param x noh raiz
10687  * @param retorma o noh mais a esquerda na arvore
10688  * @param ou subarvore
10689  */
10690 static rb_n *tree_minimum(rb_n *x)
10691 {
10692     rb_n *a = x;
10693 @@ -203,7 +225,7 @@ static rb_n* sucessor(rb_n *x)
10694
10695 int rb_delete(rb_n **t, rb_n *z)
10696 {
10697     rb_n *x;
10698 +     rb_n *x = NULL;
10699     rb_n *y = z;
10700     unsigned char yoc;
10701
10702
10703 commit fdc3d51313d507cidd96c2c7c2f45c76dc4d6b69
10704 Author: Joilnen <joilnen@gmail.com>
10705 Date:   Fri Dec 24 06:43:59 2021 -0300
10706
10707     adicionado escape code pra limpar a tela em vez da solucao
10708     ingenua anterior
10709
10710 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
10711 index 872dd50..6eabc30 100644
10712 --- a/src/conjunto_ordenado.h
10713 +++ b/src/conjunto_ordenado.h
10714 @@ -95,6 +95,17 @@ conjord *conjord_subtrai(conjord *a, conjord *b);
10715     */
10716 void conjord_imprime_rb(rb_n *t, int s, int e);
10717
10718 /**
10719  * foi utilizado macro em vez de funcao
10720  * pra evitar o footprint da chamada
10721  * pois essa funcao limpa a tela e eh chamada
10722  * varias a vezes eh preferivel q seja inline
10723  * mas como n tem funcao inline usei a macro
10724  * foi primeiro implementado com um loop
10725  * mas pensando melhor dessa forma mais rapido
10726  */
10727 #define LTELA printf("\033c");
10728 +
10729 #endif
10730
10731
10732 diff --git a/src/main.c b/src/main.c
10733 index 89ce9d7..5643809 100644
10734 --- a/src/main.c
10735 +++ b/src/main.c
10736 @@ -9,19 +9,6 @@
10737 #include <stdio.h>
10738 #include <string.h>
10739
10740 /**
10741  * foi utilizado macro em vez de funcao
10742  * pra evitar o footprint da chamada
10743  * pois essa funcao limpa a tela e eh chamada
10744  * varias a vezes eh preferivel q seja inline
10745  * mas como n tem funcao inline usei a macro
10746  * foi primeiro implementado com um loop
10747  * mas pensando melhor dessa forma mais rapido
10748  */
10749 #define TELA_PADRAO_FULL 45
10750 -static char aux_pula_tela[TELA_PADRAO_FULL + 1];
10751 #define pula_tela() \
10752     printf("%s", aux_pula_tela);
10753 /**
10754  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
10755  * @noh rb
10756 @@ -33,9 +20,6 @@ void imprime_meu_tipo(void *);
10757     */
10758 int main()
10759 {
10760     /** seta scroll da tela */
10761     memset(aux_pula_tela, '\n', TELA_PADRAO_FULL);
10762
10763     /**
10764      * definicao de um tipo exemplo pelo o usuario
10765      */
10766 @@ -44,7 +28,6 @@ int main()
10767     char *telefone;
10768     } meu_tipo;
10769
10770
10771 /**
10772  * criacao do dado para o tipo do usuario
10773  */
10774 @@ -68,7 +51,7 @@ int main()
10775     */
10776     rb_n *conjunto_a = rb_cr(10, &mt0);
10777
10778     pula_tela();

```

```

10779 + LTELA;
10780 rb_pr(conjunto_a, TCOLOR);
10781 printf("\n");
10782 printf("= arvore =====\n");
10783 @@ -89,7 +72,7 @@ int main()
10784
10785 rb_insert(&conjunto_a, novo_noh);
10786
10787 - pula_tela();
10788 + LTELA;
10789 rb_pr(conjunto_a, TCOLOR);
10790 printf("\n");
10791 printf("= arvore =====\n");
10792 @@ -109,7 +92,7 @@ int main()
10793 novo_noh->d = &mt2;
10794 rb_insert(&conjunto_a, novo_noh);
10795
10796 - pula_tela();
10797 + LTELA;
10798 rb_pr(conjunto_a, TCOLOR);
10799 printf("\n");
10800 printf("= arvore =====\n");
10801 @@ -129,7 +112,7 @@ int main()
10802 novo_noh->d = &mt3;
10803 rb_insert(&conjunto_a, novo_noh);
10804
10805 - pula_tela();
10806 + LTELA;
10807 rb_pr(conjunto_a, TCOLOR);
10808 printf("\n");
10809 printf("= arvore =====\n");
10810 @@ -149,7 +132,7 @@ int main()
10811 novo_noh->d = &mt4;
10812 rb_insert(&conjunto_a, novo_noh);
10813
10814 - pula_tela();
10815 + LTELA;
10816 rb_pr(conjunto_a, TCOLOR);
10817 printf("\n");
10818 printf("= arvore =====\n");
10819 @@ -169,7 +152,7 @@ int main()
10820 novo_noh->d = &mt5;
10821 rb_insert(&conjunto_a, novo_noh);
10822
10823 - pula_tela();
10824 + LTELA;
10825 rb_pr(conjunto_a, TCOLOR);
10826 printf("\n");
10827 printf("= arvore =====\n");
10828 @@ -188,7 +171,7 @@ int main()
10829 novo_noh->d = &mt6;
10830 rb_insert(&conjunto_a, novo_noh);
10831
10832 - pula_tela();
10833 + LTELA;
10834 rb_pr(conjunto_a, TCOLOR);
10835 printf("\n");
10836 printf("= arvore =====\n");
10837 @@ -208,7 +191,7 @@ int main()
10838 novo_noh->d = &mt7;
10839 rb_insert(&conjunto_a, novo_noh);
10840
10841 - pula_tela();
10842 + LTELA;
10843 rb_pr(conjunto_a, TCOLOR);
10844 printf("\n");
10845 printf("= arvore =====\n");
10846 @@ -219,7 +202,7 @@ int main()
10847
10848 rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
10849 printf("\n");
10850 - pula_tela();
10851 + LTELA;
10852 rb_pr(conjunto_a, TCOLOR);
10853 printf("\n");
10854 printf("= arvore =====\n");
10855 @@ -229,7 +212,7 @@ int main()
10856 getc(stdin);
10857
10858 rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
10859 - pula_tela();
10860 + LTELA;
10861 rb_pr(conjunto_a, TCOLOR);
10862 printf("\n");
10863 printf("= arvore =====\n");
10864 @@ -239,7 +222,7 @@ int main()
10865 getc(stdin);
10866
10867 rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
10868 - pula_tela();
10869 + LTELA;
10870 rb_pr(conjunto_a, TCOLOR);
10871 printf("\n");
10872 printf("= arvore =====\n");
10873 @@ -249,7 +232,7 @@ int main()
10874 getc(stdin);
10875
10876 rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
10877 - pula_tela();
10878 + LTELA;
10879 rb_pr(conjunto_a, TCOLOR);
10880 printf("\n");
10881 printf("= arvore =====\n");
10882 @@ -259,7 +242,7 @@ int main()
10883 getc(stdin);
10884
10885 rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
10886 - pula_tela();
10887 + LTELA;
10888 rb_pr(conjunto_a, TCOLOR);
10889 printf("\n");
10890 printf("= arvore =====\n");
10891 @@ -272,7 +255,7 @@ int main()
10892 rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
10893 printf("\n");
10894

```

```

10895 - pula_tela();
10896 + LTELA;
10897 rb_pr(conjunto_a, TCOLOR);
10898 printf("\n");
10899 printf("= arvore =====\n");
10900 @@ -281,7 +264,7 @@ int main()
10901 printf("* tecla enter *\n");
10902 getc(stdin);
10903 rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
10904 - pula_tela();
10905 + LTELA;
10906 printf("\n nada\n");
10907 printf("= arvore =====\n");
10908 conjord_imprime_rb(conjunto_a, 1, TCOLOR);
10909 diff --git a/src/test_rb.c b/src/test_rb.c
10910 new file mode 100644
10911 index 0000000..e69de29
10912
10913 commit 12fb73c5f11b9f6992f06cdd7d3e03aa7433a924
10914 Author: Joilnen <joilnen@gmail.com>
10915 Date: Fri Dec 24 07:42:35 2021 -0300
10916
10917 brincando com codigos de tela
10918
10919 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
10920 index 921b7cd..29fb9d2 100644
10921 --- a/src/conjunto_ordenado.c
10922 +++ b/src/conjunto_ordenado.c
10923 @@ -68,7 +68,7 @@ void conjord_imprime_rb(rb_n *t, int s, int e) {
10924     if(e == TCOLOR)
10925         printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
10926     else
10927         printf("\033[47m\033[30mk:%dc:%d\033[0m\n", t->k, t->c);
10928 + printf("\033[7mk:%dc:%d\033[0m\n", t->k, t->c);
10929 }
10930
10931 conjord_imprime_rb(t->l, s, e);
10932 diff --git a/src/red_black.c b/src/red_black.c
10933 index af583fe..73886e4 100644
10934 --- a/src/red_black.c
10935 +++ b/src/red_black.c
10936 @@ -402,7 +402,7 @@ void rb_pr(rb_n *a, int t)
10937     if(t == TCOLOR)
10938         printf("\033[41m\033[37mk:%dc:%d\033[0m ", a->k, a->c);
10939     else
10940         printf("\033[47m\033[30mk:%dc:%d\033[0m ", a->k, a->c);
10941 + printf("\033[7mk:%dc:%d\033[0m ", a->k, a->c);
10942 }
10943 rb_pr(a->l, t);
10944 rb_pr(a->r, t);
10945
10946 commit 310eb2fe1595966da5098ba34c3e661709092d00
10947 Merge: 8596eda 12fb73c
10948 Author: Joilnen <joilnen@gmail.com>
10949 Date: Fri Dec 24 13:44:01 2021 -0300
10950
10951 Merge branch 'master' of 192.168.43.29:conjunto_ordenado
10952
10953 commit 5056ab458a2344e70f3eff78e007eb7c1b5cfd59
10954 Author: Joilnen <joilnen@gmail.com>
10955 Date: Fri Dec 24 23:13:14 2021 -0300
10956
10957 curses codes
10958
10959 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
10960 index 29fb9d2..dc8253d 100644
10961 --- a/src/conjunto_ordenado.c
10962 +++ b/src/conjunto_ordenado.c
10963 @@ -31,6 +31,10 @@ int conjord_eh_vazio(const conjord *a)
10964     return !a->raiz;
10965 }
10966
10967 /**
10968  * * deleta todos os elementos do
10969  * * conjunto e zera a cardinalidade
10970  * */
10971 int conjord_esvazia(conjord *a)
10972 {
10973
10974 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
10975 index 6eabc30..3123fbb 100644
10976 --- a/src/conjunto_ordenado.h
10977 +++ b/src/conjunto_ordenado.h
10978 @@ -1,6 +1,7 @@
10979 #ifndef CONJUNTO_ORDENADO_H
10980 #define CONJUNTO_ORDENADO_H
10981
10982 #include "jcurses.h"
10983 #include "red_black.h"
10984
10985 /**
10986  @@ -104,7 +105,7 @@ void conjord_imprime_rb(rb_n *t, int s, int e);
10987  * foi primeiro implementado com um loop
10988  * mas pensando melhor dessa forma mais rapido
10989  */
10990 #define LTELA printf("\033c");
10991 #define LTELA printf(LIMPA_TELA);
10992
10993 #endif
10994
10995 diff --git a/src/jcurses.h b/src/jcurses.h
10996 new file mode 100644
10997 index 0000000..5d082e2
10998 --- /dev/null
10999 +++ b/src/jcurses.h
10000 @@ -0,0 +1,19 @@
10001 #ifndef JOILNEN_CURSES_H
10002 #define JOILNEN_CURSES_H
10003
10004 +
10005 /**
10006  * * macros que criei baseado no que pesquisei
10007  * * sobre formatacao e codificacao de terminal vt100
10008  * * que eh a base dos emuladores de terminal do linux
10009  * * curses eh como chamam o tipo de software q faz isso
10010  * * curses e ncurses sao exemplos, aqui eh soh uma tentativa
10011  * * primaria que atende as necessidades apenas dessa avaliacao

```



```

11011 + */
11012 +
11013 #define LIMPA_TELA "\033c"
11014 #define S_AZUL "\033[44m\033[37m"
11015 #define S_VERM "\033[41m\033[37m"
11016 #define S_INV "\033[7m"
11017 #define S_NORM "\033[0m"
11018 +
11019 #endif
11020 diff --git a/src/main.c b/src/main.c
11021 index 5643809..1ec8a5a 100644
11022 --- a/src/main.c
11023 +++ b/src/main.c
11024 @@ -252,10 +252,10 @@ int main()
11025     getc(stdin);
11026     rb_pr(conjunto_a, TCOLOR);
11027     printf("\n");
11028     rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
11029     printf("\n");
11030
11031     rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
11032     LTELTA;
11033     printf("\n");
11034     rb_pr(conjunto_a, TCOLOR);
11035     printf("\n");
11036     printf("= arvore =====\n");
11037 @@ -263,6 +263,7 @@ int main()
11038     printf("= =====\n");
11039     printf("= tecla enter *\n");
11040     getc(stdin);
11041 +
11042     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
11043     LTELTA;
11044     printf("\n nada\n");
11045 diff --git a/src/red_black.c b/src/red_black.c
11046 index 342edae..3789e76 100644
11047 --- a/src/red_black.c
11048 +++ b/src/red_black.c
11049 @@ -4,6 +4,7 @@
11050  * de implementacao das funcoes da redblack
11051  * utilizada no conjunto ordenado
11052  */
11053 #include "jurses.h"
11054 #include "red_black.h"
11055 #include <stdio.h>
11056
11057 @@ -279,17 +280,20 @@ int rb_delete(rb_n **t, rb_n *z)
11058     w = x->p->r;
11059     }
11060
11061     if (w->l->c == BLACK && w->r->c == BLACK)
11062     if ((w->l->c == BLACK && w->r->c == BLACK) || (w->l == BLACK && w->r == BLACK))
11063     {
11064         w->c = RED;
11065         if (!w) w->c = RED;
11066         x = x->p;
11067     }
11068     else
11069     {
11070         if (w->r->c == BLACK)
11071         if (w->r == BLACK || w->r->c == BLACK)
11072         {
11073             w->l->c = BLACK;
11074             w->c = RED;
11075             if (w)
11076             {
11077                 w->l->c = BLACK;
11078                 w->c = RED;
11079             }
11080             rb_rot_dir(t, w);
11081             w = x->p->r;
11082         }
11083     }
11084 @@ -335,8 +339,11 @@ int rb_delete(rb_n **t, rb_n *z)
11085     }
11086     }
11087     if(x) x->c = BLACK;
11088     } /* if (yoc == BLACK) */
11089
11090 +
11091 +
11092     return 0;
11093 }
11094
11095 @@ -415,16 +422,16 @@ void rb_pr(rb_n *a, int t)
11096     if (a->c == BLACK)
11097     {
11098         if (t == TCOLOR)
11099             printf("\033[44m\033[37m;dc:%d\033[0m ", a->k, a->c);
11100         printf("%sk:%dc:%d", S_AZUL, a->k, a->c, S_NORM);
11101     }
11102     else
11103         printf("k:%dc:%d ", a->k, a->c);
11104
11105     {
11106         if (t == TCOLOR)
11107             printf("\033[41m\033[37m;dc:%d\033[0m ", a->k, a->c);
11108         if (t == TCOLOR)
11109             printf("%sk:%dc:%d", S_VERM, a->k, a->c, S_NORM);
11110         else
11111             printf("\033[7m;dc:%d\033[0m ", a->k, a->c);
11112         printf("%sk:%dc:%d", S_INV, a->k, a->c, S_NORM);
11113     }
11114     rb_pr(a->l, t);
11115     rb_pr(a->r, t);
11116 diff --git a/src/test_rb.c b/src/test_rb.c
11117 index e69de29..5643809 100644
11118 --- a/src/test_rb.c
11119 +++ b/src/test_rb.c
11120 @@ -0,0 +1,290 @@
11121 +/**
11122 + * aluno: joilnen leite
11123 + */
11124 +
11125 +/**
11126 + * inclusao da biblioteca implementada baseada em arvores red black

```

```

11127 + */
11128 + #include "conjunto_ordenado.h"
11129 + #include <stdio.h>
11130 + #include <string.h>
11131 +
11132 + /**
11133 +  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
11134 +  * @noh rb
11135 +  */
11136 + void imprime_meu_tipo(void *);
11137 +
11138 + /**
11139 +  * funcao main de teste separada do codigo da biblioteca como especificado
11140 +  */
11141 + int main()
11142 + {
11143 +     /**
11144 +      * definicao de um tipo exemplo pelo o usuario
11145 +      */
11146 +     typedef struct {
11147 +         char *nome;
11148 +         char *telefone;
11149 +     } meu_tipo;
11150 +
11151 +     /**
11152 +      * criacao do dado para o tipo do usuario
11153 +      */
11154 +     meu_tipo mt0;
11155 +     mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
11156 +     strncpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
11157 +     mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
11158 +     strncpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
11159 +
11160 +     /**
11161 +      * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
11162 +      */
11163 +     rb_set Impr_elem(imprime_meu_tipo);
11164 +
11165 +     /**
11166 +      * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
11167 +      * uma interface mais intuitiva ao usuario da biblioteca
11168 +      * @parametro 0 nome variavel do noh raiz
11169 +      * @parametro 1 chave do noh raiz
11170 +      * @parametro 2 dado criado neste noh, pode ser de qqr tipo
11171 +      */
11172 +     rb_n *conjunto_a = rb_cr(10, &mt0);
11173 +
11174 +     LTEL;
11175 +     rb_pr(conjunto_a, TCOLOR);
11176 +     printf("\n");
11177 +     printf("= arvore =====\n");
11178 +     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11179 +     printf("= =====\n");
11180 +     printf(" * tecla enter *\n");
11181 +     getc(stdin);
11182 +
11183 +     meu_tipo mt1;
11184 +     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
11185 +     strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
11186 +     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
11187 +     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
11188 +
11189 +     rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
11190 +     novo_noh->k = 9;
11191 +     novo_noh->d = &mt1;
11192 +
11193 +     rb_insert(&conjunto_a, novo_noh);
11194 +
11195 +     LTEL;
11196 +     rb_pr(conjunto_a, TCOLOR);
11197 +     printf("\n");
11198 +     printf("= arvore =====\n");
11199 +     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11200 +     printf("= =====\n");
11201 +     printf(" * tecla enter *\n");
11202 +     getc(stdin);
11203 +
11204 +     meu_tipo mt2;
11205 +     mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
11206 +     strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
11207 +     mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
11208 +     strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
11209 +
11210 +     novo_noh = (rb_n *) malloc (sizeof(rb_n));
11211 +     novo_noh->k = 11;
11212 +     novo_noh->d = &mt2;
11213 +     rb_insert(&conjunto_a, novo_noh);
11214 +
11215 +     LTEL;
11216 +     rb_pr(conjunto_a, TCOLOR);
11217 +     printf("\n");
11218 +     printf("= arvore =====\n");
11219 +     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11220 +     printf("= =====\n");
11221 +     printf(" * tecla enter *\n");
11222 +     getc(stdin);
11223 +
11224 +     meu_tipo mt3;
11225 +     mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
11226 +     strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
11227 +     mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
11228 +     strncpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
11229 +
11230 +     novo_noh = (rb_n *) malloc (sizeof(rb_n));
11231 +     novo_noh->k = 8;
11232 +     novo_noh->d = &mt3;
11233 +     rb_insert(&conjunto_a, novo_noh);
11234 +
11235 +     LTEL;
11236 +     rb_pr(conjunto_a, TCOLOR);
11237 +     printf("\n");
11238 +     printf("= arvore =====\n");
11239 +     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11240 +     printf("= =====\n");
11241 +     printf(" * tecla enter *\n");
11242 +     getc(stdin);

```

```

11243 | +
11244 | + meu_tipo mt4;
11245 | + mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
11246 | + strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
11247 | + mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
11248 | + strncpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
11249 | +
11250 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
11251 | + novo_noh->k = 6;
11252 | + novo_noh->d = &mt4;
11253 | + rb_insert(&conjunto_a, novo_noh);
11254 | +
11255 | + LTELA;
11256 | + rb_pr(conjunto_a, TCOLOR);
11257 | + printf("\n");
11258 | + printf("= arvore =====\n");
11259 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11260 | + printf("= =====\n");
11261 | + printf("* tecle enter *\n");
11262 | + getc(stdin);
11263 | +
11264 | + meu_tipo mt5;
11265 | + mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
11266 | + strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
11267 | + mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
11268 | + strncpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
11269 | +
11270 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
11271 | + novo_noh->k = 5;
11272 | + novo_noh->d = &mt5;
11273 | + rb_insert(&conjunto_a, novo_noh);
11274 | +
11275 | + LTELA;
11276 | + rb_pr(conjunto_a, TCOLOR);
11277 | + printf("\n");
11278 | + printf("= arvore =====\n");
11279 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11280 | + printf("= =====\n");
11281 | + printf("* tecle enter *\n");
11282 | +
11283 | + meu_tipo mt6;
11284 | + mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
11285 | + strncpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
11286 | + mt6.telefone = (char *) malloc(sizeof("999999999") + 1);
11287 | + strncpy(mt6.telefone, "999999999", sizeof("999999999") + 1);
11288 | +
11289 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
11290 | + novo_noh->k = 4;
11291 | + novo_noh->d = &mt6;
11292 | + rb_insert(&conjunto_a, novo_noh);
11293 | +
11294 | + LTELA;
11295 | + rb_pr(conjunto_a, TCOLOR);
11296 | + printf("\n");
11297 | + printf("= arvore =====\n");
11298 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11299 | + printf("= =====\n");
11300 | + printf("* tecle enter *\n");
11301 | + getc(stdin);
11302 | +
11303 | + meu_tipo mt7;
11304 | + mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
11305 | + strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
11306 | + mt7.telefone = (char *) malloc(sizeof("999999999") + 1);
11307 | + strncpy(mt7.telefone, "999999999", sizeof("999999999") + 1);
11308 | +
11309 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
11310 | + novo_noh->k = 3;
11311 | + novo_noh->d = &mt7;
11312 | + rb_insert(&conjunto_a, novo_noh);
11313 | +
11314 | + LTELA;
11315 | + rb_pr(conjunto_a, TCOLOR);
11316 | + printf("\n");
11317 | + printf("= arvore =====\n");
11318 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11319 | + printf("= =====\n");
11320 | + printf("* tecle enter *\n");
11321 | + getc(stdin);
11322 | +
11323 | + rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
11324 | + printf("\n");
11325 | + LTELA;
11326 | + rb_pr(conjunto_a, TCOLOR);
11327 | + printf("\n");
11328 | + printf("= arvore =====\n");
11329 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11330 | + printf("= =====\n");
11331 | + printf("* tecle enter *\n");
11332 | + getc(stdin);
11333 | +
11334 | + rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
11335 | + LTELA;
11336 | + rb_pr(conjunto_a, TCOLOR);
11337 | + printf("\n");
11338 | + printf("= arvore =====\n");
11339 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11340 | + printf("= =====\n");
11341 | + printf("* tecle enter *\n");
11342 | + getc(stdin);
11343 | +
11344 | + rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
11345 | + LTELA;
11346 | + rb_pr(conjunto_a, TCOLOR);
11347 | + printf("\n");
11348 | + printf("= arvore =====\n");
11349 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11350 | + printf("= =====\n");
11351 | + printf("* tecle enter *\n");
11352 | + getc(stdin);
11353 | +
11354 | + rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
11355 | + LTELA;
11356 | + rb_pr(conjunto_a, TCOLOR);
11357 | + printf("\n");
11358 | + printf("= arvore =====\n");

```

```

11359 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11360 + printf("= =====\n");
11361 + printf(" * tecla enter *\n");
11362 + getc(stdin);
11363 +
11364 + rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
11365 + LTEL;
11366 + rb_pr(conjunto_a, TCOLOR);
11367 + printf("\n");
11368 + printf("= arvore =====\n");
11369 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11370 + printf("= =====\n");
11371 + printf(" * tecla enter *\n");
11372 + getc(stdin);
11373 + rb_pr(conjunto_a, TCOLOR);
11374 + printf("\n");
11375 + rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
11376 + printf("\n");
11377 +
11378 + LTEL;
11379 + rb_pr(conjunto_a, TCOLOR);
11380 + printf("\n");
11381 + printf("= arvore =====\n");
11382 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11383 + printf("= =====\n");
11384 + printf(" * tecla enter *\n");
11385 + getc(stdin);
11386 + rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
11387 + LTEL;
11388 + printf("\n nada\n");
11389 + printf("= arvore =====\n");
11390 + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
11391 + printf("= =====\n");
11392 + rb_pr(conjunto_a, TCOLOR);
11393 + printf("\n");
11394 + /**
11395 +  * testando apenas rb no momento
11396 +  * conjord *conj_a = conjord_cria();
11397 +  * printf("eh vaziao %d", conjord_eh_vazio(conj_a));
11398 +  */
11399 +}
11400 +
11401 +/**
11402 +  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
11403 +  * exemplo
11404 +  */
11405 +void imprime_meu_tipo(void *n)
11406 +{
11407 +
11408 +}
11409 +
11410 +
11411 +
11412 +commit 123cacce2fdd798913da438a59c6ab72fa7c7678
11413 +Author: Joilnen <joilnen@gmail.com>
11414 +Date: Sat Dec 25 18:21:49 2021 -0300
11415 +
11416 +    possibilidade de compilar do diretorio base
11417 +
11418 +diff --git a/makefile b/makefile
11419 +new file mode 100644
11420 +index 0000000..2dbc0d8
11421 +--- /dev/null
11422 +++ b/makefile
11423 @@ -0,0 +1,7 @@
11424 +all:
11425 + @make -C src
11426 + @mv src/rodeme .
11427 +clean:
11428 + @make -C src clean
11429 +
11430 +
11431 +
11432 +commit 8de643ddbdc227b4ba939c13dc4f9f2cf329f96
11433 +Author: Joilnen <joilnen@gmail.com>
11434 +Date: Tue Dec 28 18:19:28 2021 -0300
11435 +
11436 +    sentinela
11437 +
11438 +diff --git a/src/red_black.c b/src/red_black.c
11439 +index 3789e76..5c6060e 100644
11440 +--- a/src/red_black.c
11441 +++ b/src/red_black.c
11442 @@ -228,11 +228,10 @@ int rb_delete(rb_n **t, rb_n *z)
11443 +{
11444 +    rb_n *x = NULL;
11445 +    rb_n *y = z;
11446 +    unsigned char yoc;
11447 +    unsigned char yoc = z->c;
11448 +
11449 +    if (!z)
11450 +        return -1;
11451 +    yoc = y->c;
11452 +
11453 +    if (!z->l)
11454 +    {
11455 @@ -280,7 +279,7 @@ int rb_delete(rb_n **t, rb_n *z)
11456 +        w = x->p->r;
11457 +    }
11458 +
11459 +    if ((w->l->c == BLACK && w->r->c == BLACK) || (w->l == BLACK && w->r == BLACK))
11460 +    if (w->l->c == BLACK && w->r->c == BLACK)
11461 +    {
11462 +        if (!w) w->c = RED;
11463 +        x = x->p;
11464 @@ -342,7 +341,132 @@ int rb_delete(rb_n **t, rb_n *z)
11465 +        if (x) x->c = BLACK;
11466 +    } /* if (yoc == BLACK) */
11467 +
11468 +    return 0;
11469 +}
11470 +
11471 +/**
11472 +  * deleta noh da arvore red black
11473 +  * @param t ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao
11474 +  * e manter a alteracao depois do retorno da funcao

```

```

11475 + * @param z ponteiro para o noh a ser deletado, para deletar por chave o TAD deve
11476 + * passar a saida da funcao pesquisar, num arranjo funcional na chamada
11477 + */
11478 +int rb_delete_sentinela(rb_n **t, rb_n *z)
11479 +{
11480 +    rb_n *x = NULL;
11481 +    rb_n *y = z;
11482 +    unsigned char yoc = z->c;
11483 +
11484 +    if (!z)
11485 +        return -1;
11486 +
11487 +    if (!z->l)
11488 +    {
11489 +        x = z->r;
11490 +        rb_transplant(t, z, z->r);
11491 +    }
11492 +    else if (!z->r)
11493 +    {
11494 +        x = z->l;
11495 +        rb_transplant(t, z, z->l);
11496 +    }
11497 +    else
11498 +    {
11499 +        // y = tree_minimum(z->r);
11500 +        y = sucessor(z);
11501 +        yoc = y->c;
11502 +        x = y->r;
11503 +        if (y->p == z)
11504 +            if (x) x->p = y;
11505 +        else
11506 +        {
11507 +            rb_transplant(t, y, y->r);
11508 +            y->r = z->r;
11509 +            y->r->p = y;
11510 +        }
11511 +        rb_transplant(t, z, y);
11512 +        y->l = z->l;
11513 +        y->l->p = y;
11514 +        y->c = z->c;
11515 +    }
11516 +
11517 +    if (yoc == BLACK)
11518 +    {
11519 +        rb_n *w;
11520 +        while (x && x != *t && x->c == BLACK)
11521 +        {
11522 +            if (x == x->p->l)
11523 +            {
11524 +                w = x->p->r;
11525 +                if (w->c == RED)
11526 +                {
11527 +                    w->c = BLACK;
11528 +                    x->p->c = RED;
11529 +                    rb_rot_esq(t, x->p);
11530 +                    w = x->p->r;
11531 +                }
11532 +
11533 +                if (w->l->c == BLACK && w->r->c == BLACK)
11534 +                {
11535 +                    if (!w) w->c = RED;
11536 +                    x = x->p;
11537 +                }
11538 +                else
11539 +                {
11540 +                    if (w->r == BLACK || w->r->c == BLACK)
11541 +                    {
11542 +                        if (w)
11543 +                        {
11544 +                            w->l->c = BLACK;
11545 +                            w->c = RED;
11546 +                        }
11547 +                        rb_rot_dir(t, w);
11548 +                        w = x->p->r;
11549 +                    }
11550 +
11551 +                    w->c = x->p->c;
11552 +                    x->p->c = BLACK;
11553 +                    w->r->c = BLACK;
11554 +                    rb_rot_esq(t, x->p);
11555 +                    x = *t;
11556 +                }
11557 +            }
11558 +            else
11559 +            {
11560 +                w = x->p->l;
11561 +                if (w->c == RED)
11562 +                {
11563 +                    w->c = BLACK;
11564 +                    x->p->c = RED;
11565 +                    rb_rot_esq(t, x->p);
11566 +                    w = x->p->l;
11567 +                }
11568 +
11569 +                if (w->r->c == BLACK && w->l->c == BLACK)
11570 +                {
11571 +                    w->c = RED;
11572 +                    x = x->p;
11573 +                }
11574 +                else
11575 +                {
11576 +                    if (w->l->c == BLACK)
11577 +                    {
11578 +                        w->r->c = BLACK;
11579 +                        w->c = RED;
11580 +                        rb_rot_dir(t, w);
11581 +                        w = x->p->l;
11582 +                    }
11583 +
11584 +                    w->c = x->p->c;
11585 +                    x->p->c = BLACK;
11586 +                    w->l->c = BLACK;
11587 +                    rb_rot_esq(t, x->p);
11588 +                    x = *t;
11589 +                }
11590 +            }
11591 +        }
11592 +    }

```

```

11591 +     }
11592 +     if(x) x->c = BLACK;
11593 + } /* if (yoc == BLACK) */
11594
11595     return 0;
11596 }
11597 @@ -458,4 +582,38 @@ rb_n *rb_search(rb_n *t, int k)
11598     return NULL;
11599 }
11600
11601 +void rb_converte_null_sentinela(rb_n *t, rb_n *sentinela)
11602 +{
11603 +    if(t)
11604 +    {
11605 +        if(!t->p)
11606 +        {
11607 +            t->p = sentinela;
11608 +            t->sentinela = 1;
11609 +        }
11610 +        rb_converte_null_sentinela(t->l, sentinela);
11611 +        rb_converte_null_sentinela(t->r, sentinela);
11612 +    }
11613 +    t = sentinela;
11614 +}
11615
11616 +void rb_converte_sentinela_null(rb_n *t)
11617 +{
11618 +    rb_n *tmp;
11619 +    if(t)
11620 +    {
11621 +        if(t->p->sentinela)
11622 +        {
11623 +            tmp = t->p;
11624 +            t->p = NULL;
11625 +        }
11626 +        rb_converte_sentinela_null(t->l);
11627 +        rb_converte_sentinela_null(t->r);
11628 +    }
11629 +    t = NULL;
11630 +    free(tmp);
11631 +}
11632
11633 diff --git a/src/red_black.h b/src/red_black.h
11634 index e1dc33d..a42e6ae 100644
11635 --- a/src/red_black.h
11636 +++ b/src/red_black.h
11637 @@ -13,13 +13 @@
11638 #define BLACK 0
11639 #define RED 1
11640
11641 -enum {BLACK = 0, RED};
11642 +typedef enum cor_noh {BLACK = 0, RED} cor_noh;
11643
11644 /**
11645  * tipo de terminal se suporta formatacao e cor
11646  * ou se eh monocromatico simples
11647  */
11648 -enum {TMONO, TCOLOR};
11649 +typedef enum tipo_screen {TMONO, TCOLOR} tipo_screen;
11650
11651 /*
11652  * rubro negra
11653  @@ -46,6 +46,7 @@ typedef struct rb_n {
11654     unsigned char c:1;
11655     void *d;
11656     unsigned int np;
11657     unsigned char sentinela:1;
11658     struct rb_n *l, *r, *p;
11659 } rb_n;
11660
11661 @@ -88,6 +89,13 @@ rb_n *rb_search(rb_n *t, int k);
11662 */
11663 int rb_delete(rb_n **t, rb_n *z);
11664
11665 /**
11666  * tc
11667  * @param a noh raiz da arvore
11668  * @param b elemento a ser deletado
11669  */
11670 +int rb_delete_sentinela(rb_n **t, rb_n *z);
11671 +
11672 /**
11673  * @param a noh raiz da arvore
11674  * @param k chave do elemento a ser excluido
11675  @@ -165,4 +173,8 @@ int rb_sete_impr_elem(void (*i)(void *n));
11676 */
11677 void rb_pr(rb_n *a, int t);
11678
11679 +void rb_convert_null_sentinela(rb_n *t, rb_n *sentinela);
11680 +
11681 +void rb_converte_sentinela_null(rb_n *t);
11682 +
11683 #endif
11684
11685 commit 68d97c470fd681c78bc6ae068187825218bd6d0d
11686 Author: Joilnen <joilnen@gmail.com>
11687 Date: Wed Dec 29 09:30:29 2021 -0300
11688
11689     tentando adaptar sem sentinela mas vou fazer um fork, isso aqui parece
11690     q soh vai com sentinela mesmo
11691
11692 diff --git a/src/red_black.c b/src/red_black.c
11693 index 5c8060e..d53fc09 100644
11694 --- a/src/red_black.c
11695 +++ b/src/red_black.c
11696 @@ -217,6 +217,19 @@ static rb_n* sucessor(rb_n *x)
11697     return y;
11698 }
11699
11700 +static int checa_tmp_node(rb_n *n)
11701 +{
11702 +    return !(n->l || n->r || n->p);
11703 +}
11704

```

```

11707 +
11708 +static void store_tmp_node(rb_n *n, rb_n *a, lr i)
11709 +{
11710 +    n->l = a->l;
11711 +    n->r = a->r;
11712 +    n->p = a->p;
11713 +    n->dclr = i;
11714 +}
11715 +
11716 /**
11717  * deleta noh da arvore red black
11718  * @param t ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao
11719  @@ -226,9 +239,13 @@ static rb_n* sucessor(rb_n *x)
11720  */
11721 int rb_delete(rb_n **t, rb_n *z)
11722 {
11723     rb_n *x = NULL;
11724     rb_n *x = NULL, tmp;
11725     rb_n *y = z;
11726     unsigned char yoc = z->c;
11727     tmp.l = tmp.r = tmp.p = NULL;
11728     tmp.c = 0;
11729 +
11730     printf("TMP ST %d\n", checa_tmp_node(&tmp));
11731
11732     if (!z)
11733         return -1;
11734 @@ -237,11 +254,15 @@ int rb_delete(rb_n **t, rb_n *z)
11735     {
11736         x = z->r;
11737         rb_transplant(t, z, z->r);
11738         if(!x)
11739             store_tmp_node(&tmp, z, RIGHT);
11740     }
11741     else if (!z->r)
11742     {
11743         x = z->l;
11744         rb_transplant(t, z, z->l);
11745         if(!x)
11746             store_tmp_node(&tmp, z, LEFT);
11747     }
11748     else
11749     {
11750 @@ -582,7 +603,7 @@ rb_n *rb_search(rb_n *t, int k)
11751         return NULL;
11752     }
11753 }
11754 -void rb_converte_null_sentinela(rb_n *t, rb_n *sentinela)
11755 +static void rb_converte_null_sentinela(rb_n *t, rb_n *sentinela)
11756 {
11757     if(t)
11758     {
11759 @@ -599,12 +620,11 @@ void rb_converte_null_sentinela(rb_n *t, rb_n *sentinela)
11760         t = sentinela;
11761     }
11762 }
11763 -void rb_converte_sentinela_null(rb_n *t)
11764 +static void rb_converte_sentinela_null(rb_n *t)
11765 {
11766     rb_n *tmp;
11767     if(t)
11768     {
11769         if(t->p->sentinela)
11770         {
11771             tmp = t->p;
11772             diff --git a/src/red_black.h b/src/red_black.h
11773             index a42e6ae..20694f2 100644
11774             --- a/src/red_black.h
11775             +++ b/src/red_black.h
11776             @@ -47,6 +47,7 @@ typedef struct rb_n {
11777             void *d;
11778             unsigned int np;
11779             unsigned char sentinela:1;
11780             unsigned char dclr:1;
11781             struct rb_n *l, *r, *p;
11782         } rb_n;
11783
11784 @@ -173,8 +174,6 @@ int rb_set_impr_elem(void (*i)(void *n));
11785     */
11786     void rb_pr(rb_n *a, int t);
11787
11788 -void rb_convert_null_sentinela(rb_n *t, rb_n *sentinela);
11789 -
11790 -void rb_converte_sentinela_null(rb_n *t);
11791 +typedef enum lr {LEFT, RIGHT} lr;
11792
11793 #endif
11794
11795 commit 64a8e76f859d276f8c23350c60b1b4b16bfbb027
11796 Author: Joilnen <joilnen@gmail.com>
11797 Date: Wed Dec 29 14:05:30 2021 -0300
11798
11799     alterando o noh rb_n para ser tb sentinela
11800
11801 diff --git a/src/red_black.c b/src/red_black.c
11802 index d53fc09..62cfbb6 100644
11803 --- a/src/red_black.c
11804 +++ b/src/red_black.c
11805 @@ -227,7 +227,6 @@ static void store_tmp_node(rb_n *n, rb_n *a, lr i)
11806     n->l = a->l;
11807     n->r = a->r;
11808     n->p = a->p;
11809     n->dclr = i;
11810 }
11811
11812 /**
11813 @@ -603,37 +602,4 @@ rb_n *rb_search(rb_n *t, int k)
11814     return NULL;
11815 }
11816
11817 -static void rb_converte_null_sentinela(rb_n *t, rb_n *sentinela)
11818 -{
11819     if(t)
11820     {
11821
11822

```

```

11823 -         if(!t->p)
11824 -         {
11825 -             t->p = sentinela;
11826 -             t->sentinela = 1;
11827 -         }
11828 -         rb_converte_null_sentinela(t->l, sentinela);
11829 -         rb_converte_null_sentinela(t->r, sentinela);
11830 -     }
11831 -
11832 -     t = sentinela;
11833 - }
11834
11835 -static void rb_converte_sentinela_null(rb_n *t)
11836 -{
11837 -     rb_n *tmp;
11838 -     if(t)
11839 -     {
11840 -         if(t->p->sentinela)
11841 -         {
11842 -             tmp = t->p;
11843 -             t->p = NULL;
11844 -         }
11845 -         rb_converte_sentinela_null(t->l);
11846 -         rb_converte_sentinela_null(t->r);
11847 -     }
11848 -
11849 -     t = NULL;
11850 -     free(tmp);
11851 - }
11852
11853 diff --git a/src/red_black.h b/src/red_black.h
11854 index 20694f2..fc748d7 100644
11855 --- a/src/red_black.h
11856 +++ b/src/red_black.h
11857 @@ -46,8 +46,7 @@ typedef struct rb_n {
11858     unsigned char c:1;
11859     void *d;
11860     unsigned int np;
11861     unsigned char sentinela:1;
11862     unsigned char dclr:1;
11863 +     unsigned char s:1;
11864     struct rb_n *l, *r, *p;
11865 } rb_n;
11866
11867 commit 84ead572bcbbf623785881f78432bcb1af757579
11868 Author: Joilnen <joilnen@gmail.com>
11869 Date: Wed Dec 29 15:15:13 2021 -0300
11870
11871     insercao parcialmente funcionando, problema na rotacao
11872
11873 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
11874 index dc8253d..fca731f 100644
11875 --- a/src/conjunto_ordenado.c
11876 +++ b/src/conjunto_ordenado.c
11877 @@ -48,7 +48,7 @@ int conjord_esvazia(conjord *a)
11878     */
11879     void conjord_imprime_rb(rb_n *t, int s, int e) {
11880         const int dist = 8;
11881         if (!t)
11882 +         if (t == &rb_s)
11883             return;
11884
11885         s += dist;
11886 diff --git a/src/red_black.c b/src/red_black.c
11887 index 62cfbb6..6b49d3a 100644
11888 --- a/src/red_black.c
11889 +++ b/src/red_black.c
11890 @@ -8,6 +8,8 @@
11891 #include "red_black.h"
11892 #include <stdio.h>
11893
11894 +rb_n rb_s;
11895 +
11896 /**
11897  * ponteiro para funcao de impressao do tipo
11898  * dado especificado pelo codigo cliente
11899 @@ -28,7 +30,7 @@ rb_n* rb_cr(int key, void *dado)
11900     /** raiz sempre colorida com preto */
11901     a->c = BLACK;
11902     a->d = dado;
11903     a->p = a->l = a->r = NULL;
11904 +     a->p = a->l = a->r = &rb_s;
11905
11906     return a;
11907 }
11908 @@ -92,7 +94,7 @@ int rb_insert(rb_n **t, rb_n *z)
11909     * itera para chegar no noh folha
11910     * da posicao a ser inserido o novo noh
11911     */
11912     while (x)
11913 +     while (x != &rb_s)
11914     {
11915         y = x;
11916         if (z->k < x->k)
11917 @@ -113,17 +115,17 @@ int rb_insert(rb_n **t, rb_n *z)
11918     * caso n adiciona na esquerda ou diretia de acordo
11919     * com o valor da chave, noh recebe RED como padrao
11920     */
11921     if (!y)
11922 +     if (y == &rb_s)
11923         *t = z;
11924     else if (z->k < y->k)
11925         y->l = z;
11926     else
11927         y->r = z;
11928     z->l = z->r = NULL;
11929 +     z->l = z->r = &rb_s;
11930     z->c = RED;
11931
11932     /** ajuste */
11933     while (z && z->p && z->p->c == RED)
11934 +     while (z->p->c == RED)
11935     {
11936         if (z->p == z->p->p->l)
11937         {
11938 @@ -561,7 +563,7 @@ int rb_sete_impr_elem(void (*i)(void *n))

```



```

11939 void rb_pr(rb_n *a, int t)
11940 {
11941     - if (a)
11942     + if (a != &rb_s)
11943     {
11944         if (a->c == BLACK)
11945         {
11946             diff --git a/src/red_black.h b/src/red_black.h
11947             index fc748d7..bb03965 100644
11948             --- a/src/red_black.h
11949             +++ b/src/red_black.h
11950             @@ -46,10 +46,11 @@ typedef struct rb_n {
11951                 unsigned char c:1;
11952                 void *d;
11953                 unsigned int np;
11954                 unsigned char s:1;
11955                 struct rb_n *l, *r, *p;
11956             } rb_n;
11957
11958             +extern rb_n rb_s;
11959             +
11960             /**
11961              * cria o primeiro noh, a raiz da arvore rb,
11962              */
11963
11964             commit b3d1cc6a37af68e53b11dcf10edf90174b381dde
11965             Author: Joilnen <joilnen@gmail.com>
11966             Date: Wed Dec 29 15:25:28 2021 -0300
11967
11968             insersao parece estar funcionando perfeitamente nos primeiros testes
11969
11970             diff --git a/src/red_black.c b/src/red_black.c
11971             index 6b49d3a..0b68eed 100644
11972             --- a/src/red_black.c
11973             +++ b/src/red_black.c
11974             @@ -130,7 +130,7 @@ int rb_insert(rb_n **t, rb_n *z)
11975             {
11976                 if (z->p == z->p->p->l)
11977                 {
11978                     y = z->p->p->r;
11979                     if (y && y->c == RED)
11980                     + if (y->c == RED)
11981                     {
11982                         z->p->c = BLACK;
11983                         y->c = BLACK;
11984             @@ -152,7 +152,7 @@ int rb_insert(rb_n **t, rb_n *z)
11985                     else
11986                     {
11987                         y = z->p->p->l;
11988                         if (y && y->c == RED)
11989                         + if (y->c == RED)
11990                         {
11991                             z->p->c = BLACK;
11992                             y->c = BLACK;
11993             @@ -522,10 +522,10 @@ int rb_rot_esq(rb_n **t, rb_n *x)
11994             {
11995                 rb_n *y = x->r;
11996                 x->r = y->l;
11997                 - if (y->l)
11998                 + if (y->l != &rb_s)
11999                 y->l->p = x;
12000                 y->p = x->p;
12001                 - if (!x->p)
12002                 + if (x->p == &rb_s)
12003                 *t = y;
12004                 else if (x == x->p->p->l)
12005                 x->p->p->l = y;
12006             @@ -541,10 +541,10 @@ int rb_rot_dir(rb_n **t, rb_n *x)
12007             {
12008                 rb_n *y = x->l;
12009                 x->l = y->r;
12010                 - if (y->r)
12011                 + if (y->r != &rb_s)
12012                 y->r->p = x;
12013                 y->p = x->p;
12014                 - if (!x->p)
12015                 + if (x->p == &rb_s)
12016                 *t = y;
12017                 else if (x == x->p->p->r)
12018                 x->p->p->r = y;
12019
12020             commit 9ea8ca45abd9ae5d9368a30fac3bf96450f4a25a
12021             Author: Joilnen <joilnen@gmail.com>
12022             Date: Wed Dec 29 15:34:51 2021 -0300
12023
12024             delecao funcionando parcialmente, estamos quase lah
12025
12026             diff --git a/src/red_black.c b/src/red_black.c
12027             index 0b68eed..941b75b 100644
12028             --- a/src/red_black.c
12029             +++ b/src/red_black.c
12030             @@ -212,7 +212,7 @@ static rb_n *tree_minimum(rb_n *x)
12031             static rb_n* sucessor(rb_n *x)
12032             {
12033                 rb_n *y = NULL;
12034                 - if (x->r)
12035                 + if (x->r != &rb_s)
12036                 return tree_minimum(x->r);
12037                 while ((y = x->p) && x == y->r)
12038                     x = y;
12039             @@ -240,30 +240,22 @@ static void store_tmp_node(rb_n *n, rb_n *a, lr i)
12040             */
12041             int rb_delete(rb_n **t, rb_n *z)
12042             {
12043                 rb_n *x = NULL, *tmp;
12044                 rb_n *y = NULL;
12045                 rb_n *y = z;
12046                 unsigned char yoc = z->c;
12047                 tmp.l = tmp.r = tmp.p = NULL;
12048                 tmp.c = 0;
12049
12050                 printf("TMP ST %d\n", checa_tmp_node(&tmp));
12051
12052                 if (!z)
12053                     return -1;
12054

```

```

12055 | - if (!z->l)
12056 | + if (z->l == &rb_s)
12057 | {
12058 |     x = z->r;
12059 |     rb_transplant(t, z, z->r);
12060 | - if(!x)
12061 | -     store_tmp_node(&tmp, z, RIGHT);
12062 | }
12063 | - else if (!z->r)
12064 | + else if (z->r == &rb_s)
12065 | {
12066 |     x = z->l;
12067 |     rb_transplant(t, z, z->l);
12068 | - if(!x)
12069 | -     store_tmp_node(&tmp, z, LEFT);
12070 | }
12071 | else
12072 | {
12073 | @@ -288,7 +280,7 @@ int rb_delete(rb_n **t, rb_n *z)
12074 | if (yoc == BLACK)
12075 | {
12076 |     rb_n *w;
12077 | - while (x && x != *t && x->c == BLACK)
12078 | + while (x != *t && x->c == BLACK)
12079 | {
12080 |     if (x == x->p->l)
12081 |     {
12082 |
12083 | commit 8aaabc8098fde34f4f475865244abf2efb496d41
12084 | Author: Joilnen <joilnen@gmail.com>
12085 | Date: Wed Dec 29 15:58:16 2021 -0300
12086 |
12087 | delecao balanceada soh q ainda dando crash pelo o caminho
12088 |
12089 | diff --git a/src/red_black.c b/src/red_black.c
12090 | index 941b75b..0811b52 100644
12091 | --- a/src/red_black.c
12092 | +++ b/src/red_black.c
12093 | @@ -31,6 +31,7 @@ rb_n* rb_cr(int key, void *dado)
12094 |     a->c = BLACK;
12095 |     a->d = dado;
12096 |     a->p = a->l = a->r = &rb_s;
12097 | +     rb_s.c = BLACK;
12098 |
12099 |     return a;
12100 | }
12101 | @@ -182,7 +183,7 @@ int rb_insert(rb_n **t, rb_n *z)
12102 | /*
12103 |  * static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
12104 |  * {
12105 |  *     if (!u->p)
12106 |  * +     if (u->p == &rb_s)
12107 |  *         *t = v;
12108 |  *     else if (u == u->p->l)
12109 |  *         u->p->l = v;
12110 |  * @@ -295,18 +296,15 @@ int rb_delete(rb_n **t, rb_n *z)
12111 |  *
12112 |  *     if (w->l->c == BLACK && w->r->c == BLACK)
12113 |  *     {
12114 |  *         if(!w) w->c = RED;
12115 |  * +         w->c = RED;
12116 |  *         x = x->p;
12117 |  *     }
12118 |  *     else
12119 |  *     {
12120 |  *         if (w->r == BLACK || w->r->c == BLACK)
12121 |  * +         if (w->r->c == BLACK)
12122 |  *         {
12123 |  *             if (w)
12124 |  *             {
12125 |  *                 w->l->c = BLACK;
12126 |  *                 w->c = RED;
12127 |  *             }
12128 |  *             w->l->c = BLACK;
12129 |  * +             w->c = RED;
12130 |  *             rb_rot_dir(t, w);
12131 |  *             w = x->p->r;
12132 |  *         }
12133 |  * @@ -352,7 +350,7 @@ int rb_delete(rb_n **t, rb_n *z)
12134 |  *     }
12135 |  * }
12136 |  * }
12137 |  * if(x) x->c = BLACK;
12138 |  * + x->c = BLACK;
12139 |  * } /* if (yoc == BLACK) */
12140 |
12141 |     return 0;
12142 |
12143 | commit ff44c15422a577e6b3258939e150e312af618cef
12144 | Author: Joilnen <joilnen@gmail.com>
12145 | Date: Wed Dec 29 17:37:59 2021 -0300
12146 |
12147 | ainda crash na delecao da raiz o resto parece ok
12148 |
12149 | diff --git a/src/red_black.c b/src/red_black.c
12150 | index 0811b52..3759161 100644
12151 | --- a/src/red_black.c
12152 | +++ b/src/red_black.c
12153 | @@ -88,7 +88,7 @@ int rb_insert(rb_n **t, rb_n *z)
12154 |     * x eh um ponteiro temporario para iteracao feita
12155 |     * no loop while a seguir para chegar ao noh folha
12156 |     */
12157 | -     rb_n *y = NULL;
12158 | +     rb_n *y = &rb_s;
12159 |     rb_n *x = *t;
12160 |
12161 |     /**
12162 |     @@ -241,13 +241,10 @@ static void store_tmp_node(rb_n *n, rb_n *a, lr i)
12163 |     */
12164 |     int rb_delete(rb_n **t, rb_n *z)
12165 |     {
12166 | -         rb_n *x = NULL;
12167 | +         rb_n *x = &rb_s;
12168 |         rb_n *y = z;
12169 |         unsigned char yoc = z->c;
12170 |

```

```

12171 -     if (!z)
12172 -         return -1;
12173 -
12174 -     if (z->l == &rb_s)
12175 -     {
12176 -         x = z->r;
12177 @@ -265,7 +262,7 @@ int rb_delete(rb_n **t, rb_n *z)
12178         yoc = y->c;
12179         x = y->r;
12180         if (y->p == z)
12181             if(x) x->p = y;
12182 +         else
12183             x->p = y;
12184     {
12185         rb_transplant(t, y, y->r);
12186 @@ -327,7 +324,7 @@ int rb_delete(rb_n **t, rb_n *z)
12187         w = x->p->l;
12188     }
12189 -
12190 -     if (w->r->c == BLACK && w->l->c == BLACK)
12191 +     if (w->l->c == BLACK && w->r->c == BLACK)
12192     {
12193         w->c = RED;
12194         x = x->p;
12195 @@ -580,7 +577,7 @@ rb_n *rb_search(rb_n *t, int k)
12196         rb_n *y = NULL;
12197         rb_n *x = t;
12198 -
12199 -     while (x)
12200 +     while (x != &rb_s)
12201     {
12202         y = x;
12203         if (k < x->k)
12204             commit 452006fcf48e44136678029d86c4a1cad5944a26
12205 Author: Joilnen <joilnen@gmail.com>
12206 Date: Wed Dec 29 18:49:22 2021 -0300
12207
12208         ainda brigando com a delecao da raiz
12209
12210 diff --git a/src/red_black.c b/src/red_black.c
12211 index 3759161..a93e3d1 100644
12212 --- a/src/red_black.c
12213 +++ b/src/red_black.c
12214 @@ -201,7 +201,7 @@ static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
12215 static rb_n *tree_minimum(rb_n *x)
12216 {
12217     rb_n *a = x;
12218 -     while (a->l)
12219 +     while (a->l != &rb_s)
12220     {
12221         a = a->l;
12222     }
12223     return a;
12224 }
12225
12226 commit b0a40b4714920a831b9362d79e5017ae2b10d06d
12227 Author: Joilnen <joilnen@gmail.com>
12228 Date: Wed Dec 29 23:44:40 2021 -0300
12229
12230 delecao quase lah
12231
12232 diff --git a/src/red_black.c b/src/red_black.c
12233 index a93e3d1..d9f7152 100644
12234 --- a/src/red_black.c
12235 +++ b/src/red_black.c
12236 @@ -277,7 +277,7 @@ int rb_delete(rb_n **t, rb_n *z)
12237     if (yoc == BLACK)
12238     {
12239         rb_n *w;
12240         rb_n *w = &rb_s;
12241         while (x != *t && x->c == BLACK)
12242         {
12243             if (x == x->p->l)
12244 @@ -320,7 +320,7 @@ int rb_delete(rb_n **t, rb_n *z)
12245         {
12246             w->c = BLACK;
12247             x->p->c = RED;
12248             rb_rot_esq(t, x->p);
12249 +             rb_rot_dir(t, x->p);
12250             w = x->p->l;
12251         }
12252     }
12253 @@ -335,14 +335,14 @@ int rb_delete(rb_n **t, rb_n *z)
12254     {
12255         w->r->c = BLACK;
12256         w->c = RED;
12257         rb_rot_dir(t, w);
12258 +         rb_rot_esq(t, w);
12259         w = x->p->l;
12260     }
12261
12262     w->c = x->p->c;
12263     x->p->c = BLACK;
12264     w->l->c = BLACK;
12265 -     rb_rot_esq(t, x->p);
12266 +     rb_rot_dir(t, x->p);
12267     x = *t;
12268 }
12269 }
12270
12271 commit 32b6a153983658cc85d3c17622e41c4526cd2441
12272 Author: Joilnen <joilnen@gmail.com>
12273 Date: Wed Dec 29 23:51:33 2021 -0300
12274
12275 red black funcionando insercao e delecao
12276
12277 diff --git a/src/main.c b/src/main.c
12278 index 1ec8a5a..dac212b 100644
12279 --- a/src/main.c
12280 +++ b/src/main.c
12281 @@ -264,6 +264,17 @@ int main()
12282     printf("tecle enter\n");
12283     getc(stdin);
12284
12285 +     rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
12286 +     LTELA;

```

```

12287 +     printf("\n");
12288 +     rb_pr(conjunto_a, TCOLOR);
12289 +     printf("\n");
12290 +     printf("= arvore =====\n");
12291 +     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12292 +     printf("= =====\n");
12293 +     printf("** tecle enter *\n");
12294 +     getc(stdin);
12295 +
12296 +     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
12297 +     LTELA;
12298 +     printf("\n nada\n");
12299
12300 commit 1c3c7ea4e1cbf507294591afe704724095bb010f
12301 Author: Joilnen <joilnen@gmail.com>
12302 Date:   Wed Dec 29 23:58:12 2021 -0300
12303
12304     apenas formatacao
12305
12306 diff --git a/src/main.c b/src/main.c
12307 index dac212b..5d9c197 100644
12308 --- a/src/main.c
12309 +++ b/src/main.c
12310 @@ -255,7 +255,6 @@ int main()
12311
12312     rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
12313     LTELA;
12314 -     printf("\n");
12315     rb_pr(conjunto_a, TCOLOR);
12316     printf("\n");
12317     printf("= arvore =====\n");
12318 @@ -266,7 +265,6 @@ int main()
12319
12320     rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
12321     LTELA;
12322 -     printf("\n");
12323     rb_pr(conjunto_a, TCOLOR);
12324     printf("\n");
12325     printf("= arvore =====\n");
12326 @@ -277,7 +275,7 @@ int main()
12327
12328     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
12329     LTELA;
12330 -     printf("\n nada\n");
12331 +     printf("nada\n");
12332     printf("= arvore =====\n");
12333     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12334     printf("= =====\n");
12335
12336 commit c7934628e33c6d959bcb20c78dbc3705ce2be729
12337 Author: Joilnen <joilnen@gmail.com>
12338 Date:   Thu Dec 30 00:30:14 2021 -0300
12339
12340     limpando o codigo
12341
12342 diff --git a/src/red_black.c b/src/red_black.c
12343 index d9f7152..aba04c7 100644
12344 --- a/src/red_black.c
12345 +++ b/src/red_black.c
12346 @@ -8,6 +8,10 @@
12347 #include "red_black.h"
12348 #include <stdio.h>
12349
12350 /**
12351  * definicao do noh sentinela
12352  * @param rb_s red black noh setinela
12353  */
12354 rb_n rb_s;
12355
12356 /**
12357 @@ -220,18 +224,6 @@ static rb_n* sucessor(rb_n *x)
12358     return y;
12359 }
12360
12361 -static int checa_tmp_node(rb_n *n)
12362 -{
12363     return !(n->l || n->r || n->p);
12364 -}
12365 -
12366 -static void store_tmp_node(rb_n *n, rb_n *a, lr i)
12367 -{
12368     n->l = a->l;
12369     n->r = a->r;
12370     n->p = a->p;
12371 -}
12372 -
12373 /**
12374  * deleta noh da arvore red black
12375  * @param t ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao
12376 @@ -353,133 +345,6 @@ int rb_delete(rb_n **t, rb_n *z)
12377     return 0;
12378 }
12379
12380 -/**
12381  * deleta noh da arvore red black
12382  * @param t ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao
12383  * e manter a alteracao depois do retorno da funcao
12384  * @param z ponteiro para o noh a ser deletado, para deletar por chave o TAD deve
12385  * passar a saida da funcao pesquisar, num arranjo funcional na chamada
12386  */
12387 -int rb_delete_sentinela(rb_n **t, rb_n *z)
12388 -{
12389     rb_n *x = NULL;
12390     rb_n *y = z;
12391     unsigned char yoc = z->c;
12392
12393     if (!z)
12394         return -1;
12395
12396     if (!z->l)
12397     {
12398         x = z->r;
12399         rb_transplant(t, z, z->r);
12400     }
12401     else if (!z->r)
12402     {

```

```

12403 |     x = z->l;
12404 |     rb_transplant(t, z, z->l);
12405 | }
12406 | else
12407 | {
12408 |     // y = tree_minimum(z->r);
12409 |     y = successor(z);
12410 |     yoc = y->c;
12411 |     x = y->r;
12412 |     if (y->p == z)
12413 |         if (x) x->p = y;
12414 |     else
12415 |     {
12416 |         rb_transplant(t, y, y->r);
12417 |         y->r = z->r;
12418 |         y->r->p = y;
12419 |     }
12420 |     rb_transplant(t, z, y);
12421 |     y->l = z->l;
12422 |     y->l->p = y;
12423 |     y->c = z->c;
12424 | }
12425 |
12426 | if (yoc == BLACK)
12427 | {
12428 |     rb_n *w;
12429 |     while (x && x != *t && x->c == BLACK)
12430 |     {
12431 |         if (x == x->p->l)
12432 |         {
12433 |             w = x->p->r;
12434 |             if (w->c == RED)
12435 |             {
12436 |                 w->c = BLACK;
12437 |                 x->p->c = RED;
12438 |                 rb_rot_esq(t, x->p);
12439 |                 w = x->p->r;
12440 |             }
12441 |
12442 |             if (w->l->c == BLACK && w->r->c == BLACK)
12443 |             {
12444 |                 if (!w) w->c = RED;
12445 |                 x = x->p;
12446 |             }
12447 |             else
12448 |             {
12449 |                 if (w->r == BLACK || w->r->c == BLACK)
12450 |                 {
12451 |                     if (w)
12452 |                     {
12453 |                         w->l->c = BLACK;
12454 |                         w->c = RED;
12455 |                     }
12456 |                     rb_rot_dir(t, w);
12457 |                     w = x->p->r;
12458 |                 }
12459 |
12460 |                 w->c = x->p->c;
12461 |                 x->p->c = BLACK;
12462 |                 w->r->c = BLACK;
12463 |                 rb_rot_esq(t, x->p);
12464 |                 x = *t;
12465 |             }
12466 |         }
12467 |         else
12468 |         {
12469 |             w = x->p->l;
12470 |             if (w->c == RED)
12471 |             {
12472 |                 w->c = BLACK;
12473 |                 x->p->c = RED;
12474 |                 rb_rot_esq(t, x->p);
12475 |                 w = x->p->l;
12476 |             }
12477 |
12478 |             if (w->r->c == BLACK && w->l->c == BLACK)
12479 |             {
12480 |                 w->c = RED;
12481 |                 x = x->p;
12482 |             }
12483 |             else
12484 |             {
12485 |                 if (w->l->c == BLACK)
12486 |                 {
12487 |                     w->r->c = BLACK;
12488 |                     w->c = RED;
12489 |                     rb_rot_dir(t, w);
12490 |                     w = x->p->l;
12491 |                 }
12492 |
12493 |                 w->c = x->p->c;
12494 |                 x->p->c = BLACK;
12495 |                 w->l->c = BLACK;
12496 |                 rb_rot_esq(t, x->p);
12497 |                 x = *t;
12498 |             }
12499 |         }
12500 |     }
12501 |     if (x) x->c = BLACK;
12502 | } /* if (yoc == BLACK) */
12503 |
12504 | return 0;
12505 | }
12506 |
12507 | int rb_exclui_elem(rb_n *a, int k)
12508 | {
12509 |     return 0;
12510 | }
12511 | diff --git a/src/red_black.h b/src/red_black.h
12512 | index bb03965..111fd31 100644
12513 | --- a/src/red_black.h
12514 | +++ b/src/red_black.h
12515 | @@ -49,6 +49,9 @@ typedef struct rb_n {
12516 |     struct rb_n *l, *r, *p;
12517 | } rb_n;
12518 | +/**

```

```

12519 + * noh sentinela
12520 + */
12521 extern rb_n rb_s;
12522
12523 /**
12524 @@ -90,13 +93,6 @@ rb_n *rb_search(rb_n *t, int k);
12525 */
12526 int rb_delete(rb_n **t, rb_n *z);
12527
12528 /**
12529 - * tc
12530 - * @param a noh raiz da arvore
12531 - * @param b elemento a ser deletado
12532 - */
12533 -int rb_delete_sentinela(rb_n **t, rb_n *z);
12534 -
12535 /**
12536 - * @param a noh raiz da arvore
12537 - * @param k chave do elemento a ser excluido
12538 @@ -174,6 +170,4 @@ int rb_set impr_elem(void (*i)(void *n));
12539 */
12540 void rb_pr(rb_n *a, int t);
12541
12542 -typedef enum lr {LEFT, RIGHT} lr;
12543 -
12544 #endif
12545
12546 commit 398f6159236bfc1afdc05486676f845aec2cd0b
12547 Author: Joilnen <joilnen@gmail.com>
12548 Date: Thu Dec 30 10:02:06 2021 -0300
12549
12550 alterando o codigo de limpar a tela para funcionar em terminais vt100 com
12551 o minimo de recursos
12552
12553 diff --git a/src/jcurses.h b/src/jcurses.h
12554 index 5d082e2..3a8e187 100644
12555 --- a/src/jcurses.h
12556 +++ b/src/jcurses.h
12557 @@ -10,7 +10,7 @@
12558 - * primaria que atende as necessidades apenas dessa avaliacao
12559 - */
12560
12561 -#define LIMPA_TELA "\033c"
12562 +#define LIMPA_TELA "\033[H\033[J"
12563 #define S_AZUL "\033[44m\033[37m"
12564 #define S_VERM "\033[41m\033[37m"
12565 #define S_INV "\033[7m"
12566
12567 commit eaf427d19d12261327af53fcd63a5b77cfcf0825
12568 Author: Joilnen <joilnen@gmail.com>
12569 Date: Thu Dec 30 10:30:07 2021 -0300
12570
12571 movendo o codigo de teste para um arquivo e funcao separados do main
12572
12573 diff --git a/src/main.c b/src/main.c
12574 index 5d9c197..bbf1231 100644
12575 --- a/src/main.c
12576 +++ b/src/main.c
12577 @@ -1,300 +1,12 @@
12578 /**
12579 - * aluno: joilnen leite
12580 - */
12581 +#include "testa_rb.h"
12582
12583 /**
12584 - * inclusao da biblioteca implementada baseada em arvores red black
12585 - */
12586 -#include "conjunto_ordenado.h"
12587 -#include <stdio.h>
12588 -#include <string.h>
12589 -
12590 /**
12591 - * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
12592 - * @noh rb
12593 - */
12594 -void imprime_meu_tipo(void *);
12595 -
12596 /**
12597 - * funcao main de teste separada do codigo da biblioteca como especificado
12598 - */
12599 int main()
12600 {
12601 - /**
12602 - * definicao de um tipo exemplo pelo o usuario
12603 - */
12604 - typedef struct {
12605 - char *nome;
12606 - char *telefone;
12607 - } meu_tipo;
12608 -
12609 - /**
12610 - * criacao do dado para o tipo do usuario
12611 - */
12612 - meu_tipo mt0;
12613 - mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
12614 - strncpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
12615 - mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
12616 - strncpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
12617 -
12618 - /**
12619 - * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
12620 - */
12621 - rb_set impr_elem(imprime_meu_tipo);
12622 -
12623 - /**
12624 - * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
12625 - * uma interface mais intuitiva ao usuario da biblioteca
12626 - * @parametro 0 nome variavel do noh raiz
12627 - * @parametro 1 chave do noh raiz
12628 - * @parametro 2 dado criado neste noh, pode ser de qqr tipo
12629 - */
12630 - rb_n *conjunto_a = rb_cr(10, &mt0);
12631 -
12632 - LTELTA;
12633 - rb_pr(conjunto_a, TCOLOR);
12634 - printf("\n");

```

```

12635 - printf("= arvore =====\n");
12636 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12637 - printf("= =====\n");
12638 - printf("* tecla enter *\n");
12639 - getc(stdin);
12640 -
12641 - meu_tipo mt1;
12642 - mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
12643 - strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
12644 - mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
12645 - strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
12646 -
12647 - rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
12648 - novo_noh->k = 9;
12649 - novo_noh->d = &mt1;
12650 -
12651 - rb_insert(&conjunto_a, novo_noh);
12652 -
12653 - LTELA;
12654 - rb_pr(conjunto_a, TCOLOR);
12655 - printf("\n");
12656 - printf("= arvore =====\n");
12657 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12658 - printf("= =====\n");
12659 - printf("* tecla enter *\n");
12660 - getc(stdin);
12661 -
12662 - meu_tipo mt2;
12663 - mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
12664 - strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
12665 - mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
12666 - strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
12667 -
12668 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
12669 - novo_noh->k = 11;
12670 - novo_noh->d = &mt2;
12671 - rb_insert(&conjunto_a, novo_noh);
12672 -
12673 - LTELA;
12674 - rb_pr(conjunto_a, TCOLOR);
12675 - printf("\n");
12676 - printf("= arvore =====\n");
12677 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12678 - printf("= =====\n");
12679 - printf("* tecla enter *\n");
12680 - getc(stdin);
12681 -
12682 - meu_tipo mt3;
12683 - mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
12684 - strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
12685 - mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
12686 - strncpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
12687 -
12688 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
12689 - novo_noh->k = 8;
12690 - novo_noh->d = &mt3;
12691 - rb_insert(&conjunto_a, novo_noh);
12692 -
12693 - LTELA;
12694 - rb_pr(conjunto_a, TCOLOR);
12695 - printf("\n");
12696 - printf("= arvore =====\n");
12697 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12698 - printf("= =====\n");
12699 - printf("* tecla enter *\n");
12700 - getc(stdin);
12701 -
12702 - meu_tipo mt4;
12703 - mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
12704 - strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
12705 - mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
12706 - strncpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
12707 -
12708 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
12709 - novo_noh->k = 6;
12710 - novo_noh->d = &mt4;
12711 - rb_insert(&conjunto_a, novo_noh);
12712 + testa_arvore_rb();
12713 -
12714 - LTELA;
12715 - rb_pr(conjunto_a, TCOLOR);
12716 - printf("\n");
12717 - printf("= arvore =====\n");
12718 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12719 - printf("= =====\n");
12720 - printf("* tecla enter *\n");
12721 - getc(stdin);
12722 -
12723 - meu_tipo mt5;
12724 - mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
12725 - strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
12726 - mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
12727 - strncpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
12728 -
12729 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
12730 - novo_noh->k = 5;
12731 - novo_noh->d = &mt5;
12732 - rb_insert(&conjunto_a, novo_noh);
12733 -
12734 - LTELA;
12735 - rb_pr(conjunto_a, TCOLOR);
12736 - printf("\n");
12737 - printf("= arvore =====\n");
12738 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12739 - printf("= =====\n");
12740 - printf("* tecla enter *\n");
12741 -
12742 - meu_tipo mt6;
12743 - mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
12744 - strncpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
12745 - mt6.telefone = (char *) malloc(sizeof("999999999") + 1);
12746 - strncpy(mt6.telefone, "999999999", sizeof("999999999") + 1);
12747 -
12748 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
12749 - novo_noh->k = 4;
12750 - novo_noh->d = &mt6;

```

```

12751 - rb_insert(&conjunto_a, novo_noh);
12752 -
12753 - LTELA;
12754 - rb_pr(conjunto_a, TCOLOR);
12755 - printf("\n");
12756 - printf("= arvore =====\n");
12757 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12758 - printf("= =====\n");
12759 - printf("* tecla enter *\n");
12760 - getc(stdin);
12761 -
12762 - meu_tipo mt7;
12763 - mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
12764 - strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
12765 - mt7.telefone = (char *) malloc(sizeof("999999999") + 1);
12766 - strncpy(mt7.telefone, "999999999", sizeof("999999999") + 1);
12767 -
12768 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
12769 - novo_noh->k = 3;
12770 - novo_noh->d = &mt7;
12771 - rb_insert(&conjunto_a, novo_noh);
12772 -
12773 - LTELA;
12774 - rb_pr(conjunto_a, TCOLOR);
12775 - printf("\n");
12776 - printf("= arvore =====\n");
12777 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12778 - printf("= =====\n");
12779 - printf("* tecla enter *\n");
12780 - getc(stdin);
12781 -
12782 - rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
12783 - printf("\n");
12784 - LTELA;
12785 - rb_pr(conjunto_a, TCOLOR);
12786 - printf("\n");
12787 - printf("= arvore =====\n");
12788 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12789 - printf("= =====\n");
12790 - printf("* tecla enter *\n");
12791 - getc(stdin);
12792 -
12793 - rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
12794 - LTELA;
12795 - rb_pr(conjunto_a, TCOLOR);
12796 - printf("\n");
12797 - printf("= arvore =====\n");
12798 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12799 - printf("= =====\n");
12800 - printf("* tecla enter *\n");
12801 - getc(stdin);
12802 -
12803 - rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
12804 - LTELA;
12805 - rb_pr(conjunto_a, TCOLOR);
12806 - printf("\n");
12807 - printf("= arvore =====\n");
12808 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12809 - printf("= =====\n");
12810 - printf("* tecla enter *\n");
12811 - getc(stdin);
12812 -
12813 - rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
12814 - LTELA;
12815 - rb_pr(conjunto_a, TCOLOR);
12816 - printf("\n");
12817 - printf("= arvore =====\n");
12818 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12819 - printf("= =====\n");
12820 - printf("* tecla enter *\n");
12821 - getc(stdin);
12822 -
12823 - rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
12824 - LTELA;
12825 - rb_pr(conjunto_a, TCOLOR);
12826 - printf("\n");
12827 - printf("= arvore =====\n");
12828 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12829 - printf("= =====\n");
12830 - printf("* tecla enter *\n");
12831 - getc(stdin);
12832 - rb_pr(conjunto_a, TCOLOR);
12833 - printf("\n");
12834 -
12835 - rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
12836 - LTELA;
12837 - rb_pr(conjunto_a, TCOLOR);
12838 - printf("\n");
12839 - printf("= arvore =====\n");
12840 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12841 - printf("= =====\n");
12842 - printf("* tecla enter *\n");
12843 - getc(stdin);
12844 -
12845 - rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
12846 - LTELA;
12847 - rb_pr(conjunto_a, TCOLOR);
12848 - printf("\n");
12849 - printf("= arvore =====\n");
12850 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12851 - printf("= =====\n");
12852 - printf("* tecla enter *\n");
12853 - getc(stdin);
12854 -
12855 - rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
12856 - LTELA;
12857 - printf("nada\n");
12858 - printf("= arvore =====\n");
12859 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12860 - printf("= =====\n");
12861 - rb_pr(conjunto_a, TCOLOR);
12862 - printf("\n");
12863 - /**
12864 -  * testando apenas rb no momento
12865 -  * conjord *conj_a = conjord_cria();
12866 -  * printf("eh vaziao %d", conjord_eh_vazio(conj_a));

```



```

12867 | - */
12868 | -}
12869 | -
12870 | -/**
12871 | - * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
12872 | - * exemplo
12873 | - */
12874 | -void imprime_meu_tipo(void *n)
12875 | -{
12876 | +     return 0;
12877 | }
12878 | -
12879 | -
12880 | -
12881 | diff --git a/src/makefile b/src/makefile
12882 | index d201161..944ed46 100644
12883 | --- a/src/makefile
12884 | +++ b/src/makefile
12885 | @@ -3,8 +3,8 @@
12886 | #
12887 |
12888 | C=gcc
12889 | -S=main.c conjunto_ordenado.c red_black.c
12890 | -O=main.o conjunto_ordenado.o red_black.o
12891 | +S=main.c conjunto_ordenado.c red_black.c testa_rb.c
12892 | +O=main.o conjunto_ordenado.o red_black.o testa_rb.o
12893 | E=rodeme
12894 | D=-g
12895 |
12896 | diff --git a/src/test_rb.c b/src/testa_rb.c
12897 | similarity index 96%
12898 | rename from src/test_rb.c
12899 | rename to src/testa_rb.c
12900 | index 5643809..1a7a74e 100644
12901 | --- a/src/test_rb.c
12902 | +++ b/src/testa_rb.c
12903 | @@ -18,7 +18,7 @@ void imprime_meu_tipo(void *);
12904 | /**
12905 |  * funcao main de teste separada do codigo da biblioteca como especificado
12906 |  */
12907 | -int main()
12908 | +int testa_arvore_rb()
12909 | {
12910 |     /**
12911 |     * definicao de um tipo exemplo pelo o usuario
12912 |     @@ -252,9 +252,18 @@ int main()
12913 |         getc(stdin);
12914 |         rb_pr(conjunto_a, TCOLOR);
12915 |         printf("\n");
12916 |
12917 |         rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
12918 |         LTELA;
12919 |         rb_pr(conjunto_a, TCOLOR);
12920 |         printf("\n");
12921 |         printf("= arvore =====\n");
12922 |         conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12923 |         printf("= =====\n");
12924 |         printf("= * tecla enter *\n");
12925 |         getc(stdin);
12926 |
12927 |         rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
12928 |         LTELA;
12929 |         rb_pr(conjunto_a, TCOLOR);
12930 |         printf("\n");
12931 |     @@ -263,9 +272,10 @@ int main()
12932 |         printf("= =====\n");
12933 |         printf("= * tecla enter *\n");
12934 |         getc(stdin);
12935 |
12936 |         rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
12937 |         LTELA;
12938 |         printf("\n nada\n");
12939 |         printf("nada\n");
12940 |         printf("= arvore =====\n");
12941 |         conjord_imprime_rb(conjunto_a, 1, TCOLOR);
12942 |         printf("= =====\n");
12943 | diff --git a/src/testa_rb.h b/src/testa_rb.h
12944 | new file mode 100644
12945 | index 0000000..30e4996
12946 | --- /dev/null
12947 | +++ b/src/testa_rb.h
12948 | @@ -0,0 +1,16 @@
12949 | +#ifndef TESTA_ARVORE_RB_
12950 | +#define TESTA_ARVORE_RB_
12951 | +
12952 | +/**
12953 | + * aluno: joilnen leite
12954 | + */
12955 | +
12956 | +/**
12957 | + * declaracao da funcao de teste da
12958 | + * arvore vermelha e preta
12959 | + */
12960 | +void testa_arvore_rb();
12961 | +
12962 | +#endif
12963 | +
12964 | +
12965 |
12966 | commit 8aa14fa10cf9a1d58625af816d910fcce337f228
12967 | Author: Joilnen <joilnen@gmail.com>
12968 | Date: Thu Dec 30 10:50:34 2021 -0300
12969 |
12970 |     funcao insere noh retorna erro se ja existe chave igual, foi apagado
12971 |     codigo de uma funcao insere antiga que funcionava mas sem sentinela
12972 |     q eh como estamos usando agora
12973 |
12974 | diff --git a/src/red_black.c b/src/red_black.c
12975 | index aba04c7..54afdad 100644
12976 | --- a/src/red_black.c
12977 | +++ b/src/red_black.c
12978 | @@ -40,48 +40,6 @@ rb_n* rb_cr(int key, void *dado)
12979 |         return a;
12980 |     }
12981 |
12982 | -/**

```

```

12983 - * @param a noh raiz
12984 - * @param k chave escolhida
12985 - * @param e dado especificado pelo o codigo cliente
12986 - */
12987 -int rb_ins_elem(rb_n *a, int k, void *e)
12988 -{
12989     rb_n *it = a;
12990     printf("k %d it->k %d\n", k, it->k);
12991
12992     while (it)
12993     {
12994         rb_n *tmp = it;
12995         int rl = 0;
12996         if (k < it->k)
12997             it = it->l, rl = !0;
12998         else if (k > it->k)
12999             it = it->r, rl = !0;
13000         else return -1;
13001
13002         printf("rl %d\n", rl);
13003
13004         if (!it)
13005         {
13006             it = (rb_n *) malloc(sizeof(rb_n));
13007             it->d = e;
13008             it->k = k;
13009             /** cor vermelha por default na insercao */
13010             it->c = 1;
13011             it->l = it->r = NULL;
13012
13013             if (rl & 0)
13014                 tmp->l = it;
13015             else
13016                 tmp->r = it;
13017             return 0;
13018         }
13019     }
13020
13021     return -1;
13022 -}
13023
13024 /**
13025  * inclusao baseado no livro do cormen
13026  */
13027 @@ -104,8 +62,10 @@ int rb_insert(rb_n **t, rb_n *z)
13028     y = x;
13029     if (z->k < x->k)
13030         x = x->l;
13031     else if (z->k > x->k)
13032         x = x->r;
13033     else
13034         return -1;
13035 }
13036
13037 /**
13038 @@ -178,6 +138,8 @@ int rb_insert(rb_n **t, rb_n *z)
13039     }
13040     (t)->c = BLACK;
13041 +
13042 +     return 0;
13043 }
13044
13045 /**
13046 diff --git a/src/red_black.h b/src/red_black.h
13047 index 111fd31..6b6c1ce 100644
13048 --- a/src/red_black.h
13049 +++ b/src/red_black.h
13050 @@ -65,13 +65,6 @@ rb_n* rb_cr(int key, void *dado);
13051 - */
13052 -unsigned int rb_nelem(rb_n *a);
13053
13054 -/**
13055 - * @param a noh raiz da arvore
13056 - * @param k chave do elemento a ser inserido
13057 - * @param e tipo generico de dado a ser guardado neste noh
13058 - */
13059 -int rb_ins_elem(rb_n *a, int k, void *e);
13060
13061 /**
13062  * tc
13063  * @param a noh raiz da arvore
13064 diff --git a/src/testa_rb.c b/src/testa_rb.c
13065 index 1a7a74e..09dbc3e 100644
13066 --- a/src/testa_rb.c
13067 +++ b/src/testa_rb.c
13068 @@ -60,13 +60,27 @@ int testa_arvore_rb()
13069     printf("tecle enter *\n");
13070     getc(stdin);
13071
13072     meu_tipo mt8;
13073     mt8.nome = (char *) malloc(sizeof("nome para teste") + 1);
13074     strncpy(mt8.nome, "nome para teste", sizeof("nome para teste") + 1);
13075     mt8.telefone = (char *) malloc(sizeof("999999999") + 1);
13076     strncpy(mt8.telefone, "999999999", sizeof("999999999") + 1);
13077
13078     rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
13079     novo_noh->k = 9;
13080     novo_noh->d = &mt8;
13081
13082     rb_insert(&conjunto_a, novo_noh);
13083
13084     meu_tipo mt1;
13085     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
13086     strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
13087     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
13088     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
13089
13090     rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
13091     novo_noh = (rb_n *) malloc (sizeof(rb_n));
13092     novo_noh->k = 9;
13093     novo_noh->d = &mt1;
13094

```

```

13099 |
13100 | commit 94a049df392c385e8e97dc8530f8a66268f16609
13101 | Author: Joilnen <joilnen@gmail.com>
13102 | Date: Thu Dec 30 13:02:44 2021 -0300
13103 |
13104 | reestruturando main e funcoes auxiliares para chamar testes sobre o TAD
13105 |
13106 | diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
13107 | index 3123fbb..9ea7863 100644
13108 | --- a/src/conjunto_ordenado.h
13109 | +++ b/src/conjunto_ordenado.h
13110 | @@ -10,7 +10,7 @@
13111 |  * suas funcoes e funcionamento interno do cliente
13112 |  * expando apenas funcoes e operacoes com semantica
13113 |  * relativa ao uso do cliente aqui nesta avaliacao
13114 |  * conjunto ordenado as operacoes e funcoes
13115 |  * conjunto ordenado as operacoes e funcoes
13116 |  * exigidas no documento da avaliacao para este TAD
13117 |  */
13118 | typedef struct conjord {
13119 | diff --git a/src/main.c b/src/main.c
13120 | index bbf1231..70e64d5 100644
13121 | --- a/src/main.c
13122 | +++ b/src/main.c
13123 | @@ -1,12 +1,77 @@
13124 | /**
13125 |  * aluno: joilnen leite
13126 |  */
13127 | #include <stdio.h>
13128 | #include "testa_item_1.h"
13129 | #include "testa_item_2.h"
13130 | #include "testa_item_3.h"
13131 | #include "testa_rb.h"
13132 | #include "jurses.h"
13133 | +
13134 | +/**
13135 | + * estas funcoes mostram testes referentes
13136 | + * aos itens da avaliacao
13137 | + */
13138 | +static void tela_item_1();
13139 | +static void tela_item_2();
13140 | +static void tela_item_3();
13141 | +static void tela_testa_rb();
13142 | +
13143 | int main()
13144 | {
13145 |     testa_arvore_rb();
13146 |     char op, nitems = 4;
13147 | +
13148 | +    /**
13149 | +     * aqui poderia ser escrito
13150 | +     * "case 0:
13151 | +     *     tela_item_1();
13152 | +     *     getc(stdin);
13153 | +     *     testa_item_1();
13154 | +     *     break;
13155 | +     * "
13156 | +     * em um unico swtich mas assim ficou mais legivel
13157 | +     * e limpo
13158 | +     */
13159 | +    for (op = 0; op < nitems; ++op)
13160 | +    {
13161 | +        switch (op)
13162 | +        {
13163 | +            case 0: tela_item_1(); break;
13164 | +            case 1: tela_item_2(); break;
13165 | +            case 2: tela_item_3(); break;
13166 | +            case 3: tela_testa_rb();
13167 | +
13168 | +            getc(stdin);
13169 | +            switch (op)
13170 | +            {
13171 | +                case 0: testa_item_1(); break;
13172 | +                case 1: testa_item_1(); break;
13173 | +                case 2: testa_item_1(); break;
13174 | +                case 3: testa_arvore_rb();
13175 | +            }
13176 | +        }
13177 | +
13178 | +        return 0;
13179 | +}
13180 | +
13181 | +void tela_item_1()
13182 | +{
13183 | +    printf("Testa item 1\n");
13184 | +}
13185 | +
13186 | +void tela_item_2()
13187 | +{
13188 | +    printf("Testa item 2\n");
13189 | +}
13190 | +
13191 | +void tela_item_3()
13192 | +{
13193 | +    printf("Testa item 3\n");
13194 | +}
13195 | +
13196 | +void tela_testa_rb()
13197 | +{
13198 | +    printf("Testa rb\n");
13199 | +}
13200 | +
13201 | +
13202 | diff --git a/src/makefile b/src/makefile
13203 | index 944ed46..94af2b5 100644
13204 | --- a/src/makefile
13205 | +++ b/src/makefile
13206 | @@ -3,8 +3,10 @@
13207 | #
13208 |
13209 | C=gcc
13210 | -S=main.c conjunto_ordenado.c red_black.c testa_rb.c
13211 | -O=main.o conjunto_ordenado.o red_black.o testa_rb.o
13212 | +S=main.c conjunto_ordenado.c red_black.c testa_rb.c \
13213 | +    testa_item_1.c testa_item_2.c testa_item_3.c
13214 | +O=main.o conjunto_ordenado.o red_black.o testa_rb.o \

```

```

13215 + testa_item_1.o testa_item_2.o testa_item_3.o
13216 E=rodeme
13217 D=-g
13218
13219 @@ -15,3 +17,4 @@ all: $(0)
13220 clean:
13221 rm -f *.o $(E)
13222
13223 +
13224 diff --git a/src/red_black.c b/src/red_black.c
13225 index 54afdad..96a1c53 100644
13226 --- a/src/red_black.c
13227 +++ b/src/red_black.c
13228 @@ -2,7 +2,7 @@
13229 * aluno: joilnen leite
13230 * arquivo separado do main, como especificado
13231 * de implementacao das funcoes da redblack
13232 - * utilizada no conjunto ordenado
13233 + * utilizada como base no TAD conjunto ordenado
13234 */
13235 #include "jurses.h"
13236 #include "red_black.h"
13237 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
13238 new file mode 100644
13239 index 0000000..5fab149
13240 --- /dev/null
13241 +++ b/src/testa_item_1.c
13242 @@ -0,0 +1,8 @@
13243 #include "testa_item_1.h"
13244 +
13245 +void testa_item_1()
13246 +{
13247 +
13248 +}
13249 +
13250 +
13251 diff --git a/src/testa_item_1.h b/src/testa_item_1.h
13252 new file mode 100644
13253 index 0000000..38d9cb8
13254 --- /dev/null
13255 +++ b/src/testa_item_1.h
13256 @@ -0,0 +1,7 @@
13257 #ifndef TESTA_ITEM_1_H_
13258 #define TESTA_ITEM_1_H_
13259 +
13260 +void testa_item_1();
13261 +
13262 #endif
13263 +
13264 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
13265 new file mode 100644
13266 index 0000000..56c9478
13267 --- /dev/null
13268 +++ b/src/testa_item_2.c
13269 @@ -0,0 +1,7 @@
13270 #include "testa_item_2.h"
13271 +
13272 +void testa_item_2()
13273 +{
13274 +
13275 +}
13276 +
13277 diff --git a/src/testa_item_2.h b/src/testa_item_2.h
13278 new file mode 100644
13279 index 0000000..8f0417b
13280 --- /dev/null
13281 +++ b/src/testa_item_2.h
13282 @@ -0,0 +1,7 @@
13283 #ifndef TESTA_ITEM_2_H_
13284 #define TESTA_ITEM_2_H_
13285 +
13286 +void testa_item_2();
13287 +
13288 #endif
13289 +
13290 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
13291 new file mode 100644
13292 index 0000000..c68e970
13293 --- /dev/null
13294 +++ b/src/testa_item_3.c
13295 @@ -0,0 +1,7 @@
13296 #include "testa_item_3.h"
13297 +
13298 +void testa_item_3()
13299 +{
13300 +
13301 +}
13302 +
13303 diff --git a/src/testa_item_3.h b/src/testa_item_3.h
13304 new file mode 100644
13305 index 0000000..db34708
13306 --- /dev/null
13307 +++ b/src/testa_item_3.h
13308 @@ -0,0 +1,7 @@
13309 #ifndef TESTA_ITEM_3_H_
13310 #define TESTA_ITEM_3_H_
13311 +
13312 +void testa_item_3();
13313 +
13314 #endif
13315 +
13316 +
13317 commit e3e7e67ff083119e7ba2aa7840cd92f5e2e3508e
13318 Author: Joilnen <joilnen@gmail.com>
13319 Date: Thu Dec 30 13:38:39 2021 -0300
13320
13321     organizando codigo de testes
13322
13323 diff --git a/src/main.c b/src/main.c
13324 index 70e64d5..87e586b 100644
13325 --- a/src/main.c
13326 +++ b/src/main.c
13327 @@ -2,25 +2,27 @@
13328 * aluno: joilnen leite
13329 */
13330 #include <stdio.h>

```

```

13331 -#include "testa_item_1.h"
13332 -#include "testa_item_2.h"
13333 -#include "testa_item_3.h"
13334 -#include "testa_rb.h"
13335 -#include "jurses.h"
13336 +#include "testa.h"
13337
13338 /**
13339  * estas funcoes mostram testes referentes
13340  * aos itens da avaliacao
13341  * * sao pares com testa_item*
13342  */
13343 -static void tela_item_1();
13344 -static void tela_item_2();
13345 -static void tela_item_3();
13346 -static void tela_testa_rb();
13347 +static void tela_item_1(const char *m);
13348 +static void tela_item_2(const char *m);
13349 +static void tela_item_3(const char *m);
13350 +static void tela_testa_rb(const char *m);
13351
13352 int main()
13353 {
13354     char op, nitems = 4;
13355
13356     /**
13357     * mensagem de controle
13358     */
13359     const char mc[] = "tecle enter ";
13360
13361     /**
13362     * aqui poderia ser escrito
13363     * "case 0:
13364     @@ -36,42 +38,46 @@ int main()
13365     {
13366         switch (op)
13367         {
13368             case 0: tela_item_1(); break;
13369             case 1: tela_item_2(); break;
13370             case 2: tela_item_3(); break;
13371             case 3: tela_testa_rb();
13372             + case 0: tela_item_1(mc); break;
13373             + case 1: tela_item_2(mc); break;
13374             + case 2: tela_item_3(mc); break;
13375             + case 3: tela_testa_rb(mc);
13376         }
13377         getc(stdin);
13378         switch (op)
13379         {
13380             case 0: testa_item_1(); break;
13381             case 1: testa_item_1(); break;
13382             case 2: testa_item_1(); break;
13383             case 3: testa_arvore_rb();
13384             + case 0: testa_item_1(mc); break;
13385             + case 1: testa_item_1(mc); break;
13386             + case 2: testa_item_1(mc); break;
13387             + case 3: testa_arvore_rb(mc);
13388         }
13389     }
13390
13391     return 0;
13392 }
13393
13394 -void tela_item_1()
13395 +void tela_item_1(const char *m)
13396 {
13397     printf("Testa item 1\n");
13398     + printf("Sera efetuado testes do item 1\n");
13399     printf("%s\n", m);
13400 }
13401
13402 -void tela_item_2()
13403 +void tela_item_2(const char *m)
13404 {
13405     printf("Testa item 2\n");
13406     + printf("%s\n", m);
13407 }
13408
13409 -void tela_item_3()
13410 +void tela_item_3(const char *m)
13411 {
13412     printf("Testa item 3\n");
13413     + printf("%s\n", m);
13414 }
13415
13416 -void tela_testa_rb()
13417 +void tela_testa_rb(const char *m)
13418 {
13419     printf("Testa rb\n");
13420     + printf("%s\n", m);
13421 }
13422
13423
13424 diff --git a/src/testa.h b/src/testa.h
13425 new file mode 100644
13426 index 0000000..494ed68
13427 --- /dev/null
13428 +++ b/src/testa.h
13429 @@ -0,0 +1,11 @@
13430 +#ifndef TESTA_H_
13431 +#define TESTA_H_
13432 +
13433 +#include "jurses.h"
13434 +#include "testa_item_1.h"
13435 +#include "testa_item_2.h"
13436 +#include "testa_item_3.h"
13437 +#include "testa_rb.h"
13438 +
13439 +#endif
13440 +
13441 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
13442 index 5fab149..87127d4 100644
13443 --- a/src/testa_item_1.c
13444 +++ b/src/testa_item_1.c
13445 @@ -1,6 +1,6 @@
13446 +#include "testa_item_1.h"

```

```

13447
13448 -void testa_item_1()
13449 +void testa_item_1(const char *m)
13450 {
13451 }
13452 }
13453 diff --git a/src/testa_item_1.h b/src/testa_item_1.h
13454 index 38d9cb8..095609e 100644
13455 --- a/src/testa_item_1.h
13456 +++ b/src/testa_item_1.h
13457 @@ -1,7 +1,7 @@
13458 #ifndef TESTA_ITEM_1_H_
13459 #define TESTA_ITEM_1_H_
13460
13461 -void testa_item_1();
13462 +void testa_item_1(const char *m);
13463
13464 #endif
13465
13466 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
13467 index 56c9478..9ba76fc 100644
13468 --- a/src/testa_item_2.c
13469 +++ b/src/testa_item_2.c
13470 @@ -1,6 +1,6 @@
13471 #include "testa_item_2.h"
13472
13473 -void testa_item_2()
13474 +void testa_item_2(const char *m)
13475 {
13476 }
13477 }
13478 diff --git a/src/testa_item_2.h b/src/testa_item_2.h
13479 index 8f0417b..e674e4c 100644
13480 --- a/src/testa_item_2.h
13481 +++ b/src/testa_item_2.h
13482 @@ -1,7 +1,7 @@
13483 #ifndef TESTA_ITEM_2_H_
13484 #define TESTA_ITEM_2_H_
13485
13486 -void testa_item_2();
13487 +void testa_item_2(const char *m);
13488
13489 #endif
13490
13491 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
13492 index c68e970..032dbbf 100644
13493 --- a/src/testa_item_3.c
13494 +++ b/src/testa_item_3.c
13495 @@ -1,6 +1,6 @@
13496 #include "testa_item_3.h"
13497
13498 -void testa_item_3()
13499 +void testa_item_3(const char *m)
13500 {
13501 }
13502 }
13503 diff --git a/src/testa_item_3.h b/src/testa_item_3.h
13504 index db34708..e1a9564 100644
13505 --- a/src/testa_item_3.h
13506 +++ b/src/testa_item_3.h
13507 @@ -1,7 +1,7 @@
13508 #ifndef TESTA_ITEM_3_H_
13509 #define TESTA_ITEM_3_H_
13510
13511 -void testa_item_3();
13512 +void testa_item_3(const char *m);
13513
13514 #endif
13515
13516 diff --git a/src/testa_rb.c b/src/testa_rb.c
13517 index 09dbc3e..60f20cf 100644
13518 --- a/src/testa_rb.c
13519 +++ b/src/testa_rb.c
13520 @@ -18,7 +18,7 @@ void imprime_meu_tipo(void *);
13521 /**
13522  * funcao main de teste separada do codigo da biblioteca como especificado
13523  */
13524 -int testa_arvore_rb()
13525 +int testa_arvore_rb(const char *m)
13526 {
13527     /**
13528      * definicao de um tipo exemplo pelo o usuario
13529 diff --git a/src/testa_rb.h b/src/testa_rb.h
13530 index 30e4996..3ef3e40 100644
13531 --- a/src/testa_rb.h
13532 +++ b/src/testa_rb.h
13533 @@ -9,7 +9,7 @@
13534 * declaracao da funcao de teste da
13535 * arvore vermelha e preta
13536 */
13537 -void testa_arvore_rb();
13538 +void testa_arvore_rb(const char *m);
13539
13540 #endif
13541
13542
13543 commit 839252593a7200de2e48a7ee4c4031ee8651812a
13544 Author: Joilnen <joilnen@gmail.com>
13545 Date: Thu Dec 30 16:22:09 2021 -0300
13546
13547     modularizando melhor os casos de teste
13548
13549 diff --git a/src/main.c b/src/main.c
13550 index 87e586b..b781b2d 100644
13551 --- a/src/main.c
13552 +++ b/src/main.c
13553 @@ -4,16 +4,6 @@
13554 #include <stdio.h>
13555 #include "testa.h"
13556
13557 -/**
13558 - * estas funcoes mostram testes referentes
13559 - * aos itens da avaliacao
13560 - * sao pares com testa_item*
13561 - */
13562 -static void tela_item_1(const char *m);

```

```

13563 -static void tela_item_2(const char *m);
13564 -static void tela_item_3(const char *m);
13565 -static void tela_testa_rb(const char *m);
13566 -
13567 int main()
13568 {
13569     char op, nitems = 4;
13570 @@ -56,28 +46,3 @@ int main()
13571     return 0;
13572 }
13573
13574 -void tela_item_1(const char *m)
13575 -{
13576     printf("Sera efetuado testes do item 1\n");
13577     printf("%s\n", m);
13578 -}
13579 -
13580 -void tela_item_2(const char *m)
13581 -{
13582     printf("Testa item 2\n");
13583     printf("%s\n", m);
13584 -}
13585 -
13586 -void tela_item_3(const char *m)
13587 -{
13588     printf("Testa item 3\n");
13589     printf("%s\n", m);
13590 -}
13591 -
13592 -void tela_testa_rb(const char *m)
13593 -{
13594     printf("Testa rb\n");
13595     printf("%s\n", m);
13596 -}
13597 -
13598 -
13599 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
13600 index 87127d4..07bf3b4 100644
13601 --- a/src/testa_item_1.c
13602 +++ b/src/testa_item_1.c
13603 @@ -1,3 +1,4 @@
13604 +#include <stdio.h>
13605 +#include "testa_item_1.h"
13606
13607 void testa_item_1(const char *m)
13608 @@ -5,4 +6,9 @@ void testa_item_1(const char *m)
13609 }
13610
13611 +void tela_item_1(const char *m)
13612 +{
13613     printf("Sera efetuado testes do item 1\n");
13614     printf("%s\n", m);
13615 +}
13616 +
13617 diff --git a/src/testa_item_1.h b/src/testa_item_1.h
13618 index 095609e..d22ff07 100644
13619 --- a/src/testa_item_1.h
13620 +++ b/src/testa_item_1.h
13621 @@ -1,6 +1,7 @@
13622 @@ -1,6 +1,7 @@
13623 #ifndef TESTA_ITEM_1_H_
13624 #define TESTA_ITEM_1_H_
13625
13626 +void tela_item_1(const char *m);
13627 void testa_item_1(const char *m);
13628
13629 #endif
13630 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
13631 index 9ba76fc..c5db65c 100644
13632 --- a/src/testa_item_2.c
13633 +++ b/src/testa_item_2.c
13634 @@ -1,5 +1,13 @@
13635 +#include <stdio.h>
13636 +#include "testa_item_2.h"
13637
13638 +void tela_item_2(const char *m)
13639 +{
13640     printf("Testa item 2\n");
13641     printf("%s\n", m);
13642 +}
13643 +
13644 +
13645 void testa_item_2(const char *m)
13646 {
13647
13648 diff --git a/src/testa_item_2.h b/src/testa_item_2.h
13649 index e674e4c..ad0f154 100644
13650 --- a/src/testa_item_2.h
13651 +++ b/src/testa_item_2.h
13652 @@ -1,6 +1,7 @@
13653 @@ -1,6 +1,7 @@
13654 #ifndef TESTA_ITEM_2_H_
13655 #define TESTA_ITEM_2_H_
13656
13657 +void tela_item_2(const char *m);
13658 void testa_item_2(const char *m);
13659
13660 #endif
13661 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
13662 index 032dbbf..8c87537 100644
13663 --- a/src/testa_item_3.c
13664 +++ b/src/testa_item_3.c
13665 @@ -1,5 +1,13 @@
13666 +#include <stdio.h>
13667 +#include "testa_item_3.h"
13668
13669 +void tela_item_3(const char *m)
13670 +{
13671     printf("Testa item 3\n");
13672     printf("%s\n", m);
13673 +}
13674 +
13675 +
13676 void testa_item_3(const char *m)
13677 {
13678
13679 diff --git a/src/testa_item_3.h b/src/testa_item_3.h

```

```

13679 index e1a9564..3808f42 100644
13680 --- a/src/testa_item_3.h
13681 +++ b/src/testa_item_3.h
13682 @@ -1,6 +1,7 @@
13683 #ifndef TESTA_ITEM_3_H_
13684 #define TESTA_ITEM_3_H_
13685
13686 +void tela_item_3(const char *m);
13687 void testa_item_3(const char *m);
13688
13689 #endif
13690 diff --git a/src/testa_rb.c b/src/testa_rb.c
13691 index 60f20cf..c1cd79e 100644
13692 --- a/src/testa_rb.c
13693 +++ b/src/testa_rb.c
13694 @@ -9,6 +9,12 @@
13695 #include <stdio.h>
13696 #include <string.h>
13697
13698 +void tela_testa_rb(const char *m)
13699 +{
13700 +    printf("Testa rb\n");
13701 +    printf("%s\n", m);
13702 +}
13703 +
13704 /**
13705  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
13706  * @noh rb
13707 diff --git a/src/testa_rb.h b/src/testa_rb.h
13708 index 3ef3e40..d568f6b 100644
13709 --- a/src/testa_rb.h
13710 +++ b/src/testa_rb.h
13711 @@ -9,6 +9,7 @@
13712 * declaracao da funcao de teste da
13713 * arvore vermelha e preta
13714 */
13715 +void tela_testa_rb(const char *m);
13716 void testa_arvore_rb(const char *m);
13717
13718 #endif
13719
13720 commit 2ea2d4d7266cdc915a69675e366b757b5afce4fc
13721 Author: Joilnen <joilnen@gmail.com>
13722 Date: Thu Dec 30 17:39:25 2021 -0300
13723
13724     melhorando a formatacao das telas de teste
13725
13726 diff --git a/src/main.c b/src/main.c
13727 index b781b2d..0c5ddeb 100644
13728 --- a/src/main.c
13729 +++ b/src/main.c
13730 @@ -11,7 +11,7 @@ int main()
13731     /**
13732     * mensagem de controle
13733     */
13734     const char mc[] = "tecle enter ";
13735     const char mc[] = "\033[24;65f* tecle enter * ";
13736
13737     /**
13738     * aqui poderia ser escrito
13739 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
13740 index 07bf3b4..54986a7 100644
13741 --- a/src/testa_item_1.c
13742 +++ b/src/testa_item_1.c
13743 @@ -1,4 +1,5 @@
13744 #include <stdio.h>
13745 #include "conjunto_ordenado.h"
13746 #include "testa_item_1.h"
13747
13748 void testa_item_1(const char *m)
13749 @@ -8,7 +9,8 @@ void testa_item_1(const char *m)
13750
13751 void tela_item_1(const char *m)
13752 {
13753     LTELA;
13754     printf("Sera efetuado testes do item 1\n");
13755     printf("%s\n", m);
13756     printf("%s", m);
13757 }
13758
13759 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
13760 index c5db65c..1c7407a 100644
13761 --- a/src/testa_item_2.c
13762 +++ b/src/testa_item_2.c
13763 @@ -1,10 +1,12 @@
13764 #include <stdio.h>
13765 #include "conjunto_ordenado.h"
13766 #include "testa_item_2.h"
13767
13768 void tela_item_2(const char *m)
13769 {
13770     LTELA;
13771     printf("Testa item 2\n");
13772     printf("%s\n", m);
13773     printf("%s", m);
13774 }
13775
13776 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
13777 index 8c87537..7142cb2 100644
13778 --- a/src/testa_item_3.c
13779 +++ b/src/testa_item_3.c
13780 @@ -1,10 +1,12 @@
13781 #include <stdio.h>
13782 #include "conjunto_ordenado.h"
13783 #include "testa_item_3.h"
13784
13785 void tela_item_3(const char *m)
13786 {
13787     LTELA;
13788     printf("Testa item 3\n");
13789     printf("%s\n", m);
13790     printf("%s", m);
13791 }
13792
13793
13794

```



```

13795 diff --git a/src/testa_rb.c b/src/testa_rb.c
13796 index c1cd79e..b5381bf 100644
13797 --- a/src/testa_rb.c
13798 +++ b/src/testa_rb.c
13799 @@ -11,8 +11,10 @@
13800
13801 void tela_testa_rb(const char *m)
13802 {
13803     LTELA;
13804     printf("Testa rb\n");
13805     printf("%s\n", m);
13806     printf("\033[24d");
13807     printf("%s", m);
13808 }
13809
13810 /**
13811 @@ -63,7 +65,7 @@ int testa_arvore_rb(const char *m)
13812     printf("= arvore =====\n");
13813     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13814     printf("= =====\n");
13815     printf("* tecle enter *\n");
13816     printf("%s", m);
13817     getc(stdin);
13818
13819     meu_tipo mt8;
13820 @@ -78,8 +80,6 @@ int testa_arvore_rb(const char *m)
13821
13822     rb_insert(&conjunto_a, novo_noh);
13823
13824     -
13825     -
13826     meu_tipo mt1;
13827     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
13828     strcpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
13829 @@ -98,7 +98,7 @@ int testa_arvore_rb(const char *m)
13830     printf("= arvore =====\n");
13831     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13832     printf("= =====\n");
13833     printf("* tecle enter *\n");
13834     printf("%s", m);
13835     getc(stdin);
13836
13837     meu_tipo mt2;
13838 @@ -118,7 +118,7 @@ int testa_arvore_rb(const char *m)
13839     printf("= arvore =====\n");
13840     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13841     printf("= =====\n");
13842     printf("* tecle enter *\n");
13843     printf("%s", m);
13844     getc(stdin);
13845
13846     meu_tipo mt3;
13847 @@ -138,7 +138,7 @@ int testa_arvore_rb(const char *m)
13848     printf("= arvore =====\n");
13849     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13850     printf("= =====\n");
13851     printf("* tecle enter *\n");
13852     printf("%s", m);
13853     getc(stdin);
13854
13855     meu_tipo mt4;
13856 @@ -158,7 +158,7 @@ int testa_arvore_rb(const char *m)
13857     printf("= arvore =====\n");
13858     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13859     printf("= =====\n");
13860     printf("* tecle enter *\n");
13861     printf("%s", m);
13862     getc(stdin);
13863
13864     meu_tipo mt5;
13865 @@ -178,7 +178,7 @@ int testa_arvore_rb(const char *m)
13866     printf("= arvore =====\n");
13867     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13868     printf("= =====\n");
13869     printf("* tecle enter *\n");
13870     printf("%s", m);
13871
13872     meu_tipo mt6;
13873     mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
13874 @@ -197,7 +197,7 @@ int testa_arvore_rb(const char *m)
13875     printf("= arvore =====\n");
13876     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13877     printf("= =====\n");
13878     printf("* tecle enter *\n");
13879     printf("%s", m);
13880     getc(stdin);
13881
13882     meu_tipo mt7;
13883 @@ -217,7 +217,7 @@ int testa_arvore_rb(const char *m)
13884     printf("= arvore =====\n");
13885     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13886     printf("= =====\n");
13887     printf("* tecle enter *\n");
13888     printf("%s", m);
13889     getc(stdin);
13890
13891     rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
13892 @@ -228,7 +228,7 @@ int testa_arvore_rb(const char *m)
13893     printf("= arvore =====\n");
13894     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13895     printf("= =====\n");
13896     printf("* tecle enter *\n");
13897     printf("%s", m);
13898     getc(stdin);
13899
13900     rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
13901 @@ -238,7 +238,7 @@ int testa_arvore_rb(const char *m)
13902     printf("= arvore =====\n");
13903     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13904     printf("= =====\n");
13905     printf("* tecle enter *\n");
13906     printf("%s", m);
13907     getc(stdin);
13908
13909     rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
13910 @@ -248,7 +248,7 @@ int testa_arvore_rb(const char *m)

```

```

13911     printf("= arvore =====\n");
13912     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13913     printf("= =====\n");
13914 -     printf("* tecla enter *\n");
13915 +     printf("%s", m);
13916     getc(stdin);
13917
13918     rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
13919 @@ -258,7 +258,7 @@ int testa_arvore_rb(const char *m)
13920     printf("= arvore =====\n");
13921     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13922     printf("= =====\n");
13923 -     printf("* tecla enter *\n");
13924 +     printf("%s", m);
13925     getc(stdin);
13926
13927     rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
13928 @@ -268,7 +268,7 @@ int testa_arvore_rb(const char *m)
13929     printf("= arvore =====\n");
13930     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13931     printf("= =====\n");
13932 -     printf("* tecla enter *\n");
13933 +     printf("%s", m);
13934     getc(stdin);
13935     rb_pr(conjunto_a, TCOLOR);
13936     printf("\n");
13937 @@ -280,7 +280,7 @@ int testa_arvore_rb(const char *m)
13938     printf("= arvore =====\n");
13939     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13940     printf("= =====\n");
13941 -     printf("* tecla enter *\n");
13942 +     printf("%s", m);
13943     getc(stdin);
13944
13945     rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
13946 @@ -290,7 +290,7 @@ int testa_arvore_rb(const char *m)
13947     printf("= arvore =====\n");
13948     conjord_imprime_rb(conjunto_a, 1, TCOLOR);
13949     printf("= =====\n");
13950 -     printf("* tecla enter *\n");
13951 +     printf("%s", m);
13952     getc(stdin);
13953
13954     rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
13955
13956 commit 0f09061f4fb9e2eca112aec43d79f86115af8d79
13957 Author: Joilnen <joilnen@gmail.com>
13958 Date:   Fri Dec 31 13:11:14 2021 -0300
13959
13960     criando o TAD "conjord" ele faz a interface e oculta a arvore, o cliente
13961     n lida com funcoes da arvore ou sua estrutura diretamente nunca
13962
13963 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
13964 index fca731f..5d5f73f 100644
13965 --- a/src/conjunto_ordenado.c
13966 +++ b/src/conjunto_ordenado.c
13967 @@ -1,19 +1,44 @@
13968 #include <stdio.h>
13969 #include "conjunto_ordenado.h"
13970
13971 +static rb_n **raiz;
13972 +static unsigned int r_counter = 0;
13973 +
13974 /**
13975  * cria conjunto ordenado e retorna o ponteiro para ele
13976  */
13977 conjord *conjord_cria(char *nome)
13978 {
13979     conjord *c = (conjord *) malloc (sizeof(conjord));
13980 -     c->raiz = NULL;
13981 +     c->pr = NULL;
13982 +     c->idx = r_counter;
13983     c->nome = nome;
13984     c->cardinalidade = 0;
13985
13986     return c;
13987 }
13988
13989 /**
13990  * insere elemento no conjunto ordenado
13991  */
13992 void conjord_insere_elemento(conjord *a, int chave, void *elemento)
13993 {
13994     if (a->pr)
13995     {
13996         rb_n *n = (rb_n *) malloc(sizeof(rb_n));
13997         n->k = chave;
13998         n->d = elemento;
13999         if(!rb_insert(&a->pr, n))
14000             a->cardinalidade++;
14001     }
14002     else
14003     {
14004         raiz = (rb_n **) malloc(sizeof(rb_n));
14005         a->pr = raiz[a->idx] = rb_cr(chave, elemento);
14006         a->cardinalidade++;
14007     }
14008 }
14009
14010 /**
14011  * imprime elementos do conjunto subitem 1 do item 2
14012  * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
14013 @@ -28,7 +53,7 @@ void conjord_imprime()
14014 */
14015 int conjord_eh_vazio(const conjord *a)
14016 {
14017     return !a->raiz;
14018 +     return !a->pr;
14019 }
14020
14021 /**
14022 @@ -40,6 +65,41 @@ int conjord_esvazia(conjord *a)
14023 }
14024 }
14025
14026 void conjord_imprime_arvore_rb(conjord *c, int s)

```

```

14027 +{
14028 +    const int dist = 8;
14029 +    int e = TIPO_DE_TERMINAL;
14030 +    rb_n *t = c->pr;
14031 +    if (t == &rb_s)
14032 +        return;
14033 +
14034 +    s += dist;
14035 +
14036 +    conjord_imprime_rb(t->r, s, e);
14037 +
14038 +    // for (int i = dist; i < s; i++)
14039 +    //     printf("%s", s, "");
14040 +    // printf("%*c", s - dist, ' ');
14041 +    printf("%*c", s - dist, ' ');
14042 +    if(t->c == BLACK)
14043 +    {
14044 +        if(e == TCOLOR)
14045 +            printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
14046 +        else
14047 +            printf("k:%dc:%d\n", t->k, t->c);
14048 +    }
14049 +    else
14050 +    {
14051 +        if(e == TCOLOR)
14052 +            printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
14053 +        else
14054 +            printf("\033[7mk:%dc:%d\033[0m\n", t->k, t->c);
14055 +    }
14056 +
14057 +    conjord_imprime_rb(t->l, s, e);
14058 +}
14059 +
14060 +
14061 +/**
14062 + * imprime arvore
14063 + * @param t noh raiz
14064 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
14065 index 9ea7863..de0b121 100644
14066 --- a/src/conjunto_ordenado.h
14067 +++ b/src/conjunto_ordenado.h
14068 @@ -14,7 +14,8 @@
14069  * exigidas no documento da avaliacao para este TAD
14070  */
14071 typedef struct conjord {
14072     rb_n *raiz;
14073     rb_n *pr;
14074     unsigned int idx;
14075     char *nome;
14076     int cardinalidade;
14077 } conjord;
14078 @@ -24,6 +25,11 @@ typedef struct conjord {
14079     */
14080     conjord *conjord_cria(char *nome);
14081
14082 +/**
14083 + * insere elemento no conjunto ordenado
14084 + */
14085 +void conjord_insere_elemento(conjord *a, int chave, void *elemento);
14086 +
14087 +/**
14088 + * imprime elementos do conjunto subitem 1 do item 2
14089 + * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
14090 @@ -50,11 +56,6 @@ int conjord_esvazia();
14091     */
14092     void conjord_destroi(conjord **a);
14093
14094 -/**
14095 - * insere elemento no conjunto ordenado
14096 - */
14097 -void conjord_insere_elemento(conjord *a, int chave, void *elemento);
14098 -
14099 -/**
14100 - * exclui elemento do conjunto ordenado através da chave
14101 - * é o identificador de acordo com especificacao
14102 @@ -96,6 +97,16 @@ conjord *conjord_subtrai(conjord *a, conjord *b);
14103     */
14104     void conjord_imprime_rb(rb_n *t, int s, int e);
14105
14106 +/**
14107 + * item 3 da especificacao
14108 + * "Imprime a arvore rubro negra
14109 + * nesta operacao, cada noh da arvore serah
14110 + * representado pela sua chave de identificacao seguida da sua cor"
14111 + * @param c ponteiro para o conjunto ordenado
14112 + * @param s espaco entre os nohs na apresentacao da arvore
14113 + */
14114 +void conjord_imprime_arvore_rb(conjord *c, int s);
14115 +
14116 +/**
14117 + * foi utilizado macro em vez de funcao
14118 + * pra evitar o footprint da chamada
14119 @@ -105,7 +116,8 @@ void conjord_imprime_rb(rb_n *t, int s, int e);
14120 + * foi primeiro implementado com um loop
14121 + * mas pensando melhor dessa forma mais rapido
14122 + */
14123 #define LTELA printf(LIMPA_TELA);
14124 #define LTELA printf(LIMPA_TELA)
14125 #define TIPO_DE_TERMINAL TCOLOR
14126
14127 #endif
14128
14129 diff --git a/src/testa_rb.c b/src/testa_rb.c
14130 index b5381bf..9659682 100644
14131 --- a/src/testa_rb.c
14132 +++ b/src/testa_rb.c
14133 @@ -50,257 +50,273 @@ int testa_arvore_rb(const char *m)
14134     */
14135     rb_sete_impr_elem(imprime_meu_tipo);
14136
14137 - /**
14138 - * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
14139 - * uma interface mais intuitiva ao usuario da biblioteca
14140 - * @parametro 0 nome variavel do noh raiz
14141 - * @parametro 1 chave do noh raiz
14142 - * @parametro 2 dado criado neste noh, pode ser de qqr tipo

```

```

14143 - */
14144 - rb_n *conjunto_a = rb_cr(10, &mt0);
14145 -
14146 + conjord *cj_a = conjord_cria("A");
14147 + conjord_insere_elemento(cj_a, 10, &mt0);
14148 - LTELA;
14149 - rb_pr(conjunto_a, TCOLOR);
14150 - printf("\n");
14151 - printf("= arvore =====\n");
14152 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14153 - printf("= =====\n");
14154 - printf("%s", m);
14155 + conjord_imprime_arvore_rb(cj_a, 1);
14156 - getc(stdin);
14157 -
14158 - meu_tipo mt8;
14159 - mt8.nome = (char *) malloc(sizeof("nome para teste") + 1);
14160 - strncpy(mt8.nome, "nome para teste", sizeof("nome para teste") + 1);
14161 - mt8.telefone = (char *) malloc(sizeof("999999999") + 1);
14162 - strncpy(mt8.telefone, "999999999", sizeof("999999999") + 1);
14163 -
14164 - rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
14165 - novo_noh->k = 9;
14166 - novo_noh->d = &mt8;
14167 -
14168 - rb_insert(&conjunto_a, novo_noh);
14169 -
14170 - meu_tipo mt1;
14171 - mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
14172 - strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
14173 - mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
14174 - strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
14175 -
14176 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
14177 - novo_noh->k = 9;
14178 - novo_noh->d = &mt1;
14179 -
14180 - rb_insert(&conjunto_a, novo_noh);
14181 -
14182 - LTELA;
14183 - rb_pr(conjunto_a, TCOLOR);
14184 - printf("\n");
14185 - printf("= arvore =====\n");
14186 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14187 - printf("= =====\n");
14188 - printf("%s", m);
14189 - getc(stdin);
14190 -
14191 - meu_tipo mt2;
14192 - mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
14193 - strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
14194 - mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
14195 - strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
14196 -
14197 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
14198 - novo_noh->k = 11;
14199 - novo_noh->d = &mt2;
14200 - rb_insert(&conjunto_a, novo_noh);
14201 -
14202 - LTELA;
14203 - rb_pr(conjunto_a, TCOLOR);
14204 - printf("\n");
14205 - printf("= arvore =====\n");
14206 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14207 - printf("= =====\n");
14208 - printf("%s", m);
14209 - getc(stdin);
14210 -
14211 - meu_tipo mt3;
14212 - mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
14213 - strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
14214 - mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
14215 - strncpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
14216 -
14217 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
14218 - novo_noh->k = 8;
14219 - novo_noh->d = &mt3;
14220 - rb_insert(&conjunto_a, novo_noh);
14221 -
14222 - LTELA;
14223 - rb_pr(conjunto_a, TCOLOR);
14224 - printf("\n");
14225 - printf("= arvore =====\n");
14226 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14227 - printf("= =====\n");
14228 - printf("%s", m);
14229 - getc(stdin);
14230 -
14231 - meu_tipo mt4;
14232 - mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
14233 - strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
14234 - mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
14235 - strncpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
14236 -
14237 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
14238 - novo_noh->k = 6;
14239 - novo_noh->d = &mt4;
14240 - rb_insert(&conjunto_a, novo_noh);
14241 -
14242 - LTELA;
14243 - rb_pr(conjunto_a, TCOLOR);
14244 - printf("\n");
14245 - printf("= arvore =====\n");
14246 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14247 - printf("= =====\n");
14248 - printf("%s", m);
14249 - getc(stdin);
14250 -
14251 - meu_tipo mt5;
14252 - mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
14253 - strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
14254 - mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
14255 - strncpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
14256 -
14257 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
14258 - novo_noh->k = 5;

```

```

14259 - novo_noh->d = &mt5;
14260 - rb_insert(&conjunto_a, novo_noh);
14261 -
14262 + conjord_insererelemento(cj_a, 11, &mt1);
14263 - LTELA;
14264 - rb_pr(conjunto_a, TCOLOR);
14265 - printf("\n");
14266 - printf("= arvore =====\n");
14267 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14268 - printf("= =====\n");
14269 - printf("%s", m);
14270 + conjord_imprime_arvore_rb(cj_a, 1);
14271 -
14272 - meu_tipo mt6;
14273 - mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
14274 - strncpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
14275 - mt6.telefone = (char *) malloc(sizeof("999999999") + 1);
14276 - strncpy(mt6.telefone, "999999999", sizeof("999999999") + 1);
14277 -
14278 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
14279 - novo_noh->k = 4;
14280 - novo_noh->d = &mt6;
14281 - rb_insert(&conjunto_a, novo_noh);
14282 -
14283 - LTELA;
14284 - rb_pr(conjunto_a, TCOLOR);
14285 - printf("\n");
14286 - printf("= arvore =====\n");
14287 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14288 - printf("= =====\n");
14289 - printf("%s", m);
14290 - getc(stdin);
14291 -
14292 - meu_tipo mt7;
14293 - mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
14294 - strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
14295 - mt7.telefone = (char *) malloc(sizeof("999999999") + 1);
14296 - strncpy(mt7.telefone, "999999999", sizeof("999999999") + 1);
14297 -
14298 - novo_noh = (rb_n *) malloc (sizeof(rb_n));
14299 - novo_noh->k = 3;
14300 - novo_noh->d = &mt7;
14301 - rb_insert(&conjunto_a, novo_noh);
14302 -
14303 - LTELA;
14304 - rb_pr(conjunto_a, TCOLOR);
14305 - printf("\n");
14306 - printf("= arvore =====\n");
14307 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14308 - printf("= =====\n");
14309 - printf("%s", m);
14310 - getc(stdin);
14311 -
14312 - rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
14313 - printf("\n");
14314 - LTELA;
14315 - rb_pr(conjunto_a, TCOLOR);
14316 - printf("\n");
14317 - printf("= arvore =====\n");
14318 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14319 - printf("= =====\n");
14320 - printf("%s", m);
14321 - getc(stdin);
14322 -
14323 - rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
14324 - LTELA;
14325 - rb_pr(conjunto_a, TCOLOR);
14326 - printf("\n");
14327 - printf("= arvore =====\n");
14328 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14329 - printf("= =====\n");
14330 - printf("%s", m);
14331 - getc(stdin);
14332 -
14333 - rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
14334 - LTELA;
14335 - rb_pr(conjunto_a, TCOLOR);
14336 - printf("\n");
14337 - printf("= arvore =====\n");
14338 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14339 - printf("= =====\n");
14340 - printf("%s", m);
14341 - getc(stdin);
14342 -
14343 - rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
14344 - LTELA;
14345 - rb_pr(conjunto_a, TCOLOR);
14346 - printf("\n");
14347 - printf("= arvore =====\n");
14348 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14349 - printf("= =====\n");
14350 - printf("%s", m);
14351 - getc(stdin);
14352 -
14353 - rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
14354 - LTELA;
14355 - rb_pr(conjunto_a, TCOLOR);
14356 - printf("\n");
14357 - printf("= arvore =====\n");
14358 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14359 - printf("= =====\n");
14360 - printf("%s", m);
14361 - getc(stdin);
14362 - rb_pr(conjunto_a, TCOLOR);
14363 - printf("\n");
14364 -
14365 - rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
14366 - LTELA;
14367 - rb_pr(conjunto_a, TCOLOR);
14368 - printf("\n");
14369 - printf("= arvore =====\n");
14370 - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14371 - printf("= =====\n");
14372 - printf("%s", m);
14373 - getc(stdin);
14374 -

```

```

14375 | - rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
14376 | - LTELA;
14377 | - rb_pr(conjunto_a, TCOLOR);
14378 | - printf("\n");
14379 | - printf("= arvore =====\n");
14380 | - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14381 | - printf("= =====\n");
14382 | - printf("%s", m);
14383 | - getc(stdin);
14384 | -
14385 | - rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
14386 | - LTELA;
14387 | - printf("nada\n");
14388 | - printf("= arvore =====\n");
14389 | - conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14390 | - printf("= =====\n");
14391 | - rb_pr(conjunto_a, TCOLOR);
14392 | - printf("\n");
14393 | +
14394 | + /**
14395 | +  * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
14396 | +  * uma interface mais intuitiva ao usuario da biblioteca
14397 | +  * @parametro 0 nome variavel do noh raiz
14398 | +  * @parametro 1 chave do noh raiz
14399 | +  * @parametro 2 dado criado neste noh, pode ser de qqr tipo
14400 | +  */
14401 | + rb_n *conjunto_a = rb_cr(10, &mt0);
14402 | +
14403 | + LTELA;
14404 | + rb_pr(conjunto_a, TCOLOR);
14405 | + printf("\n");
14406 | + printf("= arvore =====\n");
14407 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14408 | + printf("= =====\n");
14409 | + printf("%s", m);
14410 | + getc(stdin);
14411 | +
14412 | + meu_tipo mt8;
14413 | + mt8.nome = (char *) malloc(sizeof("nome para teste") + 1);
14414 | + strncpy(mt8.nome, "nome para teste", sizeof("nome para teste") + 1);
14415 | + mt8.telefone = (char *) malloc(sizeof("999999999") + 1);
14416 | + strncpy(mt8.telefone, "999999999", sizeof("999999999") + 1);
14417 | +
14418 | + rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
14419 | + novo_noh->k = 9;
14420 | + novo_noh->d = &mt8;
14421 | +
14422 | + rb_insert(&conjunto_a, novo_noh);
14423 | +
14424 | + meu_tipo mt1;
14425 | + mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
14426 | + strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
14427 | + mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
14428 | + strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
14429 | +
14430 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14431 | + novo_noh->k = 9;
14432 | + novo_noh->d = &mt1;
14433 | +
14434 | + rb_insert(&conjunto_a, novo_noh);
14435 | +
14436 | + LTELA;
14437 | + rb_pr(conjunto_a, TCOLOR);
14438 | + printf("\n");
14439 | + printf("= arvore =====\n");
14440 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14441 | + printf("= =====\n");
14442 | + printf("%s", m);
14443 | + getc(stdin);
14444 | +
14445 | + meu_tipo mt2;
14446 | + mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
14447 | + strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
14448 | + mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
14449 | + strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
14450 | +
14451 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14452 | + novo_noh->k = 11;
14453 | + novo_noh->d = &mt2;
14454 | + rb_insert(&conjunto_a, novo_noh);
14455 | +
14456 | + LTELA;
14457 | + rb_pr(conjunto_a, TCOLOR);
14458 | + printf("\n");
14459 | + printf("= arvore =====\n");
14460 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14461 | + printf("= =====\n");
14462 | + printf("%s", m);
14463 | + getc(stdin);
14464 | +
14465 | + meu_tipo mt3;
14466 | + mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
14467 | + strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
14468 | + mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
14469 | + strncpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
14470 | +
14471 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14472 | + novo_noh->k = 8;
14473 | + novo_noh->d = &mt3;
14474 | + rb_insert(&conjunto_a, novo_noh);
14475 | +
14476 | + LTELA;
14477 | + rb_pr(conjunto_a, TCOLOR);
14478 | + printf("\n");
14479 | + printf("= arvore =====\n");
14480 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14481 | + printf("= =====\n");
14482 | + printf("%s", m);
14483 | + getc(stdin);
14484 | +
14485 | + meu_tipo mt4;
14486 | + mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
14487 | + strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
14488 | + mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
14489 | + strncpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
14490 | +
14491 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));

```

```

14491 //      novo_noh->k = 6;
14492 //      novo_noh->d = &mt4;
14493 //      rb_insert(&conjunto_a, novo_noh);
14494 //
14495 //      LTELA;
14496 //      rb_pr(conjunto_a, TCOLOR);
14497 //      printf("\n");
14498 //      printf("= arvore =====\n");
14499 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14500 //      printf("= =====\n");
14501 //      printf("%s", m);
14502 //      getc(stdin);
14503 //
14504 //      neu_tipo mt5;
14505 //      mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
14506 //      strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
14507 //      mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
14508 //      strncpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
14509 //
14510 //      novo_noh = (rb_n *) malloc (sizeof(rb_n));
14511 //      novo_noh->k = 5;
14512 //      novo_noh->d = &mt5;
14513 //      rb_insert(&conjunto_a, novo_noh);
14514 //
14515 //      LTELA;
14516 //      rb_pr(conjunto_a, TCOLOR);
14517 //      printf("\n");
14518 //      printf("= arvore =====\n");
14519 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14520 //      printf("= =====\n");
14521 //      printf("%s", m);
14522 //
14523 //      neu_tipo mt6;
14524 //      mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
14525 //      strncpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
14526 //      mt6.telefone = (char *) malloc(sizeof("999999999") + 1);
14527 //      strncpy(mt6.telefone, "999999999", sizeof("999999999") + 1);
14528 //
14529 //      novo_noh = (rb_n *) malloc (sizeof(rb_n));
14530 //      novo_noh->k = 4;
14531 //      novo_noh->d = &mt6;
14532 //      rb_insert(&conjunto_a, novo_noh);
14533 //
14534 //      LTELA;
14535 //      rb_pr(conjunto_a, TCOLOR);
14536 //      printf("\n");
14537 //      printf("= arvore =====\n");
14538 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14539 //      printf("= =====\n");
14540 //      printf("%s", m);
14541 //      getc(stdin);
14542 //
14543 //      neu_tipo mt7;
14544 //      mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
14545 //      strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
14546 //      mt7.telefone = (char *) malloc(sizeof("999999999") + 1);
14547 //      strncpy(mt7.telefone, "999999999", sizeof("999999999") + 1);
14548 //
14549 //      novo_noh = (rb_n *) malloc (sizeof(rb_n));
14550 //      novo_noh->k = 3;
14551 //      novo_noh->d = &mt7;
14552 //      rb_insert(&conjunto_a, novo_noh);
14553 //
14554 //      LTELA;
14555 //      rb_pr(conjunto_a, TCOLOR);
14556 //      printf("\n");
14557 //      printf("= arvore =====\n");
14558 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14559 //      printf("= =====\n");
14560 //      printf("%s", m);
14561 //      getc(stdin);
14562 //
14563 //      rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
14564 //      printf("\n");
14565 //      LTELA;
14566 //      rb_pr(conjunto_a, TCOLOR);
14567 //      printf("\n");
14568 //      printf("= arvore =====\n");
14569 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14570 //      printf("= =====\n");
14571 //      printf("%s", m);
14572 //      getc(stdin);
14573 //
14574 //      rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
14575 //      LTELA;
14576 //      rb_pr(conjunto_a, TCOLOR);
14577 //      printf("\n");
14578 //      printf("= arvore =====\n");
14579 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14580 //      printf("= =====\n");
14581 //      printf("%s", m);
14582 //      getc(stdin);
14583 //
14584 //      rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
14585 //      LTELA;
14586 //      rb_pr(conjunto_a, TCOLOR);
14587 //      printf("\n");
14588 //      printf("= arvore =====\n");
14589 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14590 //      printf("= =====\n");
14591 //      printf("%s", m);
14592 //      getc(stdin);
14593 //
14594 //      rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
14595 //      LTELA;
14596 //      rb_pr(conjunto_a, TCOLOR);
14597 //      printf("\n");
14598 //      printf("= arvore =====\n");
14599 //      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14600 //      printf("= =====\n");
14601 //      printf("%s", m);
14602 //      getc(stdin);
14603 //
14604 //      rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
14605 //      LTELA;
14606 //      rb_pr(conjunto_a, TCOLOR);

```

```

14607 +//      printf("\n");
14608 +//      printf("= arvore =====\n");
14609 +//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14610 +//      printf("= =====\n");
14611 +//      printf("%s", m);
14612 +//      getc(stdin);
14613 +//      rb_pr(conjunto_a, TCOLOR);
14614 +//      printf("\n");
14615 +//
14616 +//      rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
14617 +//      LTELA;
14618 +//      rb_pr(conjunto_a, TCOLOR);
14619 +//      printf("\n");
14620 +//      printf("= arvore =====\n");
14621 +//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14622 +//      printf("= =====\n");
14623 +//      printf("%s", m);
14624 +//      getc(stdin);
14625 +//
14626 +//      rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
14627 +//      LTELA;
14628 +//      rb_pr(conjunto_a, TCOLOR);
14629 +//      printf("\n");
14630 +//      printf("= arvore =====\n");
14631 +//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14632 +//      printf("= =====\n");
14633 +//      printf("%s", m);
14634 +//      getc(stdin);
14635 +//
14636 +//      rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
14637 +//      LTELA;
14638 +//      printf("nada\n");
14639 +//      printf("= arvore =====\n");
14640 +//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14641 +//      printf("= =====\n");
14642 +//      rb_pr(conjunto_a, TCOLOR);
14643 +//      printf("\n");
14644 +//
14645 +//      * testando apenas rb no momento
14646 +//      conjord *conj_a = conjord_cria();
14647 +//      diff --git a/src/testa_rb.c b/src/testa_rb.c
14648 +//      new file mode 100644
14649 +//      index 0000000..b5381bf
14650 +//      --- /dev/null
14651 +//      +++ b/src/testa_rb.c
14652 +//      @@ -0,0 +1,320 @@
14653 +//
14654 +//      * * aluno: joilnen leite
14655 +//      * */
14656 +//
14657 +//      * */
14658 +//      * inclusao da biblioteca implementada baseada em arvores red black
14659 +//      * */
14660 +//      #include "conjunto_ordenado.h"
14661 +//      #include <stdio.h>
14662 +//      #include <string.h>
14663 +//
14664 +//      void tela_testa_rb(const char *m)
14665 +//      {
14666 +//          LTELA;
14667 +//          printf("Testa rb\n");
14668 +//          printf("\033[24d");
14669 +//          printf("%s", m);
14670 +//      }
14671 +//
14672 +//      * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
14673 +//      * * @noh rb
14674 +//      * */
14675 +//
14676 +//      void imprime_meu_tipo(void *);
14677 +//
14678 +//      * */
14679 +//      * funcao main de teste separada do codigo da biblioteca como especificado
14680 +//      * */
14681 +//      int testa_arvore_rb(const char *m)
14682 +//      {
14683 +//          /**
14684 +//           * definicao de um tipo exemplo pelo o usuario
14685 +//           */
14686 +//          typedef struct {
14687 +//              char *nome;
14688 +//              char *telefone;
14689 +//          } meu_tipo;
14690 +//
14691 +//          /**
14692 +//           * criacao do dado para o tipo do usuario
14693 +//           */
14694 +//          meu_tipo mt0;
14695 +//          mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
14696 +//          strncpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
14697 +//          mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
14698 +//          strncpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
14699 +//
14700 +//          /**
14701 +//           * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
14702 +//           */
14703 +//          rb_sete_impr_elem(imprime_meu_tipo);
14704 +//
14705 +//          /**
14706 +//           * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
14707 +//           * uma interface mais intuitiva ao usuario da biblioteca
14708 +//           * @parametro 0 nome variavel do noh raiz
14709 +//           * @parametro 1 chave do noh raiz
14710 +//           * @parametro 2 dado criado neste noh, pode ser de qqr tipo
14711 +//           */
14712 +//          rb_n *conjunto_a = rb_cr(10, &mt0);
14713 +//
14714 +//          LTELA;
14715 +//          rb_pr(conjunto_a, TCOLOR);
14716 +//          printf("\n");
14717 +//          printf("= arvore =====\n");
14718 +//          conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14719 +//          printf("= =====\n");
14720 +//          printf("%s", m);
14721 +//          getc(stdin);
14722 +//

```



```

14723 | + meu_tipo mt8;
14724 | + mt8.nome = (char *) malloc(sizeof("nome para teste") + 1);
14725 | + strncpy(mt8.nome, "nome para teste", sizeof("nome para teste") + 1);
14726 | + mt8.telefone = (char *) malloc(sizeof("999999999") + 1);
14727 | + strncpy(mt8.telefone, "999999999", sizeof("999999999") + 1);
14728 | +
14729 | + rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
14730 | + novo_noh->k = 9;
14731 | + novo_noh->d = &mt8;
14732 | +
14733 | + rb_insert(&conjunto_a, novo_noh);
14734 | +
14735 | + meu_tipo mt1;
14736 | + mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
14737 | + strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
14738 | + mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
14739 | + strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
14740 | +
14741 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14742 | + novo_noh->k = 9;
14743 | + novo_noh->d = &mt1;
14744 | +
14745 | + rb_insert(&conjunto_a, novo_noh);
14746 | +
14747 | + LTELA;
14748 | + rb_pr(conjunto_a, TCOLOR);
14749 | + printf("\n");
14750 | + printf("= arvore =====\n");
14751 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14752 | + printf("= =====\n");
14753 | + printf("%s", m);
14754 | + getc(stdin);
14755 | +
14756 | + meu_tipo mt2;
14757 | + mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
14758 | + strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
14759 | + mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
14760 | + strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
14761 | +
14762 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14763 | + novo_noh->k = 11;
14764 | + novo_noh->d = &mt2;
14765 | + rb_insert(&conjunto_a, novo_noh);
14766 | +
14767 | + LTELA;
14768 | + rb_pr(conjunto_a, TCOLOR);
14769 | + printf("\n");
14770 | + printf("= arvore =====\n");
14771 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14772 | + printf("= =====\n");
14773 | + printf("%s", m);
14774 | + getc(stdin);
14775 | +
14776 | + meu_tipo mt3;
14777 | + mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
14778 | + strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
14779 | + mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
14780 | + strncpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
14781 | +
14782 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14783 | + novo_noh->k = 8;
14784 | + novo_noh->d = &mt3;
14785 | + rb_insert(&conjunto_a, novo_noh);
14786 | +
14787 | + LTELA;
14788 | + rb_pr(conjunto_a, TCOLOR);
14789 | + printf("\n");
14790 | + printf("= arvore =====\n");
14791 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14792 | + printf("= =====\n");
14793 | + printf("%s", m);
14794 | + getc(stdin);
14795 | +
14796 | + meu_tipo mt4;
14797 | + mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
14798 | + strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
14799 | + mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
14800 | + strncpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
14801 | +
14802 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14803 | + novo_noh->k = 6;
14804 | + novo_noh->d = &mt4;
14805 | + rb_insert(&conjunto_a, novo_noh);
14806 | +
14807 | + LTELA;
14808 | + rb_pr(conjunto_a, TCOLOR);
14809 | + printf("\n");
14810 | + printf("= arvore =====\n");
14811 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14812 | + printf("= =====\n");
14813 | + printf("%s", m);
14814 | + getc(stdin);
14815 | +
14816 | + meu_tipo mt5;
14817 | + mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
14818 | + strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
14819 | + mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
14820 | + strncpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
14821 | +
14822 | + novo_noh = (rb_n *) malloc (sizeof(rb_n));
14823 | + novo_noh->k = 5;
14824 | + novo_noh->d = &mt5;
14825 | + rb_insert(&conjunto_a, novo_noh);
14826 | +
14827 | + LTELA;
14828 | + rb_pr(conjunto_a, TCOLOR);
14829 | + printf("\n");
14830 | + printf("= arvore =====\n");
14831 | + conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14832 | + printf("= =====\n");
14833 | + printf("%s", m);
14834 | +
14835 | + meu_tipo mt6;
14836 | + mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
14837 | + strncpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
14838 | + mt6.telefone = (char *) malloc(sizeof("999999999") + 1);

```

```

14839 +   strncpy(mt6.telefone, "999999999", sizeof("999999999") + 1);
14840 +
14841 +   novo_noh = (rb_n *) malloc (sizeof(rb_n));
14842 +   novo_noh->k = 4;
14843 +   novo_noh->d = &mt6;
14844 +   rb_insert(&conjunto_a, novo_noh);
14845 +
14846 +   LTELA;
14847 +   rb_pr(conjunto_a, TCOLOR);
14848 +   printf("\n");
14849 +   printf("= arvore =====\n");
14850 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14851 +   printf("= =====\n");
14852 +   printf("%s", m);
14853 +   getc(stdin);
14854 +
14855 +   meu_tipo mt7;
14856 +   mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
14857 +   strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
14858 +   mt7.telefone = (char *) malloc(sizeof("999999999") + 1);
14859 +   strncpy(mt7.telefone, "999999999", sizeof("999999999") + 1);
14860 +
14861 +   novo_noh = (rb_n *) malloc (sizeof(rb_n));
14862 +   novo_noh->k = 3;
14863 +   novo_noh->d = &mt7;
14864 +   rb_insert(&conjunto_a, novo_noh);
14865 +
14866 +   LTELA;
14867 +   rb_pr(conjunto_a, TCOLOR);
14868 +   printf("\n");
14869 +   printf("= arvore =====\n");
14870 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14871 +   printf("= =====\n");
14872 +   printf("%s", m);
14873 +   getc(stdin);
14874 +
14875 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
14876 +   printf("\n");
14877 +   LTELA;
14878 +   rb_pr(conjunto_a, TCOLOR);
14879 +   printf("\n");
14880 +   printf("= arvore =====\n");
14881 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14882 +   printf("= =====\n");
14883 +   printf("%s", m);
14884 +   getc(stdin);
14885 +
14886 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
14887 +   LTELA;
14888 +   rb_pr(conjunto_a, TCOLOR);
14889 +   printf("\n");
14890 +   printf("= arvore =====\n");
14891 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14892 +   printf("= =====\n");
14893 +   printf("%s", m);
14894 +   getc(stdin);
14895 +
14896 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
14897 +   LTELA;
14898 +   rb_pr(conjunto_a, TCOLOR);
14899 +   printf("\n");
14900 +   printf("= arvore =====\n");
14901 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14902 +   printf("= =====\n");
14903 +   printf("%s", m);
14904 +   getc(stdin);
14905 +
14906 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
14907 +   LTELA;
14908 +   rb_pr(conjunto_a, TCOLOR);
14909 +   printf("\n");
14910 +   printf("= arvore =====\n");
14911 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14912 +   printf("= =====\n");
14913 +   printf("%s", m);
14914 +   getc(stdin);
14915 +
14916 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
14917 +   LTELA;
14918 +   rb_pr(conjunto_a, TCOLOR);
14919 +   printf("\n");
14920 +   printf("= arvore =====\n");
14921 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14922 +   printf("= =====\n");
14923 +   printf("%s", m);
14924 +   getc(stdin);
14925 +   rb_pr(conjunto_a, TCOLOR);
14926 +   printf("\n");
14927 +
14928 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
14929 +   LTELA;
14930 +   rb_pr(conjunto_a, TCOLOR);
14931 +   printf("\n");
14932 +   printf("= arvore =====\n");
14933 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14934 +   printf("= =====\n");
14935 +   printf("%s", m);
14936 +   getc(stdin);
14937 +
14938 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
14939 +   LTELA;
14940 +   rb_pr(conjunto_a, TCOLOR);
14941 +   printf("\n");
14942 +   printf("= arvore =====\n");
14943 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14944 +   printf("= =====\n");
14945 +   printf("%s", m);
14946 +   getc(stdin);
14947 +
14948 +   rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
14949 +   LTELA;
14950 +   printf("nada\n");
14951 +   printf("= arvore =====\n");
14952 +   conjord_imprime_rb(conjunto_a, 1, TCOLOR);
14953 +   printf("= =====\n");
14954 +   rb_pr(conjunto_a, TCOLOR);

```

```

14955 +     printf("\n");
14956 +     /**
14957 +      * testando apenas rb no momento
14958 +      conjord *conj_a = conjord_cria();
14959 +      printf("eh vazio %d", conjord_eh_vazio(conj_a));
14960 +      */
14961 + }
14962 +
14963 + /**
14964 +  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
14965 +  * exemplo
14966 +  */
14967 + void imprime_meu_tipo(void *n)
14968 + {
14969 + }
14970 +
14971 +
14972 +
14973 diff --git a/src/testa_rb.h b/src/testa_rb.h
14974 new file mode 100644
14975 index 0000000..d568f6b
14976 --- /dev/null
14977 +++ b/src/testa_rb.h
14978 @@ -0,0 +1,17 @@
14979 + #ifndef TESTA_ARVORE_RB_
14980 + #define TESTA_ARVORE_RB_
14981 +
14982 + /**
14983 +  * aluno: joilnen leite
14984 +  */
14985 +
14986 + /**
14987 +  * declaracao da funcao de teste da
14988 +  * arvore vermelha e preta
14989 +  */
14990 + void tela_testa_rb(const char *m);
14991 + void testa_arvore_rb(const char *m);
14992 +
14993 + #endif
14994 +
14995 +
14996 +
14997 commit d34ae11e21929bcb68af921767b81b5a7a7991e78
14998 Author: Joilnen <joilnen@gmail.com>
14999 Date: Fri Dec 31 13:45:43 2021 -0300
15000
15001     adicionando funcao que verifica a cardinalidade do conjunto
15002
15003 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
15004 index 5d5f73f..69788bf 100644
15005 --- a/src/conjunto_ordenado.c
15006 +++ b/src/conjunto_ordenado.c
15007 @@ -39,6 +39,14 @@ void conjord_insere_elemento(conjord *a, int chave, void *elemento)
15008     }
15009 }
15010
15011 +int conjord_cardinalidade(conjord *c)
15012 +{
15013 +    if(c)
15014 +        return c->cardinalidade;
15015 +    else
15016 +        return -1;
15017 +}
15018 +
15019 + /**
15020 +  * imprime elementos do conjunto subitem 1 do item 2
15021 +  * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
15022 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
15023 index de0b121..fb527ed 100644
15024 --- a/src/conjunto_ordenado.h
15025 +++ b/src/conjunto_ordenado.h
15026 @@ -39,7 +39,7 @@ void conjord_imprime();
15027 + /**
15028 +  * retorna a cardinalidade do conjunto
15029 +  */
15030 + unsigned int conjord_cardinalidade();
15031 + int conjord_cardinalidade();
15032 +
15033 + /**
15034 +  * testa se o conjunto estah vazio
15035 +  */
15036 +
15037 commit 6419abfeef9fe808f1d8a727aa58d23d98239705
15038 Author: Joilnen <joilnen@gmail.com>
15039 Date: Sat Jan 1 10:57:14 2022 -0300
15040
15041     melhorando o codigo de formatacao e adaptando a terminais mais primitivos
15042
15043 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
15044 index fb527ed..dfc6cd4 100644
15045 --- a/src/conjunto_ordenado.h
15046 +++ b/src/conjunto_ordenado.h
15047 @@ -118,6 +118,7 @@ void conjord_imprime_arvore_rb(conjord *c, int s);
15048 +
15049 + #define LTELA printf(LIMPA_TELA)
15050 + #define TIPO_DE_TERMINAL TCOLOR
15051 + #define CM printf("%s", m); fflush(stdout); getc(stdin)
15052 +
15053 + #endif
15054 +
15055 diff --git a/src/main.c b/src/main.c
15056 index 0c5ddeb..2b46c12 100644
15057 --- a/src/main.c
15058 +++ b/src/main.c
15059 @@ -11,7 +11,7 @@ int main()
15060 + /**
15061 +  * mensagem de controle
15062 +  */
15063 + const char mc[] = "\033[24;65f* tecle enter * ";
15064 + const char mc[] = "\033[24;65H* tecle enter * ";
15065 +
15066 + /**
15067 +  * aqui poderia ser escrito
15068 +  */
15069 + {
15070 +     switch (op)
15071 +     {

```

```

15071 -         case 0: tela_item_1(mc); break;
15072 -         case 1: tela_item_2(mc); break;
15073 -         case 2: tela_item_3(mc); break;
15074 -         case 3: tela_testa_rb(mc);
15075 -     }
15076 -     getc(stdin);
15077 -     switch (op)
15078 -     {
15079 -         case 0: testa_item_1(mc); break;
15080 -         case 1: testa_item_1(mc); break;
15081 -         case 2: testa_item_1(mc); break;
15082 -         case 3: testa_arvore_rb(mc);
15083 +         case 0:
15084 +             tela_item_1(mc);
15085 +             testa_item_1(mc);
15086 +             break;
15087 +         case 1:
15088 +             tela_item_2(mc);
15089 +             testa_item_1(mc);
15090 +             break;
15091 +         case 2:
15092 +             tela_item_3(mc);
15093 +             testa_item_1(mc);
15094 +             break;
15095 +         case 3:
15096 +             tela_testa_rb(mc);
15097 +             testa_arvore_rb(mc);
15098     }
15099 }
15100
15101     return 0;
15102 }
15103
15104 +
15105 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
15106 index 54986a7..612a912 100644
15107 --- a/src/testa_item_1.c
15108 +++ b/src/testa_item_1.c
15109 @@ -11,6 +11,6 @@ void tela_item_1(const char *m)
15110 {
15111     LTELA;
15112     printf("Sera efetuado testes do item 1\n");
15113 -     printf("%s", m);
15114 +     CM;
15115 }
15116
15117 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
15118 index 1c7407a..481f6e6 100644
15119 --- a/src/testa_item_2.c
15120 +++ b/src/testa_item_2.c
15121 @@ -6,7 +6,7 @@ void tela_item_2(const char *m)
15122 {
15123     LTELA;
15124     printf("Testa item 2\n");
15125 -     printf("%s", m);
15126 +     CM;
15127 }
15128
15129 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
15130 index 7142cb2..358dia8 100644
15131 --- a/src/testa_item_3.c
15132 +++ b/src/testa_item_3.c
15133 @@ -6,7 +6,7 @@ void tela_item_3(const char *m)
15134 {
15135     LTELA;
15136     printf("Testa item 3\n");
15137 -     printf("%s", m);
15138 +     CM;
15139 }
15140
15141 diff --git a/src/testa_rb.c b/src/testa_rb.c
15142 index 9659682..d5fe07a 100644
15143 --- a/src/testa_rb.c
15144 +++ b/src/testa_rb.c
15145 @@ -13,8 +13,7 @@ void tela_testa_rb(const char *m)
15146 {
15147     LTELA;
15148     printf("Testa rb\n");
15149 -     printf("\033[24d");
15150 -     printf("%s", m);
15151 +     CM;
15152 }
15153
15154 /**
15155  * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
15156  */
15157 @@ -54,7 +53,7 @@ int testa_arvore_rb(const char *m)
15158     conjord_inserir_elemento(cj_a, 10, &mt0);
15159     LTELA;
15160     conjord_imprime_arvore_rb(cj_a, 1);
15161 -     getc(stdin);
15162 +     CM;
15163
15164     meu_tipo mt1;
15165     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
15166 @@ -65,6 +64,7 @@ int testa_arvore_rb(const char *m)
15167     conjord_inserir_elemento(cj_a, 11, &mt1);
15168     LTELA;
15169     conjord_imprime_arvore_rb(cj_a, 1);
15170 +     CM;
15171
15172     /**
15173      * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
15174      */
15175
15176 commit d57fd62ae426ffb03d0c2b05973eb0c3a571334f
15177 Author: Joilnen <joilnen@gmail.com>
15178 Date: Sat Jan 1 12:09:24 2022 -0300
15179
15180     ajustado a posicao da mensagem de controle pra caber no mais rudimentar
15181     vt100 de 21(4)x80(1)
15182
15183 diff --git a/src/main.c b/src/main.c
15184 index 2b46c12..b5a68f2 100644
15185 --- a/src/main.c
15186 +++ b/src/main.c
15187 @@ -11,7 +11,7 @@ int main()

```

```

15187 /**
15188  * mensagem de controle
15189  */
15190 + const char mc[] = "\033[24;65H* tecle enter * ";
15191 + const char mc[] = "\033[24;64H* tecle enter * ";
15192
15193 /**
15194  * aqui poderia ser escrito
15195
15196 commit 65031dda6033a9ff25add41b54bc036848aa3dic
15197 Author: Joillnen <joillnen@gmail.com>
15198 Date: Sat Jan 1 12:44:58 2022 -0300
15199
15200 passei a funcao de impressao da arvore para o fonte da arvore nada da arvore
15201 eh manipulado fora do seu fonte
15202
15203 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
15204 index 69788bf..65745cf 100644
15205 --- a/src/conjunto_ordenado.c
15206 +++ b/src/conjunto_ordenado.c
15207 @@ -75,75 +75,7 @@ int conjord_esvazia(conjord *a)
15208
15209 void conjord_imprime_arvore_rb(conjord *c, int s)
15210 {
15211 - const int dist = 8;
15212 - int e = TIPO_DE_TERMINAL;
15213 - rb_n *t = c->r;
15214 - if (t == &rb_s)
15215 - return;
15216 -
15217 - s += dist;
15218 -
15219 - conjord_imprime_rb(t->r, s, e);
15220 -
15221 - // for (int i = dist; i < s; i++)
15222 - // printf(" ");
15223 - // printf("%*s", s, "");
15224 - printf("%*c", s - dist, ' ');
15225 - if(t->c == BLACK)
15226 - {
15227 - if(e == TCOLOR)
15228 - printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
15229 - else
15230 - printf("k:%dc:%d\n", t->k, t->c);
15231 - }
15232 - else
15233 - {
15234 - if(e == TCOLOR)
15235 - printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
15236 - else
15237 - printf("\033[7mk:%dc:%d\033[0m\n", t->k, t->c);
15238 - }
15239 -
15240 - conjord_imprime_rb(t->l, s, e);
15241 + rb_pr2(c->r, s, TCOLOR);
15242 + }
15243 +
15244 + /**
15245 + * imprime arvore
15246 + * @param t noh raiz
15247 + * @param s espaco entre os nohs na impressao
15248 + * @param e tipo do terminal
15249 + */
15250 + void conjord_imprime_rb(rb_n *t, int s, int e) {
15251 - const int dist = 8;
15252 - if (t == &rb_s)
15253 - return;
15254 -
15255 - s += dist;
15256 -
15257 - conjord_imprime_rb(t->r, s, e);
15258 -
15259 - // for (int i = dist; i < s; i++)
15260 - // printf(" ");
15261 - // printf("%*s", s, "");
15262 - printf("%*c", s - dist, ' ');
15263 - if(t->c == BLACK)
15264 - {
15265 - if(e == TCOLOR)
15266 - printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
15267 - else
15268 - printf("k:%dc:%d\n", t->k, t->c);
15269 - }
15270 - else
15271 - {
15272 - if(e == TCOLOR)
15273 - printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
15274 - else
15275 - printf("\033[7mk:%dc:%d\033[0m\n", t->k, t->c);
15276 - }
15277 -
15278 - conjord_imprime_rb(t->l, s, e);
15279 - }
15280 -
15281 - }
15282 -
15283
15284 diff --git a/src/red_black.c b/src/red_black.c
15285 index 96a1c53..4628fa7 100644
15286 --- a/src/red_black.c
15287 +++ b/src/red_black.c
15288 @@ -398,6 +398,37 @@ void rb_pr(rb_n *a, int t)
15289 }
15290
15291 + void rb_pr2(rb_n *t, int s, int e) {
15292 + const int dist = 8;
15293 + if (t == &rb_s)
15294 + return;
15295 +
15296 + s += dist;
15297 +
15298 + rb_pr2(t->r, s, e);
15299 +
15300 + // for (int i = dist; i < s; i++)
15301 + // printf(" ");
15302 +

```

```

15303 + // printf("%s", s, "");
15304 + printf("%*c", s - dist, ' ');
15305 + if(t->c == BLACK)
15306 + {
15307 +     if(e == TCOLOR)
15308 +         printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
15309 +     else
15310 +         printf("k:%dc:%d\n", t->k, t->c);
15311 + }
15312 + }
15313 + else
15314 + {
15315 +     if(e == TCOLOR)
15316 +         printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
15317 +     else
15318 +         printf("\033[7mk:%dc:%d\033[0m\n", t->k, t->c);
15319 + }
15320 + }
15321 + rb_pr2(t->l, s, e);
15322 +}
15323
15324 rb_n *rb_search(rb_n *t, int k)
15325 {
15326 diff --git a/src/red_black.h b/src/red_black.h
15327 index 6b6c1ce..eff729e 100644
15328 --- a/src/red_black.h
15329 +++ b/src/red_black.h
15330 @@ -163,4 +163,11 @@ int rb_set_impr_elem(void (*i)(void *n));
15331 */
15332 void rb_pr(rb_n *a, int t);
15333
15334 /**
15335  * imprime arvore uma segunda versao
15336  * @param a raiz
15337  * @param t tipo de terminal
15338  */
15339 void rb_pr2(rb_n *t, int s, int e);
15340 +
15341 #endif
15342
15343 commit a380e2b75a5394085e45a2ccf9b61dab5c7ebc0e
15344 Author: Joilnen <joilnen@gmail.com>
15345 Date: Sat Jan 1 13:37:22 2022 -0300
15346
15347     aumentando o teste na arvore, parece robusta
15348
15349 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
15350 index 65745cf..a0c92d3 100644
15351 --- a/src/conjunto_ordenado.c
15352 +++ b/src/conjunto_ordenado.c
15353 @@ -39,6 +39,11 @@ void conjord_insererelemento(conjord *a, int chave, void *elemento)
15354 }
15355 }
15356
15357 void conjord_deleta_elemento(conjord *a, int chave)
15358 {
15359 +     rb_delete(&a->pr, rb_search(a->pr, chave));
15360 +}
15361 +
15362 int conjord_cardinalidade(conjord *c)
15363 {
15364     if(c)
15365 @@ -75,7 +80,7 @@ int conjord_esvazia(conjord *a)
15366
15367 void conjord_imprime_arvore_rb(conjord *c, int s)
15368 {
15369 -     rb_pr2(c->pr, s, TCOLOR);
15370 +     rb_pr(c->pr, s, TCOLOR);
15371 }
15372
15373 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
15374 index dfc6cd4..660aa13 100644
15375 --- a/src/conjunto_ordenado.h
15376 +++ b/src/conjunto_ordenado.h
15377 @@ -30,6 +30,11 @@ conjord *conjord_cria(char *nome);
15378 */
15379 void conjord_insererelemento(conjord *a, int chave, void *elemento);
15380
15381 /**
15382  * insere elemento no conjunto ordenado
15383  */
15384 void conjord_deleta_elemento(conjord *a, int chave);
15385 +
15386 /**
15387  * imprime elementos do conjunto subitem 1 do item 2
15388  * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
15389 diff --git a/src/red_black.c b/src/red_black.c
15390 index 4628fa7..d75b7fd 100644
15391 --- a/src/red_black.c
15392 +++ b/src/red_black.c
15393 @@ -375,30 +375,38 @@ int rb_set_impr_elem(void (*i)(void *n))
15394     rb_impr_elem = i;
15395 }
15396
15397 -void rb_pr(rb_n *a, int t)
15398 +void rb_pr(rb_n *t, int s, int e)
15399 {
15400     if (a != &rb_s)
15401 +     const int dist = 5;
15402     if (t == &rb_s)
15403         return;
15404
15405     s += dist;
15406
15407     rb_pr(t->r, s, e);
15408
15409     printf("%*c", s - dist, ' ');
15410     if(t->c == BLACK)
15411     {
15412         if (a->c == BLACK)
15413         {
15414             if(t == TCOLOR)
15415                 printf("%sk:%dc:%d%s ", S_AZUL, a->k, a->c, S_NORM);
15416             else
15417                 printf("k:%dc:%d ", a->k, a->c);
15418

```

```

15419 -     }
15420 +     if(e == TCOLOR)
15421 +         printf("\033[44m\033[37m%d\033[0m\n", t->k);
15422 -     else
15423 -     {
15424 -         if (t == TCOLOR)
15425 -             printf("%sk:%dc:%d%s ", S_VERM, a->k, a->c, S_NORM);
15426 -         else
15427 -             printf("%sk:%dc:%d%s ", S_INV, a->k, a->c, S_NORM);
15428 -     }
15429 -     rb_pr(a->l, t);
15430 -     rb_pr(a->r, t);
15431 +     printf("%d\n", t->k);
15432 +
15433 + }
15434 + else
15435 + {
15436 +     if(e == TCOLOR)
15437 +         printf("\033[41m\033[37m%d\033[0m\n", t->k);
15438 +     else
15439 +         printf("\033[7m%d\033[0m\n", t->k, t->c);
15440 + }
15441 +
15442 + rb_pr(t->l, s, e);
15443 + }
15444
15445 -void rb_pr2(rb_n *t, int s, int e) {
15446 +void rb_pr2(rb_n *t, int s, int e)
15447 +{
15448 +    const int dist = 8;
15449 +    if (t == &rb_s)
15450 +        return;
15451 diff --git a/src/red_black.h b/src/red_black.h
15452 index eff729e..b349178 100644
15453 --- a/src/red_black.h
15454 +++ b/src/red_black.h
15455 @@ -161,7 +161,7 @@ int rb_sete_impr_elem(void (*i)(void *n));
15456 * @param a raiz
15457 * @param t tipo de terminal
15458 */
15459 -void rb_pr(rb_n *a, int t);
15460 +void rb_pr(rb_n *t, int s, int e);
15461
15462 /**
15463  * imprime arvore uma segunda versao
15464 diff --git a/src/testa_rb.c b/src/testa_rb.c
15465 index d5fe07a..b4559fa 100644
15466 --- a/src/testa_rb.c
15467 +++ b/src/testa_rb.c
15468 @@ -66,6 +66,39 @@ int testa_arvore_rb(const char *m)
15469     conjord_imprime_arvore_rb(cj_a, 1);
15470     CM;
15471
15472 + int ctest;
15473 + conjord *cj_b = conjord_cria("B");
15474 + for (ctest = 12; ctest < 23; ++ctest)
15475 + {
15476 +     conjord_insererelemento(cj_b, ctest, NULL);
15477 +     LTEL;
15478 +     conjord_imprime_arvore_rb(cj_b, 1);
15479 +     CM;
15480 + }
15481 +
15482 + for (ctest = 0; ctest < 13; ++ctest)
15483 + {
15484 +     conjord_insererelemento(cj_b, ctest, NULL);
15485 +     LTEL;
15486 +     conjord_imprime_arvore_rb(cj_b, 1);
15487 +     CM;
15488 + }
15489 +
15490 + for (ctest = 13; ctest < 23; ++ctest)
15491 + {
15492 +     conjord_deleta_elemento(cj_b, ctest);
15493 +     LTEL;
15494 +     conjord_imprime_arvore_rb(cj_b, 1);
15495 +     CM;
15496 + }
15497 +
15498 + for (ctest = 0; ctest < 13; ++ctest)
15499 + {
15500 +     conjord_deleta_elemento(cj_b, ctest);
15501 +     LTEL;
15502 +     conjord_imprime_arvore_rb(cj_b, 1);
15503 +     CM;
15504 + }
15505 + /**
15506  * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
15507  * uma interface mais intuitiva ao usuario da biblioteca
15508
15509 commit 7426f462a6f234b9c363292cc7f92b07b03ae8f5
15510 Author: Joilnen <joilnen@gmail.com>
15511 Date: Sat Jan 1 15:17:06 2022 -0300
15512
15513     teste massivo na arvore rb, funcionando perfeitamente, muito bom
15514
15515 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
15516 index a0c92d3..d284131 100644
15517 --- a/src/conjunto_ordenado.c
15518 +++ b/src/conjunto_ordenado.c
15519 @@ -83,4 +83,13 @@ void conjord_imprime_arvore_rb(conjord *c, int s)
15520     rb_pr(c->pr, s, TCOLOR);
15521 }
15522
15523 +void conjord_imprime_lista(int *l, int s)
15524 +{
15525 +    int i;
15526 +    for (i = 0; i < s; i++)
15527 +    {
15528 +        printf("\033[%d;78H%d", l[i] + 1, l[i]);
15529 +    }
15530 +}
15531 +
15532 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
15533 index 660aa13..b2cb82b 100644

```

```

15535 --- a/src/conjunto_ordenado.h
15536 +++ b/src/conjunto_ordenado.h
15537 @@ -112,6 +112,8 @@ void conjord_imprime_rb(rb_n *, int s, int e);
15538 */
15539 void conjord_imprime_arvore_rb(conjord *c, int s);
15540
15541 +void conjord_imprime_lista(int *l, int s);
15542 +
15543 /**
15544  * foi utilizado macro em vez de funcao
15545  * pra evitar o footprint da chamada
15546 diff --git a/src/testa_rb.c b/src/testa_rb.c
15547 index b4559fa..2e01fec 100644
15548 --- a/src/testa_rb.c
15549 +++ b/src/testa_rb.c
15550 @@ -68,37 +68,38 @@ int testa_arvore_rb(const char *m)
15551
15552     int ctest;
15553     conjord *cj_b = conjord_cria("B");
15554     for (ctest = 12; ctest < 23; ++ctest)
15555 +     const int size_n = 23;
15556 +     int test_n[size_n];
15557 +     for (ctest = 0; ctest < size_n; ++ctest)
15558     {
15559         conjord_inserir_elemento(cj_b, ctest, NULL);
15560         LTELA;
15561         conjord_imprime_arvore_rb(cj_b, 1);
15562         CM;
15563         if(ctest < 10) test_n[ctest] = ctest + 13;
15564         else test_n[ctest] = ctest - 10;
15565     }
15566
15567     for (ctest = 0; ctest < 13; ++ctest)
15568 +     LTELA;
15569 +     conjord_imprime_lista(test_n, size_n);
15570 +     CM;
15571 +     for (ctest = 0; ctest < size_n; ++ctest)
15572     {
15573         conjord_inserir_elemento(cj_b, ctest, NULL);
15574         conjord_inserir_elemento(cj_b, test_n[ctest], NULL);
15575         LTELA;
15576         conjord_imprime_lista(&test_n[ctest + 1], size_n - ctest - 1);
15577         printf("\033[H");
15578         conjord_imprime_arvore_rb(cj_b, 1);
15579         fflush(stdout);
15580         CM;
15581     }
15582
15583     for (ctest = 13; ctest < 23; ++ctest)
15584 +     for (ctest = 0; ctest < 23; ++ctest)
15585     {
15586         conjord_deleta_elemento(cj_b, ctest);
15587         LTELA;
15588         conjord_imprime_lista(test_n, ctest + 1);
15589         conjord_deleta_elemento(cj_b, test_n[ctest]);
15590         printf("\033[H");
15591         conjord_imprime_arvore_rb(cj_b, 1);
15592         CM;
15593     }
15594
15595     for (ctest = 0; ctest < 13; ++ctest)
15596     {
15597         conjord_deleta_elemento(cj_b, ctest);
15598         LTELA;
15599         conjord_imprime_arvore_rb(cj_b, 1);
15600         CM;
15601     }
15602     /**
15603      * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
15604      * uma interface mais intuitiva ao usuario da biblioteca
15605
15606 commit 205d70bd8f89b9718e3fc888da4533544281fae8
15607 Author: Joilnen <joilnen@gmail.com>
15608 Date: Sat Jan 1 15:19:28 2022 -0300
15609
15610     teste massivo na arvore
15611
15612 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
15613 index d284131..90f99c4 100644
15614 --- a/src/conjunto_ordenado.c
15615 +++ b/src/conjunto_ordenado.c
15616 @@ -87,9 +87,7 @@ void conjord_imprime_lista(int *l, int s)
15617     {
15618         int i;
15619         for (i = 0; i < s; i++)
15620         {
15621             printf("\033[%d;78H%d", l[i] + 1, l[i]);
15622         }
15623     }
15624
15625
15626
15627 commit 5c83aaaf1d166e620793cc871c19680a53b98517
15628 Author: Joilnen <joilnen@gmail.com>
15629 Date: Sat Jan 1 15:36:15 2022 -0300
15630
15631     ajustando para 21 linhas que sao os terminais mais antigos
15632
15633 diff --git a/src/main.c b/src/main.c
15634 index b5a68f2..3155c68 100644
15635 --- a/src/main.c
15636 +++ b/src/main.c
15637 @@ -11,7 +11,7 @@ int main()
15638     /**
15639      * mensagem de controle
15640      */
15641     const char mc[] = "\033[24;64H* tecla enter * ";
15642 +     const char mc[] = "\033[24;62H* tecla enter * ";
15643
15644     /**
15645      * aqui poderia ser escrito
15646 diff --git a/src/testa_rb.c b/src/testa_rb.c
15647 index 2e01fec..8d4e13f 100644
15648 --- a/src/testa_rb.c
15649 +++ b/src/testa_rb.c
15650 @@ -68,11 +68,11 @@ int testa_arvore_rb(const char *m)

```



```

15651 int ctest;
15652 conjord *cj_b = conjord_cria("B");
15653 - const int size_n = 23;
15654 + const int size_n = 21;
15655 int test_n[size_n];
15656 for (ctest = 0; ctest < size_n; ++ctest)
15657 {
15658     if(ctest < 10) test_n[ctest] = ctest + 13;
15659 + if(ctest < 10) test_n[ctest] = ctest + 11;
15660 else test_n[ctest] = ctest - 10;
15661 }
15662
15663 @@ -90,7 +90,7 @@ int testa_arvore_rb(const char *m)
15664 CM;
15665 }
15666 }
15667
15668 - for (ctest = 0; ctest < 23; ++ctest)
15669 + for (ctest = 0; ctest < size_n; ++ctest)
15670 {
15671     LTELA;
15672     conjord_imprime_lista(test_n, ctest + 1);
15673
15674 commit fib1a079a4d65d70b841b5b6ebdfe2ea9e9e0ca8
15675 Author: Joilnen <joilnen@gmail.com>
15676 Date: Sat Jan 1 16:33:37 2022 -0300
15677
15678     pequeno ajuste no teste da arvore
15679
15680 diff --git a/src/testa_rb.c b/src/testa_rb.c
15681 index 8d4e13f..8a5729b 100644
15682 --- a/src/testa_rb.c
15683 +++ b/src/testa_rb.c
15684 @@ -51,9 +51,9 @@ int testa_arvore_rb(const char *m)
15685
15686     conjord *cj_a = conjord_cria("A");
15687     conjord_insere_elemento(cj_a, 10, &mt0);
15688 - LTELA;
15689 - conjord_imprime_arvore_rb(cj_a, 1);
15690 - CM;
15691 + // LTELA;
15692 + // conjord_imprime_arvore_rb(cj_a, 1);
15693 + // CM;
15694
15695     meu_tipo mt1;
15696     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
15697 @@ -62,9 +62,9 @@ int testa_arvore_rb(const char *m)
15698     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
15699
15700     conjord_insere_elemento(cj_a, 11, &mt1);
15701 - LTELA;
15702 - conjord_imprime_arvore_rb(cj_a, 1);
15703 - CM;
15704 + // LTELA;
15705 + // conjord_imprime_arvore_rb(cj_a, 1);
15706 + // CM;
15707
15708     int ctest;
15709     conjord *cj_b = conjord_cria("B");
15710 @@ -86,7 +86,6 @@ int testa_arvore_rb(const char *m)
15711     conjord_imprime_lista(&test_n[ctest + 1], size_n - ctest - 1);
15712     printf("\033[H");
15713     conjord_imprime_arvore_rb(cj_b, 1);
15714 - fflush(stdout);
15715     CM;
15716 }
15717
15718 @@ -99,6 +98,7 @@ int testa_arvore_rb(const char *m)
15719     conjord_imprime_arvore_rb(cj_b, 1);
15720     CM;
15721 }
15722 + LTELA;
15723
15724 /**
15725  * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
15726
15727 commit 637e6a6ef176073c5a2dc508caf32ffd0086089e
15728 Author: Joilnen <joilnen@gmail.com>
15729 Date: Sat Jan 1 16:35:15 2022 -0300
15730
15731     movendo arquivos desnecessarios no momento
15732
15733 diff --git a/src/testa_rb.c b/tests/testa_rb.c
15734 similarity index 100%
15735 rename from src/testa_rb.c
15736 rename to tests/testa_rb.c
15737 diff --git a/src/testa_rb.h b/tests/testa_rb.h
15738 similarity index 100%
15739 rename from src/testa_rb.h
15740 rename to tests/testa_rb.h
15741
15742 commit 82225ccfc50804e2170be7795bf3e4fbaca03303
15743 Author: Joilnen <joilnen@gmail.com>
15744 Date: Sun Jan 2 00:07:39 2022 -0300
15745
15746     impressao de elementos precisa ordenar e passar o CM para esse caso de teste
15747
15748 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
15749 index 90f99c4..b969f97 100644
15750 --- a/src/conjunto_ordenado.c
15751 +++ b/src/conjunto_ordenado.c
15752 @@ -90,4 +90,28 @@ void conjord_imprime_lista(int *l, int s)
15753     printf("\033[%d;78H%d", l[i] + 1, l[i]);
15754 }
15755
15756 +void (*conjord_imprime_elemento)(void *n);
15757 +
15758 +int conjord_seta_imprime_elemento(void (*)(void *n))
15759 +{
15760     conjord_imprime_elemento = i;
15761 +}
15762 +
15763 +static void conjord_imprime_conjunto_helper(rb_n *n)
15764 +{
15765     if (n == &rb_s)
15766         return;

```

```

15767 + printf("chave: %d\n", n->k);
15768 + conjord_imprime_elemento(n->d);
15769 +
15770 + getc(stdin);
15771 + conjord_imprime_conjunto_helper(n->l);
15772 + conjord_imprime_conjunto_helper(n->r);
15773 +}
15774 +
15775 +void conjord_imprime_conjunto(conjord *c)
15776 +{
15777 +    if (c && c->pr)
15778 +        conjord_imprime_conjunto_helper(c->pr);
15779 +}
15780
15781 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
15782 index b2cb82b..e4491be 100644
15783 --- a/src/conjunto_ordenado.h
15784 +++ b/src/conjunto_ordenado.h
15785 @@ -114,6 +114,15 @@ void conjord_imprime_arvore_rb(conjord *c, int s);
15786
15787 void conjord_imprime_lista(int *l, int s);
15788
15789 +extern void (*conjord_imprime_elemento)(void *n);
15790 +/**
15791 + * seta a rotina de impressao do dado criado pelo cliente
15792 + */
15793 +int conjord_sete_imprime_elemento(void (*i)(void *n));
15794 +
15795 +void conjord_imprime_conjunto(conjord *c);
15796 +
15797 +/**
15798 + * foi utilizado macro em vez de funcao
15799 + * pra evitar o footprint da chamada
15800 diff --git a/src/main.c b/src/main.c
15801 index 3155c68..7d68692 100644
15802 --- a/src/main.c
15803 +++ b/src/main.c
15804 @@ -34,11 +34,11 @@ int main()
15805 break;
15806 case 1:
15807     tela_item_2(mc);
15808     testa_item_1(mc);
15809     testa_item_2(mc);
15810 break;
15811 case 2:
15812     tela_item_3(mc);
15813     testa_item_1(mc);
15814     testa_item_3(mc);
15815 break;
15816 case 3:
15817     tela_testa_rb(mc);
15818 diff --git a/src/red_black.c b/src/red_black.c
15819 index d75b7fd..5cd3def 100644
15820 --- a/src/red_black.c
15821 +++ b/src/red_black.c
15822 @@ -370,11 +370,6 @@ int rb_rot_dir(rb_n **t, rb_n *x)
15823 return 0;
15824 }
15825
15826 -int rb_sete_impr_elem(void (*i)(void *n))
15827 -{
15828     rb_impr_elem = i;
15829 -}
15830
15831 void rb_pr(rb_n *t, int s, int e)
15832 {
15833     const int dist = 5;
15834 diff --git a/src/red_black.h b/src/red_black.h
15835 index b349178..271f6bc 100644
15836 --- a/src/red_black.h
15837 +++ b/src/red_black.h
15838 @@ -151,11 +151,6 @@ int rb_rot_dir(rb_n **t, rb_n *x);
15839 */
15840 extern void (*rb_impr_elem)(void *);
15841
15842 -/**
15843 - * seta a rotina de impressao do dado criado pelo cliente
15844 - */
15845 -int rb_sete_impr_elem(void (*i)(void *n));
15846 -
15847 /**
15848 * imprime arvore
15849 * @param a raiz
15850 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
15851 index 612a912..5f9431c 100644
15852 --- a/src/testa_item_1.c
15853 +++ b/src/testa_item_1.c
15854 @@ -1,10 +1,59 @@
15855 #include <stdio.h>
15856 #include <string.h>
15857 #include "conjunto_ordenado.h"
15858 #include "testa_item_1.h"
15859
15860 +static void imprime_meu_tipo(void *);
15861 +
15862 +typedef struct {
15863 +    char *nome;
15864 +    char *telefone;
15865 +} meu_tipo;
15866 +
15867 void testa_item_1(const char *m)
15868 {
15869     /**
15870     * definicao de um tipo exemplo pelo o usuario
15871     */
15872     /**
15873     * criacao do dado para o tipo do usuario
15874     */
15875     meu_tipo mt0;
15876     mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
15877     strcpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
15878     mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
15879     strcpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
15880

```

```

15883 + /**
15884 +  * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
15885 +  */
15886 + conjord_setar_imprime_elemento(imprime_meu_tipo);
15887
15888 + conjord *cj_a = conjord_cria("A");
15889 + conjord_inserir_elemento(cj_a, 10, &mt0);
15890 + // LTEL;
15891 + // conjord_imprime_arvore_rb(cj_a, 1);
15892 + // CM;
15893 +
15894 + meu_tipo mt1;
15895 + mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
15896 + strncpy(mt1.nome, "Gertrudes", sizeof("Gertrudes") + 1);
15897 + mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
15898 + strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
15899 +
15900 + conjord_inserir_elemento(cj_a, 11, &mt1);
15901 + // LTEL;
15902 + // conjord_imprime_arvore_rb(cj_a, 1);
15903 + // CM;
15904 +
15905 + LTEL;
15906 + conjord_imprime_elemento(&mt1);
15907 + CM;
15908 + LTEL;
15909 + conjord_imprime_elemento(&mt0);
15910 + CM;
15911 + conjord_imprime_conjunto(cj_a);
15912 + }
15913 +
15914 + void tela_item_1(const char *m)
15915 + @@ -14,3 +63,13 @@ void tela_item_1(const char *m)
15916 + CM;
15917 + }
15918 +
15919 +
15920 + void imprime_meu_tipo(void *n)
15921 + {
15922 + printf("===\n");
15923 + printf("Nome: %s\n", ((meu_tipo *)n)->nome);
15924 + printf("Telefone: %s\n", ((meu_tipo *)n)->telefone);
15925 + printf("===\n");
15926 + }
15927 +
15928 +
15929 + diff --git a/src/testa_rb.c b/src/testa_rb.c
15930 + index 8a5729b..c8aae9d 100644
15931 + --- a/src/testa_rb.c
15932 + +++ b/src/testa_rb.c
15933 + @@ -8,6 +8,7 @@
15934 + #include "conjunto_ordenado.h"
15935 + #include <stdio.h>
15936 + #include <string.h>
15937 + #include "testa_rb.h"
15938 +
15939 + void tela_testa_rb(const char *m)
15940 + {
15941 + @@ -20,12 +21,12 @@ void tela_testa_rb(const char *m)
15942 + * funcao de impressao do tipo especifico criado pelo o usuario da biblioteca
15943 + * @noh rb
15944 + */
15945 + -void imprime_meu_tipo(void *);
15946 + +static void imprime_meu_tipo(void *);
15947 +
15948 + /**
15949 +  * funcao main de teste separada do codigo da biblioteca como especificado
15950 +  */
15951 + -int testa_arvore_rb(const char *m)
15952 + +void testa_arvore_rb(const char *m)
15953 + {
15954 + /**
15955 +  * definicao de um tipo exemplo pelo o usuario
15956 +  @@ -47,7 +48,7 @@ int testa_arvore_rb(const char *m)
15957 +  /**
15958 +  * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
15959 +  */
15960 + - rb_setar_impr_elem(imprime_meu_tipo);
15961 + + conjord_setar_imprime_elemento(imprime_meu_tipo);
15962 +
15963 + conjord *cj_a = conjord_cria("A");
15964 + conjord_inserir_elemento(cj_a, 10, &mt0);
15965 +
15966 + commit 5741993e5f532d3b174e8d4a2cc1d8088d108364
15967 + Author: Joilnen <joilnen@gmail.com>
15968 + Date: Sun Jan 2 00:17:18 2022 -0300
15969 +
15970 + impressao dos elementos trabalhando nisto
15971 +
15972 + diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
15973 + index b969f97..2aebba6 100644
15974 + --- a/src/conjunto_ordenado.c
15975 + +++ b/src/conjunto_ordenado.c
15976 + @@ -102,7 +102,7 @@ static void conjord_imprime_conjunto_helper(rb_n *n)
15977 + if (n == &rb_s)
15978 + return;
15979 +
15980 + - printf("chave: %d\n", n->k);
15981 + + printf("chave: %d | ", n->k);
15982 + conjord_imprime_elemento(n->d);
15983 + getc(stdin);
15984 + conjord_imprime_conjunto_helper(n->l);
15985 + diff --git a/src/testa_item_1.c b/src/testa_item_1.c
15986 + index 5f9431c..f895ca2 100644
15987 + --- a/src/testa_item_1.c
15988 + +++ b/src/testa_item_1.c
15989 + @@ -66,10 +66,8 @@ void tela_item_1(const char *m)
15990 +
15991 + void imprime_meu_tipo(void *n)
15992 + {
15993 + printf("===\n");
15994 + - printf("Nome: %s\n", ((meu_tipo *)n)->nome);
15995 + - printf("Telefone: %s\n", ((meu_tipo *)n)->telefone);
15996 + + printf("===\n");
15997 + + printf("Nome: %s | ", ((meu_tipo *)n)->nome);
15998 + + printf("Telefone: %s", ((meu_tipo *)n)->telefone);

```

```

15999 }
16000
16001
16002
16003 commit 0eade589606ac59e8c981ae716777dbb00523a0e
16004 Author: Joilnen <joilnen@gmail.com>
16005 Date: Sun Jan 2 10:33:58 2022 -0300
16006
16007 feito exemplo do item 1.1 da avaliacao
16008
16009 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
16010 index 2aebba6..74dcaf8 100644
16011 --- a/src/conjunto_ordenado.c
16012 +++ b/src/conjunto_ordenado.c
16013 @@ -21,7 +21,7 @@ conjord *conjord_cria(char *nome)
16014 /**
16015  * insere elemento no conjunto ordenado
16016  */
16017 -void conjord_insererelemento(conjord *a, int chave, void *elemento)
16018 +int conjord_insererelemento(conjord *a, int chave, void *elemento)
16019 {
16020     if (a->pr)
16021     {
16022         @@ -30,6 +30,8 @@ void conjord_insererelemento(conjord *a, int chave, void *elemento)
16023         n->d = elemento;
16024         if (!rb_insert(&a->pr, n))
16025             a->cardinalidade++;
16026     }
16027     else
16028         return -1;
16029 }
16030 else
16031 {
16032     @@ -37,6 +39,8 @@ void conjord_insererelemento(conjord *a, int chave, void *elemento)
16033     a->pr = raiz[a->idx] = rb_cr(chave, elemento);
16034     a->cardinalidade++;
16035 }
16036 +
16037 + return 0;
16038 }
16039
16040 void conjord_deleta_elemento(conjord *a, int chave)
16041 @@ -46,7 +50,7 @@ void conjord_deleta_elemento(conjord *a, int chave)
16042
16043 int conjord_cardinalidade(conjord *c)
16044 {
16045     if (c)
16046     {
16047         if (c)
16048             return c->cardinalidade;
16049     }
16050     else
16051         return -1;
16052 @@ -102,10 +106,9 @@ static void conjord_imprime_conjunto_helper(rb_n *n)
16053     if (n == &rb_s)
16054         return;
16055
16056     printf("chave: %d | ", n->k);
16057     conjord_imprime_elemento(n->d);
16058     getc(stdin);
16059     conjord_imprime_conjunto_helper(n->l);
16060     printf("chave: %d\t", n->k);
16061     conjord_imprime_elemento(n->d);
16062     conjord_imprime_conjunto_helper(n->r);
16063 }
16064
16065 @@ -115,3 +118,4 @@ void conjord_imprime_conjunto(conjord *c)
16066     conjord_imprime_conjunto_helper(c->pr);
16067 }
16068
16069 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
16070 index e4491be..1b90579 100644
16071 --- a/src/conjunto_ordenado.h
16072 +++ b/src/conjunto_ordenado.h
16073 @@ -28,7 +28,7 @@ conjord *conjord_cria(char *nome);
16074 /**
16075  * insere elemento no conjunto ordenado
16076  */
16077 -void conjord_insererelemento(conjord *a, int chave, void *elemento);
16078 +int conjord_insererelemento(conjord *a, int chave, void *elemento);
16079
16080 /**
16081  * insere elemento no conjunto ordenado
16082  */
16083 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
16084 index f895ca2..b2da5ea 100644
16085 --- a/src/testa_item_1.c
16086 +++ b/src/testa_item_1.c
16087 @@ -1,5 +1,6 @@
16088 #include <stdio.h>
16089 #include <string.h>
16090 #include <time.h>
16091 #include "conjunto_ordenado.h"
16092 #include "testa_item_1.h"
16093
16094 @@ -10,6 +11,28 @@ typedef struct {
16095     char *telefone;
16096 } meu_tipo;
16097
16098 +static void adiciona_lista_de_elementos(conjord *c)
16099 +{
16100     int i;
16101     for (i = 0; i < 20;)
16102     {
16103         srand(time(NULL));
16104         meu_tipo *mt0 = (meu_tipo *) malloc(sizeof(meu_tipo));
16105         mt0->nome = (char *) malloc(sizeof("nome para teste1") + 1);
16106         strcpy(mt0->nome, "nome para teste1", sizeof("nome para teste1") + 1);
16107         mt0->telefone = (char *) malloc(sizeof("999999999") + 1);
16108         strcpy(mt0->telefone, "999999999", sizeof("999999999") + 1);
16109         int chave = rand() % 99 + 1;
16110         if (!conjord_insererelemento(c, chave, mt0))
16111             ++i;
16112         printf("chave:%d \t", chave);
16113         printf("nome:%s\t", mt0->nome);
16114         printf("telefone:%s\n", mt0->telefone);
16115     }
16116 }

```

```

16115 +}
16116 +
16117 void testa_item_1(const char *m)
16118 {
16119     /**
16120     @@ -31,29 +54,13 @@ void testa_item_1(const char *m)
16121     conjord_seta_imprime_elemento(imprime_meu_tipo);
16122
16123     conjord *cj_a = conjord_cria("A");
16124     conjord_insere_elemento(cj_a, 10, &mt0);
16125     // LTELA;
16126     conjord_imprime_arvore_rb(cj_a, 1);
16127     // CM;
16128
16129     meu_tipo mt1;
16130     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
16131     strncpy(mt1.nome, "Gertrudes", sizeof("Gertrudes") + 1);
16132     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
16133     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
16134
16135     conjord_insere_elemento(cj_a, 11, &mt1);
16136     // LTELA;
16137     conjord_imprime_arvore_rb(cj_a, 1);
16138     // CM;
16139
16140     LTELA;
16141     conjord_imprime_elemento(&mt1);
16142     adiciona_lista_de_elementos(cj_a);
16143     CM;
16144
16145     LTELA;
16146     conjord_imprime_elemento(&mt0);
16147     CM;
16148     conjord_imprime_conjunto(cj_a);
16149     CM;
16150 }
16151
16152 void tela_item_1(const char *m)
16153 @@ -66,8 +73,8 @@ void tela_item_1(const char *m)
16154
16155 void imprime_meu_tipo(void *n)
16156 {
16157     printf("Nome: %s | ", ((meu_tipo *)n)->nome);
16158     printf("Telefone: %s", ((meu_tipo *)n)->telefone);
16159     printf("Nome: %s \t", ((meu_tipo *)n)->nome);
16160     printf("Telefone: %s\n", ((meu_tipo *)n)->telefone);
16161 }
16162
16163
16164
16165 commit 2f1baaa0aa0bc3607571a4cf50506504e9751db3
16166 Author: Joilnen <joilnen@gmail.com>
16167 Date: Sun Jan 2 10:41:00 2022 -0300
16168
16169     consertando a geracao randomica, seed soh precisa ser chamado uma vez
16170     por chamada de funcao
16171
16172 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
16173 index b2da5ea..ad97d29 100644
16174 --- a/src/testa_item_1.c
16175 +++ b/src/testa_item_1.c
16176 @@ -14,9 +14,9 @@ typedef struct {
16177 static void adiciona_lista_de_elementos(conjord *c)
16178 {
16179     int i;
16180     srand(time(NULL));
16181     for (i = 0; i < 20; )
16182     {
16183         srand(time(NULL));
16184         meu_tipo *mt0 = (meu_tipo *) malloc(sizeof(meu_tipo));
16185         mt0->nome = (char *) malloc(sizeof("nome para teste1") + 1);
16186         strncpy(mt0->nome, "nome para teste1", sizeof("nome para teste1") + 1);
16187
16188 commit 010bf3164df357a2de71aedba35af18529277b2c
16189 Author: Joilnen <joilnen@gmail.com>
16190 Date: Sun Jan 2 12:28:00 2022 -0300
16191
16192     testes dos itens 1.1 2.1 e 3
16193
16194 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
16195 index 1b90579..4befff1 100644
16196 --- a/src/conjunto_ordenado.h
16197 +++ b/src/conjunto_ordenado.h
16198 @@ -130,10 +130,11 @@ void conjord_imprime_conjunto(conjord *c);
16199 * varias a vezes eh preferivel q seja inline
16200 * mas como n tem funcao inline usei a macro
16201 * foi primeiro implementado com um loop
16202 * mas pensando melhor dessa forma mais rapido
16203 + * mas pensando melhor dessa forma eh mais rapido
16204 */
16205 #define LTELA printf(LIMPA_TELA)
16206 #define TIPO_DE_TERMINAL TCOLOR
16207 #define INFO(x) printf("%s%s%s", S_INV, S_INFO, x, S_NORM)
16208 #define CM printf("%s", m); fflush(stdout); getc(stdin)
16209
16210 #endif
16211 diff --git a/src/jcurses.h b/src/jcurses.h
16212 index 3a8e187..930eec5 100644
16213 --- a/src/jcurses.h
16214 +++ b/src/jcurses.h
16215 @@ -15,5 +15,6 @@
16216 #define S_VERM "\033[41m\033[37m"
16217 #define S_INV "\033[7m"
16218 #define S_NORM "\033[0m"
16219 #define S_INFO "\033[21;0H"
16220
16221 #endif
16222 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
16223 index ad97d29..5dbb8ab 100644
16224 --- a/src/testa_item_1.c
16225 +++ b/src/testa_item_1.c
16226 @@ -17,6 +17,13 @@ static void adiciona_lista_de_elementos(conjord *c)
16227     srand(time(NULL));
16228     for (i = 0; i < 20; )
16229     {
16230         /**

```

```

16231 +      * definicao de um tipo exemplo pelo o usuario
16232 +      */
16233 +
16234 +      /**
16235 +      * criacao do dado para o tipo do usuario
16236 +      */
16237 +      meu_tipo *mt0 = (meu_tipo *) malloc (sizeof(meu_tipo));
16238 +      mt0->nome = (char *) malloc(sizeof("nome para teste1" + 1);
16239 +      strncpy(mt0->nome, "nome para teste1", sizeof("nome para teste1" + 1);
16240 @@ -35,19 +42,6 @@ static void adiciona_lista_de_elementos(conjord *c)
16241 {
16242     void testa_item_1(const char *m)
16243     {
16244         /**
16245         * definicao de um tipo exemplo pelo o usuario
16246         */
16247         /**
16248         * criacao do dado para o tipo do usuario
16249         */
16250         meu_tipo mt0;
16251         mt0.nome = (char *) malloc(sizeof("nome para teste" + 1);
16252         strncpy(mt0.nome, "nome para teste", sizeof("nome para teste" + 1);
16253         mt0.telefone = (char *) malloc(sizeof("999999999" + 1);
16254         strncpy(mt0.telefone, "999999999", sizeof("999999999" + 1);
16255         /**
16256         * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
16257         */
16258 @@ -56,10 +50,17 @@ void testa_item_1(const char *m)
16259     conjord *cj_a = conjord_cria("A");
16260     LTEL;
16261     adiciona_lista_de_elementos(cj_a);
16262     INFO("item 2.1 insere 20 elem. chave unica 1 a 99");
16263     CM;
16264     LTEL;
16265     conjord_imprime_arvore_rb(cj_a, 1);
16266     INFO("item 3 arvore atual");
16267     CM;
16268     LTEL;
16269     conjord_imprime_conjunto(cj_a);
16270     INFO("item 1.1 crescente por chave de id.");
16271     CM;
16272     }
16273     }
16274     }
16275     }
16276     }
16277     }
16278     }
16279     }
16280     }
16281     }
16282     }
16283     }
16284     }
16285     }
16286     }
16287     }
16288     }
16289     }
16290     }
16291     }
16292     }
16293     }
16294     }
16295     }
16296     }
16297     }
16298     }
16299     }
16300     }
16301     }
16302     }
16303     }
16304     }
16305     }
16306     }
16307     }
16308     }
16309     }
16310     }
16311     }
16312     }
16313     }
16314     }
16315     }
16316     }
16317     }
16318     }
16319     }
16320     }
16321     }
16322     }
16323     }
16324     }
16325     }
16326     }
16327     }
16328     }
16329     }
16330     }
16331     }
16332     }
16333     }
16334     }
16335     }
16336     }
16337     }
16338     }
16339     }
16340     }
16341     }
16342     }
16343     }
16344     }
16345     }
16346     }

```

```

16347 |
16348 | #endif
16349 | diff --git a/src/jcurses.h b/src/jcurses.h
16350 | index 8616442..de499cd 100644
16351 | --- a/src/jcurses.h
16352 | +++ b/src/jcurses.h
16353 | @@ -16,5 +16,6 @@
16354 | #define S_INV "\033[7m"
16355 | #define S_NORM "\033[0m"
16356 | #define S_INFO "\033[24;0H"
16357 | #define S_INFO_ST "\033[0;65H"
16358 |
16359 | #endif
16360 | diff --git a/src/testa_item_1.c b/src/testa_item_1.c
16361 | index 5dbb8ab..271d50a 100644
16362 | --- a/src/testa_item_1.c
16363 | +++ b/src/testa_item_1.c
16364 | @@ -50,7 +50,7 @@ void testa_item_1(const char *m)
16365 |     conjord *cj_a = conjord_cria("A");
16366 |     LTELA;
16367 |     adiciona_lista_de_elementos(cj_a);
16368 |     INFO("* item 2.1 insere 20 elem. chave unica 1 a 99");
16369 | + INFO("* item 2.1 insere 20 elem. chave unica entre 1 a 99");
16370 |     CM;
16371 |
16372 |     LTELA;
16373 | @@ -62,6 +62,8 @@ void testa_item_1(const char *m)
16374 |     conjord_imprime_conjunto(cj_a);
16375 |     INFO("* item 1.1 crescente por chave de id.");
16376 |     CM;
16377 | +
16378 | +     conjord_destroi(cj_a);
16379 | }
16380 |
16381 | void tela_item_1(const char *m)
16382 |
16383 | commit b090880ff232c54f785a787d9dcb9a4f731f10d8
16384 | Author: Joilnen <joilnen@gmail.com>
16385 | Date: Mon Jan 3 06:32:32 2022 -0300
16386 |
16387 |     criado o mecanismo de checagem dos testes da avaliacao, muita operacao booleana
16388 |     interessante
16389 |
16390 | diff --git a/src/checklist.c b/src/checklist.c
16391 | new file mode 100644
16392 | index 0000000..189eb62
16393 | --- /dev/null
16394 | +++ b/src/checklist.c
16395 | @@ -0,0 +1,30 @@
16396 | #include <stdio.h>
16397 | #include <string.h>
16398 | #include "conjunto_ordenado.h"
16399 | #include "checklist.h"
16400 | +
16401 | +void tela_checklist(int i)
16402 | +{
16403 | +     int l = 5, c = 24, bo = 1UL;
16404 | +     static unsigned int cklst_estado = 0;
16405 | +     cklst_estado |= i;
16406 | +
16407 | +     CKLSITEM(l++, c + 11, 0, "avaliacao");
16408 | +     CKLSITEM(l++, c, 0, " ");
16409 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_1); bo <= 1;
16410 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_2); bo <= 1;
16411 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_3); bo <= 1;
16412 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_4); bo <= 1;
16413 | +     CKLSITEM(l++, c, 0, " ");
16414 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM2_1); bo <= 1;
16415 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM2_2); bo <= 1;
16416 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM2_3); bo <= 1;
16417 | +     CKLSITEM(l++, c, 0, " ");
16418 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM3_1); bo <= 1;
16419 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM3_2); bo <= 1;
16420 | +     CKLSITEM(l++, c, cklst_estado & bo, ITEM3_3); bo <= 1;
16421 | +     CKLSITEM(l++, c, 0, " ");
16422 | +     CKLSITEM(l, c, cklst_estado & bo, ITEM4);
16423 | +}
16424 | +
16425 | +
16426 | diff --git a/src/checklist.h b/src/checklist.h
16427 | new file mode 100644
16428 | index 0000000..fbc2506
16429 | --- /dev/null
16430 | +++ b/src/checklist.h
16431 | @@ -0,0 +1,26 @@
16432 | #ifndef CHECKLIST_H_
16433 | #define CHECKLIST_H_
16434 | +
16435 | +void tela_checklist(int i);
16436 | +
16437 | +#define ITEM1_1 "1.1 listar ordenado por chave"
16438 | +#define ITEM1_2 "1.2 cardinalidade do conjunto"
16439 | +#define ITEM1_3 "1.3 verificar se conj. vazio"
16440 | +#define ITEM1_4 "1.4 tornar o conjunto vazio"
16441 | +
16442 | +#define ITEM2_1 "2.1 inserir elemento no conjunto"
16443 | +#define ITEM2_2 "2.2 excluir elemento do conjunto"
16444 | +#define ITEM2_3 "2.3 verif. se elem. esta no conj."
16445 | +
16446 | +#define ITEM3_1 "3.1 itersecao entre conjuntos"
16447 | +#define ITEM3_2 "3.2 uniao entre conjuntos"
16448 | +#define ITEM3_3 "3.3 subtracao de conjuntos"
16449 | +
16450 | +#define ITEM4 "4 imprimir arvore rubro negra"
16451 | +
16452 | +enum {I1_1, I1_2, I1_3, I1_4, I2_1, I2_2, I2_3, \
16453 | +      I3_1, I3_2, I3_3, I4};
16454 | +
16455 | #endif
16456 | +
16457 | +
16458 | diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
16459 | index b94287e..cccae08 100644
16460 | --- a/src/conjunto_ordenado.h
16461 | +++ b/src/conjunto_ordenado.h
16462 | @@ -137,6 +137,9 @@ void conjord_imprime_conjunto(conjord *c);

```

```

16463 #define INFO(x) printf("%s%s%s", S_INV, S_INFO, x, S_NORM)
16464 #define INFO_ST(x) printf("%s%s%s", S_INV, S_INFO_ST, x, S_NORM)
16465 #define CM printf("%s", m); fflush(stdout); getc(stdin)
16466 #define CKLSITEM(l,c,b,x) \
16467 + if (b) printf(S_CHKLIST_DONE, l, c, x); \
16468 + else printf(S_CHKLIST, l, c, x)
16469
16470 #endif
16471
16472 diff --git a/src/jcurses.h b/src/jcurses.h
16473 index de499cd..a1245e4 100644
16474 --- a/src/jcurses.h
16475 +++ b/src/jcurses.h
16476 @@ -17,5 +17,7 @@
16477 #define S_NORM "\033[0m"
16478 #define S_INV "\033[24;0H"
16479 #define S_INFO_ST "\033[0;65H"
16480 #define S_CHKLIST "\033[%d;%dH%s\033[0m"
16481 #define S_CHKLIST_DONE "\033[7m\033[%d;%dH%s\033[0m"
16482
16483 #endif
16484 diff --git a/src/makefile b/src/makefile
16485 index 94af2b5..2ddd9d1 100644
16486 --- a/src/makefile
16487 +++ b/src/makefile
16488 @@ -4,9 +4,9 @@
16489
16490 C=gcc
16491 S=main.c conjunto_ordenado.c red_black.c testa_rb.c \
16492 - testa_item_1.c testa_item_2.c testa_item_3.c
16493 + testa_item_1.c testa_item_2.c testa_item_3.c checklist.c
16494 O=main.o conjunto_ordenado.o red_black.o testa_rb.o \
16495 - testa_item_1.o testa_item_2.o testa_item_3.o
16496 + testa_item_1.o testa_item_2.o testa_item_3.o checklist.o
16497 E=rodeme
16498 D=-g
16499
16500 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
16501 index 271d50a..0368fd7 100644
16502 --- a/src/testa_item_1.c
16503 +++ b/src/testa_item_1.c
16504 @@ -3,6 +3,7 @@
16505 #include <time.h>
16506 #include "conjunto_ordenado.h"
16507 #include "testa_item_1.h"
16508 #include "checklist.h"
16509
16510 static void imprime_meu_tipo(void *);
16511
16512 @@ -47,7 +48,15 @@ void testa_item_1(const char *m)
16513 */
16514 conjord_set_a_imprime_elemento(imprime_meu_tipo);
16515
16516 + LTELA;
16517 + tela_checklist(0);
16518 + CM;
16519 +
16520 conjord *cj_a = conjord_cria("A");
16521 + LTELA;
16522 + tela_checklist(1 << I1_1);
16523 + CM;
16524 +
16525 LTELA;
16526 adiciona_lista_de_elementos(cj_a);
16527 INFO("* item 2.1 insere 20 elem. chave unica entre 1 a 99");
16528 @@ -55,13 +64,16 @@ void testa_item_1(const char *m)
16529
16530 LTELA;
16531 conjord_imprime_arvore_rb(cj_a, 1);
16532 - INFO("* item 3 arvore atual");
16533 + INFO("* item 4 arvore atual");
16534 CM;
16535
16536 LTELA;
16537 conjord_imprime_conjunto(cj_a);
16538 INFO("* item 1.1 crescente por chave de id.");
16539 CM;
16540 + LTELA;
16541 + tela_checklist((1 << I4));
16542 + CM;
16543
16544 conjord_destroi(cj_a);
16545 }
16546
16547 commit c8ea0e7d23cf1512d5904a92c9b1b3829c266051
16548 Author: Joilnen <joilnen@gmail.com>
16549 Date: Mon Jan 3 07:30:00 2022 -0300
16550
16551 melhorando a intercalacao das telas e as mensagens
16552
16553 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
16554 index ccca08..557b131 100644
16555 --- a/src/conjunto_ordenado.h
16556 +++ b/src/conjunto_ordenado.h
16557 @@ -134,7 +134,8 @@ void conjord_imprime_conjunto(conjord *c);
16558 */
16559 #define LTELA printf(LIMPA_TELA)
16560 #define TIPO_DE_TERMINAL TCOLOR
16561 #define INFO(x) printf("%s%s%s", S_INV, S_INFO, x, S_NORM)
16562 #define INFO2(x) printf("%s%s%s", S_INV2, x, S_NORM)
16563 #define INFO_ST(x) printf("%s%s%s", S_INV, S_INFO_ST, x, S_NORM)
16564 #define CM printf("%s", m); fflush(stdout); getc(stdin)
16565 #define CKLSITEM(l,c,b,x) \
16566 diff --git a/src/jcurses.h b/src/jcurses.h
16567 index a1245e4..212a524 100644
16568 --- a/src/jcurses.h
16569 +++ b/src/jcurses.h
16570 @@ -18,6 +18,7 @@
16571 #define S_INV "\033[24;0H"
16572 #define S_INFO_ST "\033[0;65H"
16573 #define S_CHKLIST "\033[%d;%dH%s\033[0m"
16574 #define S_CHKLIST_DONE "\033[7m\033[%d;%dH%s\033[0m"
16575 #define S_CHKLIST_DONE "\033[46m\033[30m\033[%d;%dH%s\033[0m"
16576 #define S_INFO2 "\033[46m\033[30m\033[24;0H"
16577

```



```

16579 #endif
16580 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
16581 index 0368fd7..ad8efc5 100644
16582 --- a/src/testa_item_1.c
16583 +++ b/src/testa_item_1.c
16584 @@ -53,26 +53,30 @@ void testa_item_1(const char *m)
16585     CM;
16586
16587     conjord *cj_a = conjord_cria("A");
16588     LTELA;
16589     tela_checklist(1 << I1_1);
16590     CM;
16591
16592     LTELA;
16593     adiciona_lista_de_elementos(cj_a);
16594     INFO(" * item 2.1 insere 20 elem. chave unica entre 1 a 99");
16595     CM;
16596     LTELA;
16597     tela_checklist(1 << I2_1);
16598     CM;
16599
16600     LTELA;
16601     conjord_imprime_arvore_rb(cj_a, 1);
16602     INFO(" * item 4 arvore atual");
16603     CM;
16604
16605     LTELA;
16606     tela_checklist((1 << I4));
16607     CM;
16608
16609     LTELA;
16610     conjord_imprime_conjunto(cj_a);
16611     INFO(" * item 1.1 crescente por chave de id.");
16612     CM;
16613
16614     LTELA;
16615     tela_checklist((1 << I4));
16616     tela_checklist((1 << I1_1));
16617     CM;
16618
16619     conjord_destroi(cj_a);
16620
16621 commit 350d45fbce1de5c007d3e6d9b390c4c5e966a7d1
16622 Author: Joilnen <joilnen@gmail.com>
16623 Date: Mon Jan 3 12:48:39 2022 -0300
16624
16625     decrementando cardinalidade na delecao de elemento do conjunto
16626
16627 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
16628 index 2a2ae6a..818e71d 100644
16629 --- a/src/conjunto_ordenado.c
16630 +++ b/src/conjunto_ordenado.c
16631 @@ -59,6 +59,9 @@ int conjord_insere_elemento(conjord *a, int chave, void *elemento)
16632 void conjord_deleta_elemento(conjord *a, int chave)
16633 {
16634     rb_delete(&a->pr, rb_search(a->pr, chave));
16635     a->cardinalidade--;
16636     if(a->cardinalidade < 0)
16637         a->cardinalidade = 0;
16638 }
16639
16640 int conjord_cardinalidade(conjord *c)
16641
16642 commit 940e73df5354e8b438e0d5020e21246c429ea148
16643 Author: Joilnen <joilnen@gmail.com>
16644 Date: Mon Jan 3 13:06:56 2022 -0300
16645
16646     removendo funcoes que n foram usadas
16647
16648 diff --git a/src/red_black.c b/src/red_black.c
16649 index 5cd3def..f97445b 100644
16650 --- a/src/red_black.c
16651 +++ b/src/red_black.c
16652 @@ -307,11 +307,6 @@ int rb_delete(rb_n **t, rb_n *z)
16653     return 0;
16654 }
16655
16656 -int rb_exclui_elem(rb_n *a, int k)
16657 -{
16658     return 0;
16659 -}
16660
16661 rb_n* rb_intersec(rb_n *a, rb_n *b)
16662 {
16663     return NULL;
16664 }
16665 diff --git a/src/red_black.h b/src/red_black.h
16666 index 27ff6bc..98705dc 100644
16667 --- a/src/red_black.h
16668 +++ b/src/red_black.h
16669 @@ -86,12 +86,6 @@ rb_n *rb_search(rb_n *t, int k);
16670 */
16671 int rb_delete(rb_n **t, rb_n *z);
16672
16673 /**
16674  * @param a noh raiz da arvore
16675  * @param k chave do elemento a ser excluido
16676  */
16677 -int rb_exclui_elem(rb_n *a, int k);
16678
16679 /**
16680  * @param a noh raiz da arvore
16681  * @param k chave do elemento a ser checado
16682  */
16683 commit c3bd058cfe0ab65b6cde02f11b92ccbb1b69a65f
16684 Author: Joilnen <joilnen@gmail.com>
16685 Date: Mon Jan 3 14:53:07 2022 -0300
16686
16687     adicionando arquivos para documentacao
16688
16689 diff --git a/doc/tex/code/del.c b/doc/tex/code/del.c
16690 new file mode 100644
16691 index 0000000..6af7bfe
16692 --- /dev/null
16693 +++ b/doc/tex/code/del.c
16694 @@ -0,0 +1,121 @@
16695 +/**

```

```

16695 + * deleta noh da arvore red black
16696 + * @param t ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao
16697 + * e manter a alteracao depois do retorno da funcao
16698 + * @param z ponteiro para o noh a ser deletado, para deletar por chave o TAD deve
16699 + * passar a saida da funcao pesquisar, num arranjo funcional na chamada
16700 + */
16701 +int rb_delete(rb_n **t, rb_n *z)
16702 +{
16703 +    rb_n *x = &rb_s;
16704 +    rb_n *y = z;
16705 +    unsigned char yoc = z->c;
16706 +
16707 +    if (z->l == &rb_s)
16708 +    {
16709 +        x = z->r;
16710 +        rb_transplant(t, z, z->r);
16711 +    }
16712 +    else if (z->r == &rb_s)
16713 +    {
16714 +        x = z->l;
16715 +        rb_transplant(t, z, z->l);
16716 +    }
16717 +    else
16718 +    {
16719 +        // y = tree_minimum(z->r);
16720 +        y = sucessor(z);
16721 +        yoc = y->c;
16722 +        x = y->r;
16723 +        if (y->p == z)
16724 +            x->p = y;
16725 +        else
16726 +        {
16727 +            rb_transplant(t, y, y->r);
16728 +            y->r = z->r;
16729 +            y->r->p = y;
16730 +        }
16731 +        rb_transplant(t, z, y);
16732 +        y->l = z->l;
16733 +        y->l->p = y;
16734 +        y->c = z->c;
16735 +    }
16736 +
16737 +    if (yoc == BLACK)
16738 +    {
16739 +        rb_n *w = &rb_s;
16740 +        while (x != *t && x->c == BLACK)
16741 +        {
16742 +            if (x == x->p->l)
16743 +            {
16744 +                w = x->p->r;
16745 +                if (w->c == RED)
16746 +                {
16747 +                    w->c = BLACK;
16748 +                    x->p->c = RED;
16749 +                    rb_rot_esq(t, x->p);
16750 +                    w = x->p->r;
16751 +                }
16752 +
16753 +                if (w->l->c == BLACK && w->r->c == BLACK)
16754 +                {
16755 +                    w->c = RED;
16756 +                    x = x->p;
16757 +                }
16758 +                else
16759 +                {
16760 +                    if (w->r->c == BLACK)
16761 +                    {
16762 +                        w->l->c = BLACK;
16763 +                        w->c = RED;
16764 +                        rb_rot_dir(t, w);
16765 +                        w = x->p->r;
16766 +                    }
16767 +
16768 +                    w->c = x->p->c;
16769 +                    x->p->c = BLACK;
16770 +                    w->r->c = BLACK;
16771 +                    rb_rot_esq(t, x->p);
16772 +                    x = *t;
16773 +                }
16774 +            }
16775 +            else
16776 +            {
16777 +                w = x->p->l;
16778 +                if (w->c == RED)
16779 +                {
16780 +                    w->c = BLACK;
16781 +                    x->p->c = RED;
16782 +                    rb_rot_dir(t, x->p);
16783 +                    w = x->p->l;
16784 +                }
16785 +
16786 +                if (w->l->c == BLACK && w->r->c == BLACK)
16787 +                {
16788 +                    w->c = RED;
16789 +                    x = x->p;
16790 +                }
16791 +                else
16792 +                {
16793 +                    if (w->l->c == BLACK)
16794 +                    {
16795 +                        w->r->c = BLACK;
16796 +                        w->c = RED;
16797 +                        rb_rot_esq(t, w);
16798 +                        w = x->p->l;
16799 +                    }
16800 +
16801 +                    w->c = x->p->c;
16802 +                    x->p->c = BLACK;
16803 +                    w->l->c = BLACK;
16804 +                    rb_rot_dir(t, x->p);
16805 +                    x = *t;
16806 +                }
16807 +            }
16808 +        }
16809 +        x->c = BLACK;
16810 +    } /* if (yoc == BLACK) */

```

```

16811 +
16812 +     return 0;
16813 +}
16814 +
16815 diff --git a/doc/tex/code/inc.c b/doc/tex/code/inc.c
16816 new file mode 100644
16817 index 0000000..63e50ca
16818 --- /dev/null
16819 +++ b/doc/tex/code/inc.c
16820 @@ -0,0 +1,103 @@
16821 +/**
16822 + * inclusao baseado no livro do cormen
16823 + * e nas notas de aula da prof. Luciana
16824 + */
16825 +int rb_insert(rb_n **t, rb_n *z)
16826 +{
16827 +    /**
16828 +     * y receberah noh pai imediato do noh adicionado
16829 +     * x eh um ponteiro temporario para iteracao feita
16830 +     * no loop while a seguir para chegar ao noh folha
16831 +     */
16832 +    rb_n *y = &rb_s;
16833 +    rb_n *x = *t;
16834 +
16835 +    /**
16836 +     * itera para chegar no noh folha
16837 +     * da posicao a ser inserido o novo noh
16838 +     */
16839 +    while (x != &rb_s)
16840 +    {
16841 +        y = x;
16842 +        if (z->k < x->k)
16843 +            x = x->l;
16844 +        else if (z->k > x->k)
16845 +            x = x->r;
16846 +        else
16847 +            return -1;
16848 +    }
16849 +
16850 +    /**
16851 +     * seta noh pai para o noh adicionado
16852 +     */
16853 +    z->p = y;
16854 +
16855 +    /**
16856 +     * se esse noh adicionado eh unico y permance
16857 +     * nulo que neste caso apontaria para o pai que eh nulo
16858 +     * neste cenario torna o noh adicionado raiz
16859 +     * caso n adiciona na esquerda ou direita de acordo
16860 +     * com o valor da chave, noh recebe RED como padrao
16861 +     */
16862 +    if (y == &rb_s)
16863 +        *t = z;
16864 +    else if (z->k < y->k)
16865 +        y->l = z;
16866 +    else
16867 +        y->r = z;
16868 +    z->l = z->r = &rb_s;
16869 +    z->c = RED;
16870 +
16871 +    /** ajuste */
16872 +    while (z->p->c == RED)
16873 +    {
16874 +        if (z->p == z->p->p->l)
16875 +        {
16876 +            y = z->p->p->r;
16877 +            if (y->c == RED)
16878 +            {
16879 +                z->p->c = BLACK;
16880 +                y->c = BLACK;
16881 +                z->p->p->c = RED;
16882 +                z = z->p->p;
16883 +            }
16884 +            else
16885 +            {
16886 +                if (z == z->p->r)
16887 +                {
16888 +                    z = z->p;
16889 +                    rb_rot_esq(t, z);
16890 +                }
16891 +                z->p->c = BLACK;
16892 +                z->p->p->c = RED;
16893 +                rb_rot_dir(t, z->p->p);
16894 +            }
16895 +        }
16896 +        else
16897 +        {
16898 +            y = z->p->p->l;
16899 +            if (y->c == RED)
16900 +            {
16901 +                z->p->c = BLACK;
16902 +                y->c = BLACK;
16903 +                z->p->p->c = RED;
16904 +                z = z->p->p;
16905 +            }
16906 +            else
16907 +            {
16908 +                if (z == z->p->l)
16909 +                {
16910 +                    z = z->p;
16911 +                    rb_rot_dir(t, z);
16912 +                }
16913 +                z->p->c = BLACK;
16914 +                z->p->p->c = RED;
16915 +                rb_rot_esq(t, z->p->p);
16916 +            }
16917 +        }
16918 +    }
16919 +    (*t)->c = BLACK;
16920 +    return 0;
16921 +}
16922 +}
16923 +
16924 diff --git a/doc/tex/code/incdel.c b/doc/tex/code/incdel.c
16925 new file mode 100644
16926 index 0000000..f49e5b6

```

```

16927 | --- /dev/null
16928 | +++ b/doc/tex/code/incdel.c
16929 | @@ -0,0 +1,83 @@
16930 | +/**
16931 | + * funcao que faz trocas auxiliares na delecao
16932 | + * @param u noh que eh trocado por outro em seu lugar
16933 | + * @param v noh que eh colocado no lugar do noh anterior
16934 | + */
16935 | +static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
16936 | +{
16937 | +    if (u->p == &rb_s)
16938 | +        *t = v;
16939 | +    else if (u == u->p->l)
16940 | +        u->p->l = v;
16941 | +    else
16942 | +        u->p->r = v;
16943 | +    if(v)
16944 | +        v->p = u->p;
16945 | +}
16946 | +
16947 | +/**
16948 | + * @param x noh raiz
16949 | + * retorna o noh mais a esquerda na arvore
16950 | + * ou subarvore
16951 | + */
16952 | +static rb_n *tree_minimum(rb_n *x)
16953 | +{
16954 | +    rb_n *a = x;
16955 | +    while (a->l != &rb_s)
16956 | +        a = a->l;
16957 | +    return a;
16958 | +}
16959 | +
16960 | +/**
16961 | + * sucessor baseado nas notas e no livro do cormen, reescrito
16962 | + * e consegui diminuir em duas linhas do algoritmo original
16963 | + */
16964 | +static rb_n* sucessor(rb_n *x)
16965 | +{
16966 | +    rb_n *y = NULL;
16967 | +    if (x->r != &rb_s)
16968 | +        return tree_minimum(x->r);
16969 | +    while ((y = x->p) && x == y->r)
16970 | +        x = y;
16971 | +    return y;
16972 | +}
16973 | +
16974 | +
16975 | +int rb_rot_esq(rb_n **t, rb_n *x)
16976 | +{
16977 | +    rb_n *y = x->r;
16978 | +    x->r = y->l;
16979 | +    if (y->l != &rb_s)
16980 | +        y->l->p = x;
16981 | +    y->p = x->p;
16982 | +    if (x->p == &rb_s)
16983 | +        *t = y;
16984 | +    else if (x == x->p->l)
16985 | +        x->p->l = y;
16986 | +    else
16987 | +        x->p->r = y;
16988 | +    y->l = x;
16989 | +    x->p = y;
16990 | +
16991 | +    return 0;
16992 | +}
16993 | +
16994 | +int rb_rot_dir(rb_n **t, rb_n *x)
16995 | +{
16996 | +    rb_n *y = x->l;
16997 | +    x->l = y->r;
16998 | +    if (y->r != &rb_s)
16999 | +        y->r->p = x;
17000 | +    y->p = x->p;
17001 | +    if (x->p == &rb_s)
17002 | +        *t = y;
17003 | +    else if (x == x->p->r)
17004 | +        x->p->r = y;
17005 | +    else
17006 | +        x->p->l = y;
17007 | +    y->r = x;
17008 | +    x->p = y;
17009 | +
17010 | +    return 0;
17011 | +}
17012 | +
17013 | diff --git a/doc/tex/code/utils.c b/doc/tex/code/utils.c
17014 | new file mode 100644
17015 | index 0000000..ae8b953
17016 | --- /dev/null
17017 | +++ b/doc/tex/code/utils.c
17018 | @@ -0,0 +1,84 @@
17019 | +/**
17020 | + * funcao que faz trocas auxiliares na delecao
17021 | + * @param u noh que eh trocado por outro em seu lugar
17022 | + * @param v noh que eh colocado no lugar do noh anterior
17023 | + */
17024 | +static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
17025 | +{
17026 | +    if (u->p == &rb_s)
17027 | +        *t = v;
17028 | +    else if (u == u->p->l)
17029 | +        u->p->l = v;
17030 | +    else
17031 | +        u->p->r = v;
17032 | +    if(v)
17033 | +        v->p = u->p;
17034 | +    return 0;
17035 | +}
17036 | +
17037 | +/**
17038 | + * @param x noh raiz
17039 | + * retorna o noh mais a esquerda na arvore
17040 | + * ou subarvore
17041 | + */
17042 | +static rb_n *tree_minimum(rb_n *x)

```

```

17043 | +{
17044 | +   rb_n *a = x;
17045 | +   while (a->l != &rb_s)
17046 | +       a = a->l;
17047 | +   return a;
17048 | +}
17049 | +
17050 | +/**
17051 | + * sucessor baseado nas notas e no livro do cormen, reescrito
17052 | + * e consegui diminuir em duas linhas do algoritmo original
17053 | + */
17054 | +static rb_n* sucessor(rb_n *x)
17055 | +{
17056 | +   rb_n *y = NULL;
17057 | +   if (x->r != &rb_s)
17058 | +       return tree_minimum(x->r);
17059 | +   while ((y = x->p) && x == y->r)
17060 | +       x = y;
17061 | +   return y;
17062 | +}
17063 | +
17064 | +
17065 | +int rb_rot_esq(rb_n **t, rb_n *x)
17066 | +{
17067 | +   rb_n *y = x->r;
17068 | +   x->r = y->l;
17069 | +   if (y->l != &rb_s)
17070 | +       y->l->p = x;
17071 | +   y->p = x->p;
17072 | +   if (x->p == &rb_s)
17073 | +       *t = y;
17074 | +   else if (x == x->p->l)
17075 | +       x->p->l = y;
17076 | +   else
17077 | +       x->p->r = y;
17078 | +   y->l = x;
17079 | +   x->p = y;
17080 | +   return 0;
17081 | +}
17082 | +
17083 | +
17084 | +int rb_rot_dir(rb_n **t, rb_n *x)
17085 | +{
17086 | +   rb_n *y = x->l;
17087 | +   x->l = y->r;
17088 | +   if (y->r != &rb_s)
17089 | +       y->r->p = x;
17090 | +   y->p = x->p;
17091 | +   if (x->p == &rb_s)
17092 | +       *t = y;
17093 | +   else if (x == x->p->r)
17094 | +       x->p->r = y;
17095 | +   else
17096 | +       x->p->l = y;
17097 | +   y->r = x;
17098 | +   x->p = y;
17099 | +   return 0;
17100 | +}
17101 | +
17102 | +
17103 | +diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
17104 | +index ef705d1..6145a9e 100644
17105 | +Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
17106 | +diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
17107 | +index 9c07db1..7f65f48 100644
17108 | +--- a/doc/tex/relatorio.tex
17109 | ++++ b/doc/tex/relatorio.tex
17110 | +@@ -147,7 +147,6 @@ com \enf{insere_elemento}
17111 | + * @rb_n noh red black
17112 | + * @k chave (k de key)
17113 | + * @c cor (c de color ou cor, 0-black, 1-red)
17114 | + * @t tipo (t de tipo ou type)
17115 | + * @d dado (de de data ou dados)
17116 | + * @l @r @p l-eft filho a esquerda, r-right filho a direita e p-parent, noh pai
17117 | + */
17118 | +@@ -200,9 +199,16 @@ int rotaciona_direita(rb_n *r, rb_n *w);
17119 | +
17120 | +void (*imprime_elemento)(rb_n *);
17121 | +int seta_imprime_elemento(void (*i)(rb_n *n));
17122 | +
17123 | +
17124 | +\end{lstlisting}
17125 | +
17126 | +-\begin{figure}
17127 | +\lstinputlisting [language=c] {code/inc.c}
17128 | +\lstinputlisting [language=c] {code/del.c}
17129 | +\lstinputlisting [language=c] {code/utills.c}
17130 | +
17131 | +\section{TAD}
17132 | +\begin{figure}[H]
17133 | +\begin{center}
17134 | +\includegraphics [scale = .4] {figs/ADT}
17135 | +\caption{Diagrama TAD}
17136 | +diff --git a/src/red_black.c b/src/red_black.c
17137 | +index f97445b..447027f 100644
17138 | +--- a/src/red_black.c
17139 | ++++ b/src/red_black.c
17140 | +@@ -157,6 +157,8 @@ static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
17141 | +   u->p->r = v;
17142 | +   if (v)
17143 | +       v->p = u->p;
17144 | +   return 0;
17145 | +}
17146 | +
17147 | +
17148 | +/**
17149 | +
17150 | +commit 606e835804dbc1341a9aa23b5432009a907b80b9
17151 | +Author: Joilnen <joilnen@gmail.com>
17152 | +Date: Mon Jan 3 14:53:53 2022 -0300
17153 | +
17154 | +removendo arquivo nao mais usado
17155 | +
17156 | +diff --git a/doc/tex/code/incdel.c b/doc/tex/code/incdel.c
17157 | +deleted file mode 100644
17158 | +index f49e5b6..0000000

```

```

17159 --- a/doc/tex/code/incdel.c
17160 +++ /dev/null
17161 @@ -1,83 +0,0 @@
17162 -/**
17163  * funcao que faz trocas auxiliares na delecao
17164  * @param u noh que eh trocado por outro em seu lugar
17165  * @param v noh que eh colocado no lugar do noh anterior
17166  */
17167 -static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
17168 -{
17169     if (u->p == &rb_s)
17170         *t = v;
17171     else if (u == u->p->l)
17172         u->p->l = v;
17173     else
17174         u->p->r = v;
17175     if (v)
17176         v->p = u->p;
17177 -}
17178 -
17179 -/**
17180  * @param x noh raiz
17181  * retorma o noh mais a esquerda na arvore
17182  * ou subarvore
17183  */
17184 -static rb_n *tree_minimum(rb_n *x)
17185 -{
17186     rb_n *a = x;
17187     while (a->l != &rb_s)
17188         a = a->l;
17189     return a;
17190 -}
17191 -
17192 -/**
17193  * sucessor baseado nas notas e no livro do cormen, reescrito
17194  * e consegui diminuir em duas linhas do algoritmo original
17195  */
17196 -static rb_n* sucessor(rb_n *x)
17197 -{
17198     rb_n *y = NULL;
17199     if (x->r != &rb_s)
17200         return tree_minimum(x->r);
17201     while ((y = x->p) && x == y->r)
17202         x = y;
17203     return y;
17204 -}
17205 -
17206 -int rb_rot_esq(rb_n **t, rb_n *x)
17207 -{
17208     rb_n *y = x->r;
17209     x->r = y->l;
17210     if (y->l != &rb_s)
17211         y->l->p = x;
17212     y->p = x->p;
17213     if (x->p == &rb_s)
17214         *t = y;
17215     else if (x == x->p->l)
17216         x->p->l = y;
17217     else
17218         x->p->r = y;
17219     y->l = x;
17220     x->p = y;
17221 -}
17222 -
17223 -return 0;
17224 -}
17225 -
17226 -int rb_rot_dir(rb_n **t, rb_n *x)
17227 -{
17228     rb_n *y = x->l;
17229     x->l = y->r;
17230     if (y->r != &rb_s)
17231         y->r->p = x;
17232     y->p = x->p;
17233     if (x->p == &rb_s)
17234         *t = y;
17235     else if (x == x->p->r)
17236         x->p->r = y;
17237     else
17238         x->p->l = y;
17239     y->r = x;
17240     x->p = y;
17241 -}
17242 -
17243 -return 0;
17244 -}
17245 -
17246 commit f92cf9336c85bcbdf8ceb1d29de72be3eeb8249e
17247 Author: Joilnen <joilnen@gmail.com>
17248 Date: Mon Jan 3 16:16:01 2022 -0300
17249
17250     atualizando documentacao
17251
17252 diff --git a/doc/tex/code/inc.c b/doc/tex/code/inc.c
17253 index 63e50ca..96c00d1 100644
17254 --- a/doc/tex/code/inc.c
17255 +++ b/doc/tex/code/inc.c
17256 @@ -1,20 +1,25 @@
17257 -/**
17258  * inclusao baseado no livro do cormen
17259  * e nas notas de aula da prof. Luciana
17260  * inclusao baseado nas notas de aula da prof. Luciana
17261  */
17262 -int rb_insert(rb_n **t, rb_n *z)
17263 -{
17264     /**
17265      * y receberah noh pai imediato do noh adicionado
17266      * x eh um ponteiro temporario para iteracao feita
17267      * no loop while a seguir para chegar ao noh folha
17268      * y aponta para nó sentinela, rb_s (nó sentinela)
17269      * x é um ponteiro temporário para iteração feita
17270      * no loop while mas tem como valor inicial a raiz
17271      * passada em t que é ponteiro duplo por que, poderá
17272      * ser alterada nesta função e desejamos que estas alterações
17273      * persistam
17274      */

```

```

17275     rb_n *y = &rb_s;
17276     rb_n *x = *t;
17277
17278     /**
17279     -   * itera para chegar no noh folha
17280     -   * da posicao a ser inserido o novo noh
17281     +   * itera para chegar no nó folha da posicao a ser inserido o novo noh,
17282     +   * como está explicito, caso a chave seja menor que a do nó testado atualiza a referencia x
17283     +   * com o nó a esquerda caso seja maior com o nó a direita, em caso diferente destes
17284     +   * retorna a função com código de erro, seria o caso em que já existe chave igual
17285     +   * na árvore
17286     */
17287     while (x != &rb_s)
17288     {
17289     diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
17290     index 6145a9e..fa90c6d 100644
17291     Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
17292     diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
17293     index 7f65f48..fe33786 100644
17294     --- a/doc/tex/relatorio.tex
17295     +++ b/doc/tex/relatorio.tex
17296     @@ -7,10 +7,11 @@
17297
17298     % \fancyhead{}
17299     % \fancyfoot{}
17300     -\usepackage{listings}
17301     +\usepackage{listingsutf8}
17302     \usepackage{xcolor}
17303
17304     -\definecolor{codegreen}{rgb}{0,0.6,0}
17305     +% \definecolor{codegreen}{rgb}{0,0.6,0}
17306     +\definecolor{codegreen}{rgb}{.6,.3,.0}
17307     \definecolor{codegray}{rgb}{0.5,0.5,0.5}
17308     \definecolor{codepurple}{rgb}{0.58,0,0.82}
17309     \definecolor{backcolour}{rgb}{0.95,0.95,0.92}
17310     @@ -35,7 +36,12 @@
17311     }
17312     }
17313
17314     -\lstset{style=mystyle}
17315     +\lstset{
17316     +   style=mystyle,
17317     +   language=C,
17318     +   inputencoding=utf8,
17319     +   texcl=true
17320     +}
17321
17322     % \renewcommand{\headrulewidth}{0pt} \renewcommand{\footrulewidth}{0pt}
17323     % \fancyhead[L0, LE]{\thepage}
17324
17325     commit a0cf7950524a228bcb8782679de240207ca846d9
17326     Author: Joilnen <joilnen@gmail.com>
17327     Date: Mon Jan 3 17:25:43 2022 -0300
17328
17329     tela inicial de informacoes
17330
17331     diff --git a/src/testa_item_1.c b/src/testa_item_1.c
17332     index ad8efc5..b2c47b6 100644
17333     --- a/src/testa_item_1.c
17334     +++ b/src/testa_item_1.c
17335     @@ -85,11 +85,17 @@ void testa_item_1(const char *m)
17336     void testa_item_1(const char *m)
17337     {
17338     LTELA;
17339     -   printf("Sera efetuado testes do item 1\n");
17340     +   int l = 5, c = 24;
17341     +   CKLSITEM(l++, c, 0, "serah efetuado os testes:");
17342     +   CKLSITEM(l++, c, 0, "inserir de elementos no conjunto");
17343     +   CKLSITEM(l++, c, 0, "com chaves randomicas, mostra do");
17344     +   CKLSITEM(l++, c, 0, "estado da arvore red black");
17345     +   CKLSITEM(l++, c, 0, "e ordenacao dos elementos do conjunto");
17346     +   CKLSITEM(l++, c, 0, "o tipo de dado do usuario aqui eh uma");
17347     +   CKLSITEM(l++, c, 0, "estrutura com duas strings, nome e telefone");
17348     CM;
17349     }
17350
17351     -
17352     void imprime_meu_tipo(void *n)
17353     {
17354     printf("Nome: %s \t", ((meu_tipo *)n)->nome);
17355
17356     commit 08ede9cad4fad62c5d02b318fefd968723b5260d
17357     Author: Joilnen <joilnen@gmail.com>
17358     Date: Tue Jan 4 02:31:34 2022 -0300
17359
17360     inicio do segundo lote de testes
17361
17362     diff --git a/src/testa_item_1.c b/src/testa_item_1.c
17363     index b2c47b6..ebf7d4f 100644
17364     --- a/src/testa_item_1.c
17365     +++ b/src/testa_item_1.c
17366     @@ -55,7 +55,7 @@ void testa_item_1(const char *m)
17367     conjord *cj_a = conjord_cria("A");
17368     LTELA;
17369     adiciona_lista_de_elementos(cj_a);
17370     -   INFO("item 2.1 insere 20 elem. chave unica entre 1 a 99");
17371     +   INFO("item 2.1 inserir 20 elem. chave unica entre 1 a 99");
17372     CM;
17373     LTELA;
17374     tela_checklist(1 << I2_1);
17375     @@ -63,7 +63,7 @@ void testa_item_1(const char *m)
17376
17377     LTELA;
17378     conjord_imprime_arvore_rb(cj_a, 1);
17379     -   INFO("item 4 arvore atual");
17380     +   INFO("item 4 mostrar arvore atual");
17381     CM;
17382
17383     LTELA;
17384     @@ -72,7 +72,7 @@ void testa_item_1(const char *m)
17385
17386     LTELA;
17387     conjord_imprime_conjunto(cj_a);
17388     -   INFO("item 1.1 crescente por chave de id.");
17389     +   INFO("item 1.1 ordenar por chave de identificacao");
17390     CM;

```

```

17391
17392     LTELTA;
17393 @@ -87,7 +87,7 @@ void tela_item_1(const char *m)
17394     LTELTA;
17395     int l = 5, c = 24;
17396     CKLSITEM(l++, c, 0, "serah efetuado os testes:");
17397 - CKLSITEM(l++, c, 0, "insercao de elementos no conjunto");
17398 + CKLSITEM(l++, c, 0, "insercao de elementos no conjunto");
17399     CKLSITEM(l++, c, 0, "com chaves randomicas, mostra do");
17400     CKLSITEM(l++, c, 0, "estado da arvore red black");
17401     CKLSITEM(l++, c, 0, "e ordenacao dos elementos do conjunto");
17402 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
17403 index 481f6e6..349d25e 100644
17404 --- a/src/testa_item_2.c
17405 +++ b/src/testa_item_2.c
17406 @@ -5,13 +5,26 @@
17407 void tela_item_2(const char *m)
17408 {
17409     LTELTA;
17410     printf("Testa item 2\n");
17411 +     int l = 5, c = 24, f;
17412 +     char mens[6][40] =
17413 +     {
17414 +         "serah efetuado os testes:",
17415 +         "exclusao de elemento do conjunto",
17416 +         "verif. se elemento pertence ao conjunto",
17417 +         "cardinalide do conjunto",
17418 +         "verif. se conjunto eh vazio",
17419 +         "tornar conjunto vazio"
17420 +     };
17421 +     for (f = 0; f < 7; f++)
17422 +     CKLSITEM(l++, c, 0, mens[f]);
17423     CM;
17424 }
17425
17426 void testa_item_2(const char *m)
17427 {
17428 -
17429 +     LTELTA;
17430 +     printf("Testa item 2\n");
17431 +     CM;
17432 }
17433
17434
17435
17436 commit 100022cf815464feeaf910c8b514f28fb9b62163
17437 Author: Joilnen <joilnen@gmail.com>
17438 Date: Tue Jan 4 02:59:12 2022 -0300
17439
17440     ressaltando algumas mensagens na apresentacao dos testes
17441
17442 diff --git a/src/jcurses.h b/src/jcurses.h
17443 index 212a524..cb6e417 100644
17444 --- a/src/jcurses.h
17445 +++ b/src/jcurses.h
17446 @@ -15,7 +15,7 @@
17447 #define S_VERM "\033[41m\033[37m"
17448 #define S_INV "\033[7m"
17449 #define S_NORM "\033[0m"
17450 -#define S_INFO "\033[24;0H"
17451 +#define S_INFO "\033[24;0H\033[7m"
17452 #define S_INFO_ST "\033[0;65H"
17453 #define S_CHKLIST "\033[%d;%dHs\033[0m"
17454 #define S_CHKLIST_DONE "\033[46m\033[30m\033[%d;%dHs\033[0m"
17455 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
17456 index ebf7d4f..90757c4 100644
17457 --- a/src/testa_item_1.c
17458 +++ b/src/testa_item_1.c
17459 @@ -86,7 +86,7 @@ void tela_item_1(const char *m)
17460 {
17461     LTELTA;
17462     int l = 5, c = 24;
17463 - CKLSITEM(l++, c, 0, "serah efetuado os testes:");
17464 + CKLSITEM(l++, c, 0, "\033[21mserah efetuado os testes:\033[0m");
17465     CKLSITEM(l++, c, 0, "insercao de elementos no conjunto");
17466     CKLSITEM(l++, c, 0, "com chaves randomicas, mostra do");
17467     CKLSITEM(l++, c, 0, "estado da arvore red black");
17468 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
17469 index 349d25e..80d0813 100644
17470 --- a/src/testa_item_2.c
17471 +++ b/src/testa_item_2.c
17472 @@ -8,7 +8,7 @@ void tela_item_2(const char *m)
17473     int l = 5, c = 24, f;
17474     char mens[6][40] =
17475     {
17476 -         "serah efetuado os testes:",
17477 +         "\033[21mserah efetuado os testes:\033[0m",
17478 +         "exclusao de elemento do conjunto",
17479 +         "verif. se elemento pertence ao conjunto",
17480 +         "cardinalide do conjunto",
17481     };
17482
17483 commit 372fbb71f972ced89f2440f43c73610d7e35a716
17484 Author: Joilnen <joilnen@gmail.com>
17485 Date: Tue Jan 4 03:25:11 2022 -0300
17486
17487     limpando o codigo de codigos escape, n eh elegante
17488
17489 diff --git a/src/jcurses.h b/src/jcurses.h
17490 index cb6e417..95c78df 100644
17491 --- a/src/jcurses.h
17492 +++ b/src/jcurses.h
17493 @@ -11,6 +11,8 @@
17494 */
17495 #define LIMPA_TELA "\033[H\033[J"
17496 #define S_CM "\033[24;62H"
17497 #define S_UNDERL "\033[21m"
17498 #define S_AZUL "\033[44m\033[37m"
17499 #define S_VERM "\033[41m\033[37m"
17500 #define S_INV "\033[7m"
17501 diff --git a/src/main.c b/src/main.c
17502 index 7d68692..9ee3774 100644
17503 --- a/src/main.c
17504 +++ b/src/main.c
17505 @@ -11,7 +11,7 @@ int main()
17506     /**

```



```

17507      * mensagem de controle
17508      */
17509 -   const char mc[] = "\033[24;62H* tecla enter * ";
17510 +   const char mc[] = S_CM"* tecla enter * ";
17511
17512      /**
17513       * aqui poderia ser escrito
17514 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
17515 index 90757c4..d60c52d 100644
17516 --- a/src/testa_item_1.c
17517 +++ b/src/testa_item_1.c
17518 @@ -86,7 +86,7 @@ void tela_item_1(const char *m)
17519 {
17520     LTELA;
17521     int l = 5, c = 24;
17522 -   CKLSITEM(1++, c, 0, "\033[21mserah efetuado os testes:\033[0m");
17523 +   CKLSITEM(1++, c, 0, S_UNDERL"serah efetuado os testes:"S_NORM);
17524 -   CKLSITEM(1++, c, 0, "insercao de elementos no conjunto");
17525 -   CKLSITEM(1++, c, 0, "com chaves randomicas, mostra do");
17526 -   CKLSITEM(1++, c, 0, "estado da arvore red black");
17527
17528 commit ba37dde91269497881bab9a995a59756bacb5452
17529 Author: Joilnen <joilnen@gmail.com>
17530 Date: Tue Jan 4 03:31:58 2022 -0300
17531
17532     removendo comentario que n eh mais necessario
17533
17534 diff --git a/src/main.c b/src/main.c
17535 index 9ee3774..a51b854 100644
17536 --- a/src/main.c
17537 +++ b/src/main.c
17538 @@ -13,17 +13,6 @@ int main()
17539     */
17540     const char mc[] = S_CM"* tecla enter * ";
17541
17542 -   /**
17543 -   * aqui poderia ser escrito
17544 -   * "case 0:
17545 -   *     tela_item_1();
17546 -   *     getc(stdin);
17547 -   *     testa_item_1();
17548 -   *     break;
17549 -   * "
17550 -   * em um unico swtich mas assim ficou mais legivel
17551 -   * e limpo
17552 -   */
17553     for (op = 0; op < nitems; ++op)
17554     {
17555         switch (op)
17556
17557 commit 5086ad523fa39de8a0f081f42cda914b2f41acc7
17558 Author: Joilnen <joilnen@gmail.com>
17559 Date: Tue Jan 4 03:41:43 2022 -0300
17560
17561     ainda limpando os codigos escapes
17562
17563 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
17564 index 818e71d..3a1d69f 100644
17565 --- a/src/conjunto_ordenado.c
17566 +++ b/src/conjunto_ordenado.c
17567 @@ -107,7 +107,7 @@ void conjord_imprime_lista(int *l, int s)
17568 {
17569     int i;
17570     for (i = 0; i < s; i++)
17571 -   printf("\033[%d;78H%d", l[i] + 1, l[i]);
17572 +   printf(S_LIST, l[i] + 1, l[i]);
17573 }
17574
17575 void (*conjord_imprime_elemento)(void *n);
17576 diff --git a/src/jcurses.h b/src/jcurses.h
17577 index 95c78df..22c0ae9 100644
17578 --- a/src/jcurses.h
17579 +++ b/src/jcurses.h
17580 @@ -22,5 +22,6 @@
17581 #define S_CHKLST "\033[%d;%dH%\033[0m"
17582 #define S_CHKLST_DONE "\033[46m\033[30m\033[%d;%dH%\033[0m"
17583 #define S_INF02 "\033[46m\033[30m\033[24;0H"
17584 #define S_LIST "\033[%d;78H%d"
17585
17586 #endif
17587 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
17588 index d60c52d..0b9b846 100644
17589 --- a/src/testa_item_1.c
17590 +++ b/src/testa_item_1.c
17591 @@ -72,7 +72,7 @@ void testa_item_1(const char *m)
17592 {
17593     LTELA;
17594     conjord_imprime_conjunto(cj_a);
17595 -   INFO(" * item 1.1 ordenar por chave de identificacao");
17596 +   INFO(" * item 1.1 ordenar por chave de identificacao");
17597     CM;
17598
17599     LTELA;
17600
17601 commit 0fdc05f41a92a0e5c25c94a58c7866071d81df92
17602 Author: Joilnen <joilnen@gmail.com>
17603 Date: Tue Jan 4 06:29:34 2022 -0300
17604
17605     adicionado autor no primeiro comentario
17606
17607 diff --git a/src/checklist.h b/src/checklist.h
17608 index fbe2506..730b616 100644
17609 --- a/src/checklist.h
17610 +++ b/src/checklist.h
17611 @@ -1,6 +1,10 @@
17612 #ifndef CHECKLIST_H_
17613 #define CHECKLIST_H_
17614
17615 /**
17616 * * aluno: joilnen leite
17617 * */
17618 +
17619 void tela_checklist(int i);
17620
17621 #define ITEM1_1 "1.1 listar ordenado por chave"
17622 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h

```

```

17623|index 557b131..ded9113 100644
17624|--- a/src/conjunto_ordenado.h
17625|+++ b/src/conjunto_ordenado.h
17626|@@ -1,6 +1,10 @@
17627|#ifndef CONJUNTO_ORDENADO_H
17628|#define CONJUNTO_ORDENADO_H
17629|
17630|+/*
17631|+ * aluno: joilnen leite
17632|+ */
17633|+
17634|#include "jurses.h"
17635|#include "red_black.h"
17636|
17637|diff --git a/src/jurses.h b/src/jurses.h
17638|index 22c0ae9..19af7b9 100644
17639|--- a/src/jurses.h
17640|+++ b/src/jurses.h
17641|@@ -1,6 +1,10 @@
17642|#ifndef JOILNEN_CURSES_H
17643|#define JOILNEN_CURSES_H
17644|
17645|+/*
17646|+ * aluno: joilnen
17647|+ */
17648|+
17649|+/**
17650|+ * marcros que criei baseado no que pesquisei
17651|+ * sobre formatacao e codificacao de terminal vt100
17652|diff --git a/src/main.c b/src/main.c
17653|index a51b854..f669c60 100644
17654|--- a/src/main.c
17655|+++ b/src/main.c
17656|@@ -1,4 +1,4 @@
17657|+/**
17658|+/*
17659|+ * aluno: joilnen leite
17660|+ */
17661|#include <stdio.h>
17662|diff --git a/src/testa.h b/src/testa.h
17663|index 494ed68..302e359 100644
17664|--- a/src/testa.h
17665|+++ b/src/testa.h
17666|@@ -1,6 +1,10 @@
17667|#ifndef TESTA_H_
17668|#define TESTA_H_
17669|
17670|+/*
17671|+ * aluno: joilnen leite
17672|+ */
17673|+
17674|#include "jurses.h"
17675|#include "testa_item_1.h"
17676|#include "testa_item_2.h"
17677|diff --git a/src/testa_item_2.c b/src/testa_item_2.c
17678|index 80d0813..9de4f95 100644
17679|--- a/src/testa_item_2.c
17680|+++ b/src/testa_item_2.c
17681|@@ -1,6 +1,10 @@
17682|#include <stdio.h>
17683|#include "conjunto_ordenado.h"
17684|-#include "testa_item_2.h"
17685|+#include "testa.h"
17686|+
17687|+/*
17688|+ * aluno: joilnen leite
17689|+ */
17690|+
17691|void tela_item_2(const char *m)
17692|{
17693|diff --git a/src/testa_item_2.h b/src/testa_item_2.h
17694|index ad0f154..3544916 100644
17695|--- a/src/testa_item_2.h
17696|+++ b/src/testa_item_2.h
17697|@@ -1,6 +1,10 @@
17698|#ifndef TESTA_ITEM_2_H_
17699|#define TESTA_ITEM_2_H_
17700|
17701|+/*
17702|+ * aluno: joilnen leite
17703|+ */
17704|+
17705|void tela_item_2(const char *m);
17706|void testa_item_2(const char *m);
17707|
17708|diff --git a/src/testa_item_3.c b/src/testa_item_3.c
17709|index 358d1a8..e86262d 100644
17710|--- a/src/testa_item_3.c
17711|+++ b/src/testa_item_3.c
17712|@@ -1,6 +1,10 @@
17713|#include "conjunto_ordenado.h"
17714|#include "testa_item_3.h"
17715|
17716|+/*
17717|+ * aluno: joilnen leite
17718|+ */
17719|+
17720|void tela_item_3(const char *m)
17721|{
17722|    LTELA;
17723|diff --git a/src/testa_item_3.h b/src/testa_item_3.h
17724|index 3808f42..30f724e 100644
17725|--- a/src/testa_item_3.h
17726|+++ b/src/testa_item_3.h
17727|@@ -1,6 +1,10 @@
17728|#ifndef TESTA_ITEM_3_H_
17729|#define TESTA_ITEM_3_H_
17730|
17731|+/*
17732|+ * aluno: joilnen leite
17733|+ */
17734|+
17735|void tela_item_3(const char *m);
17736|void testa_item_3(const char *m);
17737|
17738|diff --git a/src/testa_rb.c b/src/testa_rb.c

```

```

17739|index cbaae9d..d218910 100644
17740|--- a/src/testa_rb.c
17741|+++ b/src/testa_rb.c
17742|@@ -1,4 +1,4 @@
17743|-/**
17744|+/*
17745|+ * aluno: joilnen leite
17746|+ */
17747|
17748|diff --git a/src/testa_rb.h b/src/testa_rb.h
17749|index d568f6b..8224693 100644
17750|--- a/src/testa_rb.h
17751|+++ b/src/testa_rb.h
17752|@@ -1,7 +1,7 @@
17753|#ifndef TESTA_ARVORE_RB_
17754|#define TESTA_ARVORE_RB_
17755|
17756|-/**
17757|+/*
17758|+ * aluno: joilnen leite
17759|+ */
17760|
17761|commit f85a3e3d3d12e500fa7672fd55879a9dc26fb70f8
17762|Author: Joilnen <joilnen@gmail.com>
17763|Date: Tue Jan 4 09:38:27 2022 -0300
17764|
17765|criado segundo lote de testes, delecao, tornar vazio, cardinalidade e outros
17766|
17767|diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
17768|index 3ald69f..1ee4218 100644
17769|--- a/src/conjunto_ordenado.c
17770|+++ b/src/conjunto_ordenado.c
17771|@@ -58,10 +58,15 @@ int conjord_insere_elemento(conjord *a, int chave, void *elemento)
17772|
17773|void conjord_deleta_elemento(conjord *a, int chave)
17774|{
17775|- rb_delete(&a->pr, rb_search(a->pr, chave));
17776|- a->cardinalidade--;
17777|- if(a->cardinalidade < 0)
17778|- a->cardinalidade = 0;
17779|+ rb_n *t = rb_search(a->pr, chave);
17780|+
17781|+ if(t)
17782|+ {
17783|+ rb_delete(&a->pr, t);
17784|+ a->cardinalidade--;
17785|+ if(a->cardinalidade < 0)
17786|+ a->cardinalidade = 0;
17787|+ }
17788|}
17789|
17790|int conjord_cardinalidade(conjord *c)
17791|@@ -89,13 +94,29 @@ int conjord_eh_vazio(const conjord *a)
17792|return !a->pr;
17793|}
17794|
17795|+static void conjord_esvazia_helper(rb_n *r)
17796|+{
17797|+ if (r == &rb_s)
17798|+ return;
17799|+ rb_n *temp = r->p;
17800|+ conjord_esvazia_helper(r->l);
17801|+ free(r->d);
17802|+ r->d = NULL;
17803|+ conjord_esvazia_helper(r->r);
17804|+}
17805|+
17806|+/*
17807|+ * deleta todos os elementos do
17808|+ * conjunto e zera a cardinalidade
17809|+ */
17810|-int conjord_esvazia(conjord *a)
17811|+int conjord_esvazia(conjord *c)
17812|{
17813|-
17814|+ if (c && c->pr)
17815|+ {
17816|+ conjord_esvazia_helper(c->pr);
17817|+ c->cardinalidade = 0;
17818|+ free(c->pr);
17819|+ c->pr = NULL;
17820|+ }
17821|}
17822|
17823|void conjord_imprime_arvore_rb(conjord *c, int s)
17824|diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
17825|index ded9113..3366c55 100644
17826|--- a/src/conjunto_ordenado.h
17827|+++ b/src/conjunto_ordenado.h
17828|@@ -145,6 +145,8 @@ void conjord_imprime_conjunto(conjord *c);
17829|#define CKLSITEM(l,c,b,x) \
17830|if (b) printf(S_CHKLIST_DONE, l, c, x); \
17831|else printf(S_CHKLIST, l, c, x)
17832|+ #define INFO_FILE(x) printf("%s%s", S_INFO_ST, x, S_NORM)
17833|+ #define INFO_CARD(x) printf("%scardinalidade: %d", S_CARD, x, S_NORM)
17834|
17835|#endif
17836|
17837|diff --git a/src/jcurses.h b/src/jcurses.h
17838|index 19af7b9..767dde3 100644
17839|--- a/src/jcurses.h
17840|+++ b/src/jcurses.h
17841|@@ -23,9 +23,11 @@
17842|#define S_NORM "\033[0m"
17843|#define S_INFO "\033[24;0H\033[7m"
17844|#define S_INFO_ST "\033[0;65H"
17845|#define S_CARD "\033[0;60H"
17846|#define S_CHKLIST "\033[%d;%dH\033[0m"
17847|#define S_CHKLIST_DONE "\033[46m\033[30m\033[%d;%dH\033[0m"
17848|#define S_INFO02 "\033[46m\033[30m\033[24;0H"
17849|#define S_LIST "\033[%d;78H\033[0m"
17850|#define S_FILEN "\033[0;55H"
17851|
17852|#endif
17853|diff --git a/src/testa_item_1.c b/src/testa_item_1.c
17854|index 0b9b846..925553b 100644

```

```

17855 --- a/src/testa_item_1.c
17856 +++ b/src/testa_item_1.c
17857 @@ -55,7 +55,7 @@ void testa_item_1(const char *m)
17858     conjord *cj_a = conjord_cria("A");
17859     LTELA;
17860     adiciona_lista_de_elementos(cj_a);
17861 - INFO("* item 2.1 inserir 20 elem. chave unica entre 1 a 99");
17862 + INFO("* item 2.1 inserido 20 elem. chave unica entre 1 a 99");
17863     CM;
17864     LTELA;
17865     tela_checklist(1 << I2_1);
17866 @@ -63,7 +63,7 @@ void testa_item_1(const char *m)
17867     LTELA;
17868     conjord_imprime_arvore_rb(cj_a, 1);
17869 - INFO("* item 4 mostrar arvore atual");
17870 + INFO("* item 4 mostra da arvore atual");
17871     CM;
17872     LTELA;
17873 @@ -72,7 +72,7 @@ void testa_item_1(const char *m)
17874     LTELA;
17875     conjord_imprime_conjunto(cj_a);
17876 - INFO("* item 1.1 ordenar por chave de identificacao");
17877 + INFO("* item 1.1 ordenado por chave de identificacao");
17878     CM;
17879     LTELA;
17880 @@ -93,6 +93,7 @@ void tela_item_1(const char *m)
17881     CKLSITEM(l++, c, 0, "e ordenacao dos elementos do conjunto");
17882     CKLSITEM(l++, c, 0, "o tipo de dado do usuario aqui eh uma");
17883     CKLSITEM(l++, c, 0, "estrutura com duas strings, nome e telefone");
17884 + INFO_FILE(__FILE__);
17885     CM;
17886 }
17887
17888 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
17889 index 9de4f95..c5c36d7 100644
17890 --- a/src/testa_item_2.c
17891 +++ b/src/testa_item_2.c
17892 @@ -1,34 +1,114 @@
17893 #include <stdio.h>
17894 #include <string.h>
17895 #include <time.h>
17896 #include "conjunto_ordenado.h"
17897 #include "testa.h"
17898 #include "checklist.h"
17899
17900 /*
17901  * aluno: joilnen leite
17902  */
17903
17904 +static void imprime_meu_tipo(void *);
17905 +
17906 +typedef struct {
17907 +    char *nome;
17908 +    int milhas;
17909 +} meu_tipo2;
17910 +
17911 +static void adiciona_lista_de_elementos(conjord *c)
17912 +{
17913 +    int i;
17914 +    srand(time(NULL));
17915 +    for (i = 0; i < 20;)
17916 +    {
17917 +        /**
17918 +         * definicao de um tipo exemplo pelo o usuario
17919 +         */
17920 +        /**
17921 +         * criacao do dado para o tipo do usuario
17922 +         */
17923 +        meu_tipo2 *mt0 = (meu_tipo2 *) malloc (sizeof(meu_tipo2));
17924 +        mt0->nome = (char *) malloc(sizeof("nome para teste1") + 1);
17925 +        strncpy(mt0->nome, "nome para teste1", sizeof("nome para teste1") + 1);
17926 +        mt0->milhas = 19;
17927 +        int chave = rand() % 99 + 1;
17928 +        if (!conjord_inserir_elemento(c, chave, mt0))
17929 +        {
17930 +            ++i;
17931 +            printf("chave:%d \t", chave);
17932 +            printf("nome:%s\t", mt0->nome);
17933 +            printf("milhas:%d\n", mt0->milhas);
17934 +        }
17935 +    }
17936 +}
17937 +
17938 +void testa_item_2(const char *m)
17939 +{
17940 +    /**
17941 +     * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
17942 +     */
17943 +    conjord_setar_imprime_elemento(imprime_meu_tipo);
17944 +    conjord *cj_a = conjord_cria("A");
17945 +    int imp;
17946 +
17947 +    LTELA;
17948 +    adiciona_lista_de_elementos(cj_a);
17949 +    INFO("* item 2.2 excluir todos com chave impar");
17950 +    CM;
17951 +
17952 +    for (imp = 1; imp < 100; imp += 2)
17953 +        conjord_deleta_elemento(cj_a, imp);
17954 +
17955 +    LTELA;
17956 +    conjord_imprime_conjunto(cj_a);
17957 +    INFO("* item 1.4 excluir todos");
17958 +    INFO_CARD(conjord_cardinalidade(cj_a));
17959 +    CM;
17960 +
17961 +    LTELA;
17962 +    conjord_esvazia(cj_a);
17963 +    INFO_CARD(conjord_cardinalidade(cj_a));
17964 +    INFO("* item 1.3 conjunto vazio");
17965 +    CM;
17966 }
17967

```

```

17971 + LTELA;
17972 + tela_checklist(1 << I1_2);
17973 + tela_checklist(1 << I1_3);
17974 + tela_checklist(1 << I1_4);
17975 + tela_checklist(1 << I2_2);
17976 + tela_checklist(1 << I2_3);
17977 + CM;
17978 +}
17979 +
17980 void tela_item_2(const char *m)
17981 {
17982     LTELA;
17983     int l = 5, c = 24, f;
17984     char mens[6][40] =
17985 +     char mens[8][45] =
17986     {
17987         "\033[21mserah efetuado os testes:\033[0m",
17988         S_UNDERL"serah efetuado os testes:"S_NORM,
17989         "exclusao de elemento do conjunto",
17990         "verif. se elemento pertence ao conjunto",
17991         "cardinalide do conjunto",
17992         "verif. se conjunto eh vazio",
17993         "tornar conjunto vazio",
17994         "tornar conjunto vazio",
17995         "o tipo de dado do usuario aqui eh uma",
17996         "estrutura com uma string e um inteiro"
17997     };
17998     for (f = 0; f < 7; f++)
17999 +     for (f = 0; f < 8; f++)
18000         CKLSITEM(l++, c, 0, mens[f]);
18001 + INFO_FILE(__FILE__);
18002     CM;
18003 }
18004
18005 -
18006 -void testa_item_2(const char *m)
18007 +void imprime_meu_tipo(void *n)
18008 {
18009     LTELA;
18010     printf("Testa item 2\n");
18011     CM;
18012     if(n)
18013     {
18014         printf("nome: %s \t", ((meu_tipo2 *)n)->nome);
18015         printf("milhas: %d\n", ((meu_tipo2 *)n)->milhas);
18016     }
18017 }
18018
18019 +
18020 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
18021 index e86262d..1f68940 100644
18022 --- a/src/testa_item_3.c
18023 +++ b/src/testa_item_3.c
18024 @@ -1,4 +1,5 @@
18025 #include <stdio.h>
18026 #include "jurses.h"
18027 #include "conjunto_ordenado.h"
18028 #include "testa_item_3.h"
18029
18030 @@ -6,10 +7,32 @@
18031 * aluno: joilnen leite
18032 */
18033
18034 /*
18035 * * aluno: joilnen leite
18036 */
18037 +
18038 +static void imprime_meu_tipo(void *);
18039 +
18040 +typedef struct {
18041 +    char letra;
18042 +} meu_tipo3;
18043 +
18044 void tela_item_3(const char *m)
18045 {
18046     LTELA;
18047     printf("Testa item 3\n");
18048     int l = 5, c = 24, f;
18049     char mens[6][48] =
18050     {
18051         S_UNDERL"serah efetuado os testes:"S_NORM,
18052         "interseccao de conjuntos",
18053         "uniao de conjuntos",
18054         "subtracao de conjuntos",
18055         "o tipo de dado do usuario aqui eh um",
18056         "caracter representando letras do alfabeto"
18057     };
18058     for (f = 0; f < 6; f++)
18059         CKLSITEM(l++, c, 0, mens[f]);
18060 + INFO_FILE(__FILE__);
18061     CM;
18062 }
18063
18064 @@ -19,3 +42,11 @@ void testa_item_3(const char *m)
18065 }
18066
18067 +void imprime_meu_tipo(void *n)
18068 +{
18069 +    if(n)
18070 +    {
18071         printf("%c ", ((meu_tipo3 *)n)->letra);
18072     }
18073 +}
18074 +
18075 +
18076 +
18077 commit 33ec35c9d0e06aef527ddf78d41da1b9bddf8a82
18078 Author: Joilnen <joilnen@gmail.com>
18079 Date: Tue Jan 4 14:40:07 2022 -0300
18080
18081     inciando lote de teste 3 tentarei fazer um diagrama de venn em texto
18082
18083 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18084 index 1ee4218..77ef88e 100644
18085 --- a/src/conjunto_ordenado.c
18086 +++ b/src/conjunto_ordenado.c

```

```

18087 @@ -155,4 +155,21 @@ void conjord_imprime_conjunto(conjord *c)
18088     conjord_imprime_conjunto_helper(c->pr);
18089 }
18090
18091 +static void conjord_imprime_venn_helper(rb_n *n)
18092 +{
18093 +    static int ang = 0, ray = 0;
18094 +    if (n == &rb_s)
18095 +        return;
18096 +
18097 +    conjord_imprime_conjunto_helper(n->l);
18098 +    printf("chave: %d\t", n->k);
18099 +    conjord_imprime_elemento(n->d);
18100 +    conjord_imprime_conjunto_helper(n->r);
18101 +}
18102 +
18103 +void conjord_imprime_venn(conjord *c)
18104 +{
18105 +    if (c && c->pr)
18106 +        conjord_imprime_conjunto_helper(c->pr);
18107 +}
18108
18109 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
18110 index 3366c55..c87d10d 100644
18111 --- a/src/conjunto_ordenado.h
18112 +++ b/src/conjunto_ordenado.h
18113 @@ -126,6 +126,8 @@ int conjord_seta_imprime_elemento(void (*)(void *n));
18114
18115 void conjord_imprime_conjunto(conjord *c);
18116
18117 +void conjord_imprime_venn(conjord *c);
18118 +
18119 /**
18120  * foi utilizado macro em vez de funcao
18121 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
18122 index 1f68940..1315387 100644
18123 --- a/src/testa_item_3.c
18124 +++ b/src/testa_item_3.c
18125 @@ -1,8 +1,9 @@
18126 @@ -1,8 +1,9 @@
18127 #include <stdio.h>
18128 #include "jurses.h"
18129 #include <string.h>
18130 #include <time.h>
18131 #include "conjunto_ordenado.h"
18132 #include "testa_item_3.h"
18133 -
18134 #include "testa.h"
18135 #include "checklist.h"
18136 /*
18137 * aluno: joilnen leite
18138 */
18139 @@ -17,6 +18,12 @@ typedef struct {
18140     char letra;
18141 } meu_tipo3;
18142
18143 +
18144 +static void adiciona_venn(conjord *c, int index)
18145 +{
18146 +}
18147 +
18148 +
18149 void tela_item_3(const char *m)
18150 {
18151     LTEL;
18152 @@ -39,14 +46,43 @@ void tela_item_3(const char *m)
18153
18154 void testa_item_3(const char *m)
18155 {
18156     /**
18157      * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
18158      */
18159     conjord_seta_imprime_elemento(imprime_meu_tipo);
18160     conjord *cj_a = conjord_cria("A");
18161     int imp, pos0 = 0, pos1 = 1, pos2 = 2;
18162
18163     LTEL;
18164     adiciona_venn(cj_a, pos0);
18165     INFO("item 2.2 excluir todos com chave impar");
18166     CM;
18167
18168     // for (imp = 1; imp < 100; imp += 2)
18169     //     conjord_deleta_elemento(cj_a, imp);
18170     LTEL;
18171     conjord_imprime_conjunto(cj_a);
18172     INFO("item 1.4 excluir todos");
18173     INFO_CARD(conjord_cardinalidade(cj_a));
18174     CM;
18175
18176     LTEL;
18177     conjord_esvazia(cj_a);
18178     INFO_CARD(conjord_cardinalidade(cj_a));
18179     INFO("item 1.3 conjunto vazio");
18180     CM;
18181
18182     LTEL;
18183     tela_checklist(1 << I3_1);
18184     tela_checklist(1 << I3_2);
18185     tela_checklist(1 << I3_3);
18186     CM;
18187 }
18188
18189 void imprime_meu_tipo(void *n)
18190 {
18191     if(n)
18192     {
18193         printf("%c ", ((meu_tipo3 *)n)->letra);
18194     }
18195 }
18196
18197 +
18198 +
18199 commit cf31603f0769c38142f76f178eae220925ea3c95
18200 Author: Joilnen <joilnen@gmail.com>
18201 Date: Tue Jan 4 16:31:17 2022 -0300
18202

```

```

18203     implementando um diagrama de venn, dificil soh em texto,
18204     imprime todo fora de lugra mas n quebra
18205
18206 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18207 index 77ef88e..0f75d95 100644
18208 --- a/src/conjunto_ordenado.c
18209 +++ b/src/conjunto_ordenado.c
18210 @@ -1,4 +1,5 @@
18211 #include <stdio.h>
18212 #include <math.h>
18213 #include "conjunto_ordenado.h"
18214
18215 static rb_n **raiz;
18216 @@ -155,21 +156,66 @@ void conjord_imprime_conjunto(conjord *c)
18217     conjord_imprime_conjunto_helper(c->pr);
18218 }
18219
18220 -static void conjord_imprime_venn_helper(rb_n *n)
18221 +static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18222 {
18223     static int ang = 0, ray = 0;
18224     int l, c;
18225     static int rai = 0;
18226     if (n == &rb_s)
18227         return;
18228
18229     conjord_imprime_conjunto_helper(n->l);
18230     printf("chave: %d\t", n->k);
18231     conjord_imprime_venn_helper(n->l, ang + M_PI/8, 0);
18232
18233     if (ang > (2 * M_PI))
18234     {
18235         ang = 0;
18236         rai += 2;
18237     }
18238
18239     if (!ang && !rai)
18240     {
18241         rai = 2;
18242         ang = M_PI / 4;
18243     }
18244     else
18245     {
18246         int pl = (int)(floor(cosf(ang)) * rai);
18247         int pc = (int)(floor(sinf(ang)) * rai);
18248         int cnt, ud, lr;
18249         lr = (pl > 0) ? 1: -1;
18250         ud = (pc > 0) ? 1: -1;
18251         pl = abs(pl); pc = abs(pc);
18252
18253         for (cnt = 0; cnt < pl; cnt++);
18254         {
18255             if (lr > 0) printf("\033[C");
18256             else printf("\033[D");
18257         }
18258
18259         for (cnt = 0; cnt < pc; cnt++);
18260         {
18261             if (ud > 0) printf("\033[B");
18262             else printf("\033[A");
18263         }
18264     }
18265
18266     conjord_imprime_elemento(n->d);
18267     conjord_imprime_conjunto_helper(n->r);
18268     printf("\033[0m");
18269
18270     conjord_imprime_venn_helper(n->r, ang + M_PI/4, 0);
18271 }
18272
18273 -void conjord_imprime_venn(conjord *c)
18274 +void conjord_imprime_venn(conjord *c, int pos)
18275 {
18276     if (c && c->pr)
18277         conjord_imprime_conjunto_helper(c->pr);
18278     {
18279         if (pos == 2)
18280             printf("\033[12;55H");
18281         else if (pos == 1)
18282             printf("\033[12;15H");
18283         else
18284             printf("\033[16;40H");
18285         conjord_imprime_venn_helper(c->pr, 0, 0);
18286     }
18287 }
18288
18289 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
18290 index c87d10d..836f8d9 100644
18291 --- a/src/conjunto_ordenado.h
18292 +++ b/src/conjunto_ordenado.h
18293 @@ -126,7 +126,7 @@ int conjord_seta_imprime_elemento(void (*i)(void *n));
18294 @@ -126,7 +126,7 @@ void conjord_imprime_conjunto(conjord *c);
18295
18296 -void conjord_imprime_venn(conjord *c);
18297 +void conjord_imprime_venn(conjord *c, int pos);
18298
18299 /**
18300 diff --git a/src/jcurses.h b/src/jcurses.h
18301 index 767dde3..98b5ec3 100644
18302 --- a/src/jcurses.h
18303 +++ b/src/jcurses.h
18304 @@ -29,5 +29,6 @@
18305 #define S_INF02 "\033[46m\033[30m\033[24;0H"
18306 #define S_LIST "\033[%d;78H%d"
18307 #define S_FILEN "\033[0;55H"
18308 #define S_VENN S_CHKLIST
18309
18310 #endif
18311 diff --git a/src/makefile b/src/makefile
18312 index 2ddd9d1..064af43 100644
18313 --- a/src/makefile
18314 +++ b/src/makefile
18315 @@ -13,7 +13,7 @@ D=-g

```

```

18319 .c.o:
18320 $(C) -c $(D) $<
18321 all: $(O)
18322 - $(C) -o $(E) $(O)
18323 + $(C) -o $(E) $(O) -lm
18324 clean:
18325 rm -f *.o $(E)
18326
18327 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
18328 index 1315387..7521d2f 100644
18329 --- a/src/testa_item_3.c
18330 +++ b/src/testa_item_3.c
18331 @@ -11,17 +11,28 @@
18332 /*
18333  * aluno: joilnen leite
18334  */
18335 -
18336 static void imprime_meu_tipo(void *);
18337
18338 typedef struct {
18339     char letra;
18340 } meu_tipo3;
18341
18342 -
18343 static void adiciona_venn(conjord *c, int index)
18344 +static void adiciona_venn(conjord *c, int pos)
18345 {
18346     int i;
18347     for (i = 0; i < 10;)
18348     {
18349         /**
18350          * criacao do dado para o tipo do usuario
18351          */
18352         meu_tipo3 *mt0 = (meu_tipo3 *) malloc (sizeof(meu_tipo3));
18353         mt0->letra = 'A' + i;
18354         if (!conjord_inserir_elemento(c, i, mt0))
18355             ++i;
18356     }
18357
18358     conjord_imprime_venn(c, pos);
18359 }
18360
18361 void tela_item_3(const char *m)
18362 @@ -55,27 +66,7 @@ void testa_item_3(const char *m)
18363
18364     LTELA;
18365     adiciona_venn(cj_a, pos0);
18366     INFO("item 2.2 excluir todos com chave impar");
18367     CM;
18368
18369     // for (imp = 1; imp < 100; imp += 2)
18370     //     conjord_deleta_elemento(cj_a, imp);
18371     LTELA;
18372     conjord_imprime_conjunto(cj_a);
18373     INFO("item 1.4 excluir todos");
18374     INFO_CARD(conjord_cardinalidade(cj_a));
18375     CM;
18376
18377     LTELA;
18378     conjord_esvazia(cj_a);
18379     INFO_CARD(conjord_cardinalidade(cj_a));
18380     INFO("item 1.3 conjunto vazio");
18381     CM;
18382
18383     LTELA;
18384     tela_checklist(1 << I3_1);
18385     tela_checklist(1 << I3_2);
18386     tela_checklist(1 << I3_3);
18387     INFO("item 3.1 operacoes com conjuntos");
18388     CM;
18389 }
18390
18391
18392
18393 commit 5a3952ee1b1535e9a9bca6a07cd133115f0d9a84
18394 Author: Joilnen <joilnen@gmail.com>
18395 Date: Tue Jan 4 17:05:27 2022 -0300
18396
18397     tentando consertar o venn
18398
18399 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18400 index 0f75d95..c20c6cd 100644
18401 --- a/src/conjunto_ordenado.c
18402 +++ b/src/conjunto_ordenado.c
18403 @@ -178,12 +178,12 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18404     {
18405         else
18406         {
18407             int pl = (int)(floor(cosf(ang)) * rai);
18408             int pc = (int)(floor(sinf(ang)) * rai);
18409             int pl = (int)(cosf(ang) * rai);
18410             int pc = (int)(sinf(ang) * rai);
18411             int cnt, ud, lr;
18412             lr = (pl > 0) ? 1: -1;
18413             ud = (pc > 0) ? 1: -1;
18414             pl = abs(pl); pc = abs(pc);
18415             pl = floor(pl); pc = floor(pc);
18416
18417             for (cnt = 0; cnt < pl; cnt++);
18418             {
18419                 @@ -213,7 +213,7 @@ void conjord_imprime_venn(conjord *c, int pos)
18420                 else if (pos == 1)
18421                     printf("\033[12;15H");
18422                 else
18423                     printf("\033[16;40H");
18424                 printf("\033[18;40H");
18425                 conjord_imprime_venn_helper(c->pr, 0, 0);
18426             }
18427         }
18428     }
18429
18430 commit ff49400b5be6f5bfeed95f63c2bc8ac3373f4b8a
18431 Author: Joilnen <joilnen@gmail.com>
18432 Date: Tue Jan 4 17:42:43 2022 -0300
18433
18434     experiencias nessa ideia de venn

```



```

18435 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18436 index c20c6cd..c4713a7 100644
18437 --- a/src/conjunto_ordenado.c
18438 +++ b/src/conjunto_ordenado.c
18439 @@ -168,7 +168,7 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18440     if (ang > (2 * M_PI))
18441     {
18442         ang = 0;
18443         rai += 2;
18444         rai += 12;
18445     }
18446
18447     if (!ang && !rai)
18448 @@ -176,30 +176,33 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18449         rai = 2;
18450         ang = M_PI / 4;
18451     }
18452     else
18453     {
18454         int pl = (int)(cosf(ang) * rai);
18455         int pc = (int)(sinf(ang) * rai);
18456         int cnt, ud, lr;
18457         lr = (pl > 0) ? 1: -1;
18458         ud = (pc > 0) ? 1: -1;
18459         pl = floor(pl); pc = floor(pc);
18460
18461         printf("\033[18;40H");
18462         printf("\033[%d;%dH", 18 + pl, 40 + pc);
18463         printf("%d %d %f", pl, pc, ang);
18464         /*
18465         for (cnt = 0; cnt < pl; cnt++);
18466         {
18467             if (lr > 0) printf("\033[C");
18468             else printf("\033[D");
18469         }
18470
18471         for (cnt = 0; cnt < pc; cnt++);
18472         {
18473             int pl = (int)(cosf(ang) * rai);
18474             int pc = (int)(sinf(ang) * rai);
18475             int cnt, ud, lr;
18476             lr = (pl > 0) ? 1: -1;
18477             ud = (pc > 0) ? 1: -1;
18478             pl = floor(pl); pc = floor(pc);
18479
18480             for (cnt = 0; cnt < pl; cnt++);
18481             {
18482                 if (lr > 0) printf("\033[C");
18483                 else printf("\033[D");
18484             }
18485
18486             for (cnt = 0; cnt < pc; cnt++);
18487             {
18488                 if (ud > 0) printf("\033[B");
18489                 else printf("\033[A");
18490             }
18491             if (ud > 0) printf("\033[B");
18492             else printf("\033[A");
18493         }
18494         */
18495
18496         conjord_imprime_elemento(n->d);
18497         printf("\033[0m");
18498         getc(stdin);
18499     }
18500     conjord_imprime_venn_helper(n->r, ang + M_PI/4, 0);
18501 }
18502
18503 commit f2e0fc34b62d8f4067247e11836f36f7e25aadfa
18504 Author: Joilnen <joilnen@gmail.com>
18505 Date: Tue Jan 4 18:09:41 2022 -0300
18506
18507 venn
18508
18509 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18510 index c4713a7..5ff4920 100644
18511 --- a/src/conjunto_ordenado.c
18512 +++ b/src/conjunto_ordenado.c
18513 @@ -163,9 +163,9 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18514     if (n == &rb_s)
18515         return;
18516
18517     conjord_imprime_venn_helper(n->l, ang + M_PI/8, 0);
18518     conjord_imprime_venn_helper(n->l, ang + 22.5, 0);
18519
18520     if (ang > (2 * M_PI))
18521     if (ang > 360)
18522     {
18523         ang = 0;
18524         rai += 12;
18525 @@ -174,11 +174,11 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18526     if (!ang && !rai)
18527     {
18528         rai = 2;
18529         ang = M_PI / 4;
18530         ang = 45;
18531     }
18532
18533     int pl = (int)(cosf(ang) * rai);
18534     int pc = (int)(sinf(ang) * rai);
18535     int pl = (int)(cosf(ang * M_PI / 180) * rai);
18536     int pc = (int)(sinf(ang * M_PI / 180) * rai);
18537     int cnt, ud, lr;
18538     lr = (pl > 0) ? 1: -1;
18539     ud = (pc > 0) ? 1: -1;
18540 @@ -186,7 +186,7 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18541
18542     printf("\033[18;40H");
18543     printf("\033[%d;%dH", 18 + pl, 40 + pc);
18544     printf("%d %d %f", pl, pc, ang);
18545     /* printf("%d %d %f", pl, pc, ang);
18546     */
18547     for (cnt = 0; cnt < pl; cnt++);
18548     {
18549 @@ -204,7 +204,7 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18550     conjord_imprime_elemento(n->d);

```

```

18551     getc(stdin);
18552
18553     - conjord_imprime_venn_helper(n->r, ang + M_PI/4, 0);
18554     + conjord_imprime_venn_helper(n->r, ang + 45, 0);
18555 }
18556
18557 void conjord_imprime_venn(conjord *c, int pos)
18558
18559 commit 6508aea78932f1485fd5b46f48921ecaa009384b
18560 Author: Joilnen <joilnen@gmail.com>
18561 Date: Wed Jan 5 00:38:43 2022 -0300
18562
18563     midpoing algo pra simular um diagrama de venn
18564
18565 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18566 index 5ff4920..17a21c8 100644
18567 --- a/src/conjunto_ordenado.c
18568 +++ b/src/conjunto_ordenado.c
18569 @@ -158,6 +158,7 @@ void conjord_imprime_conjunto(conjord *c)
18570
18571 static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18572 {
18573     + /*
18574     + int l, c;
18575     + static int rai = 0;
18576     + if (n == &rb_s)
18577     @@ -186,6 +187,39 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18578
18579     printf("\033[18;40H");
18580     printf("\033[%d;%dH", 18 + pl, 40 + pc);
18581     +
18582     +
18583     + int x = 0;
18584     + int y = 4, p;
18585     + int a = 18, b = 40, r = 2;
18586     + printf("\033[%d;%dH", a, b+r);
18587     + printf("\033[%d;%dH", a, b-r);
18588     + printf("\033[%d;%dH", a-r, b);
18589     + printf("\033[%d;%dH", a+r, b);
18590     +
18591     + p = 5/4 - r;
18592     +
18593     + while (x <= y)
18594     + {
18595     +     if (p < 0)
18596     +         p += (4*x) + 6;
18597     +     else
18598     +     {
18599     +         p += (2*(x-y)) + 5;
18600     +         y--;
18601     +     }
18602     +     x++;
18603     +     printf("\033[%d;%dH", a+x, b+y);
18604     +     printf("\033[%d;%dH", a-x, b+y);
18605     +     printf("\033[%d;%dH", a+x, b-y);
18606     +     printf("\033[%d;%dH", a-x, b-y);
18607     +     printf("\033[%d;%dH", a+x, b+y);
18608     +     printf("\033[%d;%dH", a-x, b-y);
18609     +     printf("\033[%d;%dH", a-x, b+y);
18610     +     printf("\033[%d;%dH", a-x, b-y);
18611     + }
18612     +
18613     +
18614     // printf("%d %d %f", pl, pc, ang);
18615     +
18616     for (cnt = 0; cnt < pl; cnt++);
18617     @@ -201,10 +235,12 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18618     }
18619     +
18620     +
18621     +
18622     conjord_imprime_elemento(n->d);
18623     getc(stdin);
18624
18625     conjord_imprime_venn_helper(n->r, ang + 45, 0);
18626     +
18627     +
18628 }
18629
18630 void conjord_imprime_venn(conjord *c, int pos)
18631
18632 commit 0fc6c2b33b7839708675883a7956812be92af4e3
18633 Author: Joilnen <joilnen@gmail.com>
18634 Date: Wed Jan 5 06:22:48 2022 -0300
18635
18636     venn
18637
18638 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
18639 index 7521d2f..45cd0a9 100644
18640 --- a/src/testa_item_3.c
18641 +++ b/src/testa_item_3.c
18642 @@ -17,18 +17,18 @@ typedef struct {
18643     char letra;
18644 } meu_tipo3;
18645
18646 -static void adiciona_venn(conjord *c, int pos)
18647 +static void adiciona_venn(conjord *c, int pos, char *letras)
18648 {
18649     - int i;
18650     - for (i = 0; i < 10;)
18651     + char *i;
18652     + for (i = letras; *i; i++)
18653     {
18654         /**
18655         * criacao do dado para o tipo do usuario
18656         */
18657         meu_tipo3 *mt0 = (meu_tipo3 *) malloc (sizeof(meu_tipo3));
18658         mt0->letra = 'A' + i;
18659         mt0->letra = *i;
18660
18661         - if (!conjord_inserir_elemento(c, i, mt0))
18662         + if (!conjord_inserir_elemento(c, *i, mt0))
18663             ++i;
18664     }
18665
18666     @@ -62,10 +62,14 @@ void testa_item_3(const char *m)
18667     */

```

```

18667 | conjord_sete_imprime_elemento(imprime_meu_tipo);
18668 | conjord *cj_a = conjord_cria("A");
18669 | + conjord *cj_b = conjord_cria("B");
18670 | int imp, pos0 = 0, pos1 = 1, pos2 = 2;
18671 | + char elem0[] = "ABCDEFGHIIJKLMNOP";
18672 | + char elem1[] = "JKLMNOPQRSTUVWXYZ";
18673 |
18674 | LTELTA;
18675 | - adiciona_venn(cj_a, pos0);
18676 | + adiciona_venn(cj_a, pos1, elem0);
18677 | + adiciona_venn(cj_b, pos2, elem1);
18678 | INFO("* item 3.1 operacoes com conjuntos");
18679 | CM;
18680 | }
18681 |
18682 | commit 8f5fb5a53d26f7c703c4561aab72d0d42d104473
18683 | Author: Joilnen <joilnen@gmail.com>
18684 | Date: Wed Jan 5 18:12:29 2022 -0300
18685 |
18686 | diagrama de venn bem dificil tou quase desistindo
18687 |
18688 | diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18689 | index 17a21c8..d22c51b 100644
18690 | --- a/src/conjunto_ordenado.c
18691 | +++ b/src/conjunto_ordenado.c
18692 | @@ -156,47 +156,49 @@ void conjord_imprime_conjunto(conjord *c)
18693 |     conjord_imprime_conjunto_helper(c->pr);
18694 | }
18695 |
18696 | -static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18697 | +char buffer[100];
18698 | +int count_buffer = 0;
18699 | +static void conjord_retorna_elementos(rb_n *n)
18700 | {
18701 |     /****
18702 |     int l, c;
18703 |     static int rai = 0;
18704 |     if (n == &rb_s)
18705 |         return;
18706 |
18707 |     conjord_imprime_venn_helper(n->l, ang + 22.5, 0);
18708 |
18709 |     if (ang > 360)
18710 |     {
18711 |         ang = 0;
18712 |         rai += 12;
18713 |     }
18714 |
18715 |     if (!ang && !rai)
18716 |     {
18717 |         rai = 2;
18718 |         ang = 45;
18719 |         buffer[count_buffer] = '\0';
18720 |         return;
18721 |     }
18722 |
18723 |     int pl = (int)(cosf(ang * M_PI / 180) * rai);
18724 |     int pc = (int)(sinf(ang * M_PI / 180) * rai);
18725 |     int cnt, ud, lr;
18726 |     lr = (pl > 0) ? 1: -1;
18727 |     ud = (pc > 0) ? 1: -1;
18728 |     pl = floor(pl); pc = floor(pc);
18729 |
18730 |     printf("\033[18;40H");
18731 |     printf("\033[%d;%dH", 18 + pl, 40 + pc);
18732 |     ****/
18733 |     + buffer[count_buffer++] = *(char *)n->d;
18734 |     + conjord_retorna_elementos(n->l);
18735 |     + conjord_retorna_elementos(n->r);
18736 | }
18737 |
18738 | - int x = 0;
18739 | - int y = 4, p;
18740 | +static void conjord_imprime_venn_helper(char *letra)
18741 | +{
18742 |     + static int n_letra = 0;
18743 |     + int x = 0, y = 4, p;
18744 |     + int a = 18, b = 40, r = 2;
18745 |     + printf("\033[%d;%dH", a, b+r);
18746 |     + printf("\033[%d;%dH", a, b-r);
18747 |     + printf("\033[%d;%dH", a-r, b);
18748 |     + printf("\033[%d;%dH", a+r, b);
18749 |     + int c0, c1;
18750 |
18751 |     + if (!letra)
18752 |     +     return;
18753 |     +
18754 |     + for (c0 = 0; c0 < 4 && letra[c0]; c0++)
18755 |     + {
18756 |     +     switch (c0)
18757 |     +     {
18758 |     +     case 0:
18759 |     +         printf("\033[%d;%dH%c", a, b+r, letra[c0]);
18760 |     +         break;
18761 |     +     case 1:
18762 |     +         printf("\033[%d;%dH%c", a, b-r, letra[c0]);
18763 |     +         break;
18764 |     +     case 2:
18765 |     +         printf("\033[%d;%dH%c", a-r, b, letra[c0]);
18766 |     +         break;
18767 |     +     case 3:
18768 |     +         printf("\033[%d;%dH%c", a+r, b, letra[c0]);
18769 |     +     }
18770 |     + }
18771 |     +
18772 |     + getc(stdin);
18773 |     + p = 5/4 - r;
18774 |
18775 |     + while (x <= y)
18776 |     @@ -209,38 +211,37 @@ static void conjord_imprime_venn_helper(rb_n *n, float ang, int cor)
18777 |         y--;
18778 |     }
18779 |     x++;
18780 |     printf("\033[%d;%dH", a+x, b+y);
18781 |     printf("\033[%d;%dH", a-x, b+y);
18782 |     printf("\033[%d;%dH", a+x, b-y);
18783 |     printf("\033[%d;%dH", a-x, b-y);

```

```

18783 | -     printf("\033[%d;%dH", a+x, b+y);
18784 | -     printf("\033[%d;%dH", a+x, b-y);
18785 | -     printf("\033[%d;%dH", a-x, b+y);
18786 | -     printf("\033[%d;%dH", a-x, b-y);
18787 | +     for (; letra[c0]; c0++)
18788 | +     {
18789 | +         switch (c0 % 8)
18790 | +         {
18791 | +             case 0:
18792 | +                 printf("\033[%d;%dH%c", a+x, b+y, letra[c0]);
18793 | +                 break;
18794 | +             case 1:
18795 | +                 printf("\033[%d;%dH%c", a-x, b+y, letra[c0]);
18796 | +                 break;
18797 | +             case 2:
18798 | +                 printf("\033[%d;%dH%c", a+x, b-y, letra[c0]);
18799 | +                 break;
18800 | +             case 3:
18801 | +                 printf("\033[%d;%dH%c", a+x, b-y, letra[c0]);
18802 | +                 break;
18803 | +             case 4:
18804 | +                 printf("\033[%d;%dH%c", a+x, b+y, letra[c0]);
18805 | +                 break;
18806 | +             case 5:
18807 | +                 printf("\033[%d;%dH%c", a+x, b-y, letra[c0]);
18808 | +                 break;
18809 | +             case 6:
18810 | +                 printf("\033[%d;%dH%c", a-x, b+y, letra[c0]);
18811 | +                 break;
18812 | +             case 7:
18813 | +                 printf("\033[%d;%dH%c", a-x, b-y, letra[c0]);
18814 | +             }
18815 | +         getc(stdin);
18816 | +     }
18817 | + }
18818 | -
18819 | -
18820 | - // printf("%d %d %f", pl, pc, ang);
18821 | - /****
18822 | - for (cnt = 0; cnt < pl; cnt++);
18823 | - {
18824 | -     if (lr > 0) printf("\033[C");
18825 | -     else printf("\033[D");
18826 | - }
18827 | -
18828 | - for (cnt = 0; cnt < pc; cnt++);
18829 | - {
18830 | -     if (ud > 0) printf("\033[B");
18831 | -     else printf("\033[A");
18832 | - }
18833 | - ****/
18834 | -
18835 | - /****
18836 | - conjord_imprime_elemento(n->d);
18837 | - getc(stdin);
18838 | -
18839 | - conjord_imprime_venn_helper(n->r, ang + 45, 0);
18840 | - ****/
18841 | - }
18842 | -
18843 | void conjord_imprime_venn(conjord *c, int pos)
18844 | @@ -253,7 +254,10 @@ void conjord_imprime_venn(conjord *c, int pos)
18845 |     printf("\033[12;15H");
18846 |     else
18847 |         printf("\033[18;40H");
18848 |     conjord_imprime_venn_helper(c->pr, 0, 0);
18849 | +     /* tricky */
18850 | +     conjord_retorna_elementos(c->pr);
18851 | +     printf("BUFFER %s\n", buffer);
18852 | +     conjord_imprime_venn_helper(buffer);
18853 | }
18854 | }
18855 |
18856 | diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
18857 | index 836f8d9..e788918 100644
18858 | --- a/src/conjunto_ordenado.h
18859 | +++ b/src/conjunto_ordenado.h
18860 | @@ -119,6 +119,7 @@ void conjord_imprime_arvore_rb(conjord *c, int s);
18861 | void conjord_imprime_lista(int *l, int s);
18862 |
18863 | extern void (*conjord_imprime_elemento)(void *n);
18864 | +extern void* (*conjord_retorna_elemento)(void *n);
18865 | /**
18866 |  * seta a rotina de impressao do dado criado pelo cliente
18867 |  */
18868 |
18869 | commit fe7504be3a66dd69bb77630476af1df338f68eb9
18870 | Author: Joilnen <joilnen@gmail.com>
18871 | Date: Wed Jan 5 18:58:50 2022 -0300
18872 |
18873 | apenas algumas mensagens pra debug adicionadas
18874 |
18875 | diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18876 | index d22c51b..d4f8a02 100644
18877 | --- a/src/conjunto_ordenado.c
18878 | +++ b/src/conjunto_ordenado.c
18879 | @@ -166,8 +166,8 @@ static void conjord_retorna_elementos(rb_n *n)
18880 |     return;
18881 | }
18882 |
18883 | -     buffer[count_buffer++] = *(char *)n->d;
18884 | -     conjord_retorna_elementos(n->l);
18885 | +     buffer[count_buffer++] = *(char *)n->d;
18886 | -     conjord_retorna_elementos(n->r);
18887 | }
18888 |
18889 | @@ -256,7 +256,6 @@ void conjord_imprime_venn(conjord *c, int pos)
18890 |     printf("\033[18;40H");
18891 |     /* tricky */
18892 |     conjord_retorna_elementos(c->pr);
18893 |     printf("BUFFER %s\n", buffer);
18894 |     conjord_imprime_venn_helper(buffer);
18895 | }
18896 | }
18897 | diff --git a/src/testa_item_3.c b/src/testa_item_3.c
18898 | index 45cd0a9..13c3b00 100644

```

```

18899 --- a/src/testa_item_3.c
18900 +++ b/src/testa_item_3.c
18901 @@ -32,7 +32,6 @@ static void adiciona_venn(conjord *c, int pos, char *letras)
18902     ++i;
18903 }
18904
18905 - conjord_imprime_venn(c, pos);
18906 }
18907
18908 void testa_item_3(const char *m)
18909 @@ -70,6 +69,7 @@ void testa_item_3(const char *m)
18910     LTELA;
18911     adiciona_venn(cj_a, pos1, elem0);
18912     adiciona_venn(cj_b, pos2, elem1);
18913 + conjord_imprime_venn(cj_a, pos0);
18914     INFO("item 3.1 operacoes com conjuntos");
18915     CM;
18916 }
18917
18918 commit f30ce269aeb13ab515f6eff76a680687848153d1
18919 Author: Joilnen <joilnen@gmail.com>
18920 Date: Thu Jan 6 12:23:44 2022 -0300
18921
18922     consertado a inclusao do teste de conjuntos
18923
18924 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18925 index d4f8a02..89518d2 100644
18926 --- a/src/conjunto_ordenado.c
18927 +++ b/src/conjunto_ordenado.c
18928 @@ -156,7 +156,7 @@ void conjord_imprime_conjunto(conjord *c)
18929     conjord_imprime_conjunto_helper(c->pr);
18930 }
18931
18932 -char buffer[100];
18933 +char buffer[100] = "";
18934 int count_buffer = 0;
18935 static void conjord_retorna_elementos(rb_n *n)
18936 {
18937 @@ -168,6 +168,7 @@ static void conjord_retorna_elementos(rb_n *n)
18938
18939     conjord_retorna_elementos(n->l);
18940     buffer[count_buffer++] = *(char *)n->d;
18941 + // printf("%c\n", *(char *)n->d);
18942     conjord_retorna_elementos(n->r);
18943 }
18944
18945 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
18946 index 13c3b00..b6dc39b 100644
18947 --- a/src/testa_item_3.c
18948 +++ b/src/testa_item_3.c
18949 @@ -20,7 +20,7 @@ typedef struct {
18950 @@ -20,7 +20,7 @@ typedef struct {
18951     static void adiciona_venn(conjord *c, int pos, char *letras)
18952     {
18953         char *i;
18954         for (i = letras; *i; i++)
18955             for (i = letras; *i; i++)
18956                 /**
18957                  * criacao do dado para o tipo do usuario
18958 @@ -69,6 +69,9 @@ void testa_item_3(const char *m)
18959     LTELA;
18960     adiciona_venn(cj_a, pos1, elem0);
18961     adiciona_venn(cj_b, pos2, elem1);
18962 + conjord_imprime_conjunto(cj_a);
18963 + CM;
18964 + // conjord_imprime_conjunto(cj_b);
18965     conjord_imprime_venn(cj_a, pos0);
18966     INFO("item 3.1 operacoes com conjuntos");
18967     CM;
18968 @@ -77,7 +80,7 @@ void testa_item_3(const char *m)
18969 void imprime_meu_tipo(void *n)
18970 {
18971     if(n)
18972         printf("%c ", ((meu_tipo3 *)n->letra);
18973 + printf("%c \n", ((meu_tipo3 *)n->letra);
18974 }
18975
18976
18977
18978 commit 68b96c4cdb898cec0d91396dd468175f4aabedcc
18979 Author: Joilnen <joilnen@gmail.com>
18980 Date: Thu Jan 6 13:29:35 2022 -0300
18981
18982     impressao de conjuntos
18983
18984 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
18985 index 89518d2..f5706c1 100644
18986 --- a/src/conjunto_ordenado.c
18987 +++ b/src/conjunto_ordenado.c
18988 @@ -1,4 +1,5 @@
18989 #include <stdio.h>
18990 #include <string.h>
18991 #include <math.h>
18992 #include "conjunto_ordenado.h"
18993
18994 @@ -156,19 +157,14 @@ void conjord_imprime_conjunto(conjord *c)
18995     conjord_imprime_conjunto_helper(c->pr);
18996 }
18997
18998 -char buffer[100] = "";
18999 -int count_buffer = 0;
19000 +static char buffer[100] = "";
19001 +static int count_buffer = 0;
19002 static void conjord_retorna_elementos(rb_n *n)
19003 {
19004     if (n == &rb_s)
19005     {
19006         buffer[count_buffer] = '\0';
19007         return;
19008     }
19009
19010     conjord_retorna_elementos(n->l);
19011     buffer[count_buffer++] = *(char *)n->d;
19012 + // printf("%c\n", *(char *)n->d);
19013     conjord_retorna_elementos(n->r);
19014 }

```

```

19015 @@ -176,7 +172,7 @@ static void conjord_imprime_venn_helper(char *letra)
19016 {
19017     static int n_letra = 0;
19018     int x = 0, y = 4, p;
19019     - int a = 18, b = 40, r = 2;
19020     + int a = 18, b = 40, r = 8;
19021     int c0, c1;
19022
19023     if (!letra)
19024
19025     commit 3e2b62554d82b63b93e585c33352b25955d6ec30
19026     Author: Joilinen <joilinen@gmail.com>
19027     Date: Sat Jan 8 15:29:43 2022 -0300
19028
19029     uniao e subtracao de conjuntos funcionando serao usados pra fazer a interesecc
19030
19031     diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19032     index f5706c1..22e4a61 100644
19033     --- a/src/conjunto_ordenado.c
19034     +++ b/src/conjunto_ordenado.c
19035     @@ -157,104 +157,87 @@ void conjord_imprime_conjunto(conjord *c)
19036     {
19037         conjord_imprime_conjunto_helper(c->pr);
19038     }
19039
19040     -static char buffer[100] = "";
19041     -static int count_buffer = 0;
19042     -static void conjord_retorna_elementos(rb_n *n)
19043     +static void conjord_imprime_venn_helper(rb_n *n)
19044     {
19045         if (n == &rb_s)
19046             return;
19047         - conjord_retorna_elementos(n->l);
19048         - buffer[count_buffer++] = *(char *)n->d;
19049         - conjord_retorna_elementos(n->r);
19050     +
19051     + conjord_imprime_venn_helper(n->l);
19052     + conjord_imprime_elemento(n->d);
19053     + printf(" ");
19054     + conjord_imprime_venn_helper(n->r);
19055     +}
19056
19057     +void conjord_imprime_venn(conjord *c, int pos)
19058     +{
19059     +    if (c && c->pr)
19060     +    {
19061     +        printf("%s = {", c->nome);
19062     +        conjord_imprime_venn_helper(c->pr);
19063     +        printf("\033[D\033[D");
19064     +        printf("}");
19065     +    }
19066
19067     -static void conjord_imprime_venn_helper(char *letra)
19068     +/**
19069     + * insersecciona dois conjuntos formando um terceiro
19070     + * como resultado desta operacao retornando seu ponteiro
19071     + * a seguir A intesecc B = (A uniao B) - [(A - B) uniao (B - A)]
19072     + */
19073     +conjord *conjord_intersecciona(conjord *a, conjord *b)
19074     {
19075         - static int n_letra = 0;
19076         - int x = 0, y = 4, p;
19077         - int a = 18, b = 40, r = 8;
19078         - int c0, c1;
19079         + conjord *cj_c = conjord_cria("C");
19080         + conjord *cj_d = conjord_cria("D");
19081         + conjord *cj_e = conjord_cria("E");
19082         + conjord_une(conjord_une(cj_c, a), b);
19083     }
19084
19085     - if (!letra)
19086     +static void conjord_une_helper(conjord *c, rb_n *n)
19087     +{
19088     +    if (n == &rb_s)
19089     +        return;
19090
19091     - for (c0 = 0; c0 < 4 && letra[c0]; c0++)
19092     {
19093         - switch (c0)
19094         {
19095             case 0:
19096                 printf("\033[%d;%dH%c", a, b+r, letra[c0]);
19097                 break;
19098             case 1:
19099                 printf("\033[%d;%dH%c", a, b-r, letra[c0]);
19100                 break;
19101             case 2:
19102                 printf("\033[%d;%dH%c", a-r, b, letra[c0]);
19103                 break;
19104             case 3:
19105                 printf("\033[%d;%dH%c", a+r, b, letra[c0]);
19106         }
19107     }
19108     - getc(stdin);
19109     - p = 5/4 - r;
19110
19111     - while (x <= y)
19112     {
19113         - if (p < 0)
19114         -     p += (4*x) + 6;
19115         - else
19116         {
19117             - p += (2*(x-y)) + 5;
19118             - y--;
19119         }
19120         - x++;
19121         - for (; letra[c0]; c0++)
19122         {
19123             - switch (c0 % 8)
19124             {
19125                 case 0:
19126                     printf("\033[%d;%dH%c", a+x, b+y, letra[c0]);
19127                     break;
19128                 case 1:
19129                     printf("\033[%d;%dH%c", a-x, b+y, letra[c0]);
19130

```

```

19131 | -         break;
19132 | -     case 2:
19133 | -         printf("\033[%d;%dH%c", a+x, b-y, letra[c0]);
19134 | -         break;
19135 | -     case 3:
19136 | -         printf("\033[%d;%dH%c", a+x, b-y, letra[c0]);
19137 | -         break;
19138 | -     case 4:
19139 | -         printf("\033[%d;%dH%c", a+x, b+y, letra[c0]);
19140 | -         break;
19141 | -     case 5:
19142 | -         printf("\033[%d;%dH%c", a+x, b-y, letra[c0]);
19143 | -         break;
19144 | -     case 6:
19145 | -         printf("\033[%d;%dH%c", a-x, b+y, letra[c0]);
19146 | -         break;
19147 | -     case 7:
19148 | -         printf("\033[%d;%dH%c", a-x, b-y, letra[c0]);
19149 | -     }
19150 | -     getc(stdin);
19151 | - }
19152 | - }
19153 | + conjord_insere_elemento(c, n->k, n->d);
19154 | + conjord_une_helper(c, n->l);
19155 | + conjord_une_helper(c, n->r);
19156 | }
19157 |
19158 | -void conjord_imprime_venn(conjord *c, int pos)
19159 | +/**
19160 | + * une dois conjuntos formando um terceiro
19161 | + * como resultado desta operacao retornando seu ponteiro
19162 | + * a seguir
19163 | + */
19164 | +conjord *conjord_une(conjord *a, conjord *b)
19165 | +{
19166 | +    if (c && c->pr)
19167 | +    {
19168 | +        if (pos == 2)
19169 | +            printf("\033[12;55H");
19170 | +        else if (pos == 1)
19171 | +            printf("\033[12;15H");
19172 | +        else
19173 | +            printf("\033[18;40H");
19174 | +        /* tricky */
19175 | +        conjord_retorna_elementos(c->pr);
19176 | +        conjord_imprime_venn_helper(buffer);
19177 | +    }
19178 | +    conjord *cj_c = conjord_cria("C");
19179 | +    conjord_une_helper(cj_c, a->pr);
19180 | +    conjord_une_helper(cj_c, b->pr);
19181 | +    return cj_c;
19182 | +}
19183 |
19184 | +static void conjord_subtrai_helper(conjord *c, rb_n *n)
19185 | +{
19186 | +    if (n == &rb_s)
19187 | +        return;
19188 | +
19189 | +    conjord_deleta_elemento(c, n->k);
19190 | +    conjord_subtrai_helper(c, n->l);
19191 | +    conjord_subtrai_helper(c, n->r);
19192 | +}
19193 |
19194 | +/**
19195 | + * subtrai dois conjuntos formando um terceiro
19196 | + * como resultado desta operacao retornando seu ponteiro
19197 | + * a seguir
19198 | + */
19199 | +conjord *conjord_subtrai(conjord *a, conjord *b)
19200 | +{
19201 | +    conjord *cj_c = conjord_cria("C");
19202 | +    cj_c->pr = &rb_s;
19203 | +    conjord_une_helper(cj_c, a->pr);
19204 | +    conjord_subtrai_helper(cj_c, b->pr);
19205 | +    return cj_c;
19206 | +}
19207 |
19208 |
19209 |
19210 | diff --git a/src/testa_item_3.c b/src/testa_item_3.c
19211 | index b6dc39b..178608e 100644
19212 | --- a/src/testa_item_3.c
19213 | +++ b/src/testa_item_3.c
19214 | @@ -63,16 +63,17 @@ void testa_item_3(const char *m)
19215 |     conjord *cj_a = conjord_cria("A");
19216 |     conjord *cj_b = conjord_cria("B");
19217 |     int imp, pos0 = 0, pos1 = 1, pos2 = 2;
19218 |     char elem0[] = "ABCDEFGHJKLMNOP";
19219 |     char elem1[] = "JKLMNOPQRSTUVWXYZ";
19220 |     char elem0[] = "ABCDEF";
19221 |     char elem1[] = "DEFGHI";
19222 |
19223 |     LTELA;
19224 |     adiciona_venn(cj_a, pos1, elem0);
19225 |     adiciona_venn(cj_b, pos2, elem1);
19226 |     conjord_imprime_conjunto(cj_a);
19227 |     CM;
19228 |     // conjord_imprime_conjunto(cj_b);
19229 |     conjord_imprime_venn(cj_a, pos0);
19230 |     printf("\n");
19231 |     conjord_imprime_venn(cj_b, pos0);
19232 |     printf("\n");
19233 |     conjord_imprime_venn(conjord_une(cj_a, cj_b), pos0);
19234 |     INFO("item 3.1 operacoes com conjuntos");
19235 |     CM;
19236 | }
19237 | @@ -80,7 +81,7 @@ void testa_item_3(const char *m)
19238 | void imprime_meu_tipo(void *n)
19239 | {
19240 |     if(n)
19241 |         printf("%c \n", ((meu_tipo3 *)n)->letra);
19242 |     printf("%c", ((meu_tipo3 *)n)->letra);
19243 | }
19244 |
19245 |
19246 |

```

```

19247 | commit f44b022e519b3cc1c5345abb01cb2ebad6a54256
19248 | Author: Joilnen <joilnen@gmail.com>
19249 | Date: Sat Jan 8 15:37:44 2022 -0300
19250 |
19251 | trocado a checagem de NULL por noh sentinela dentro do TAD, caso quebre tudo
19252 | retornar voltar ao que era
19253 |
19254 | diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19255 | index 22e4a61..97ca104 100644
19256 | --- a/src/conjunto_ordenado.c
19257 | +++ b/src/conjunto_ordenado.c
19258 | @@ -12,7 +12,7 @@ static unsigned int r_counter = 0;
19259 | conjord *conjord_cria(char *nome)
19260 | {
19261 |     conjord *c = (conjord *) malloc (sizeof(conjord));
19262 |     c->pr = NULL;
19263 |     c->pr = &rb_s;
19264 |     c->idx = r_counter;
19265 |     c->nome = nome;
19266 |     c->cardinalidade = 0;
19267 |     @@ -117,7 +117,7 @@ int conjord_esvazia(conjord *c)
19268 |         conjord_esvazia_helper(c->pr);
19269 |         c->cardinalidade = 0;
19270 |         free(c->pr);
19271 |         c->pr = NULL;
19272 |         c->pr = &rb_s;
19273 |     }
19274 | }
19275 |
19276 | @@ -189,7 +189,8 @@ conjord *conjord_intersecciona(conjord *a, conjord *b)
19277 |     conjord *cj_c = conjord_cria("C");
19278 |     conjord *cj_d = conjord_cria("D");
19279 |     conjord *cj_e = conjord_cria("E");
19280 |     conjord_une(conjord_une(cj_c, a), b);
19281 |
19282 |     // cj_c = conjord_subtrai(conjord_une(conjord_une(cj_c, a), b), ;
19283 | }
19284 |
19285 | static void conjord_une_helper(conjord *c, rb_n *n)
19286 |
19287 | commit 2dd55415ef310e2951c011b0605d38cc05ad2729
19288 | Author: Joilnen <joilnen@gmail.com>
19289 | Date: Sat Jan 8 15:51:20 2022 -0300
19290 |
19291 | intersecao funcionando
19292 |
19293 | diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19294 | index 97ca104..0dccc039 100644
19295 | --- a/src/conjunto_ordenado.c
19296 | +++ b/src/conjunto_ordenado.c
19297 | @@ -186,11 +186,12 @@ void conjord_imprime_venn(conjord *c, int pos)
19298 | /*
19299 | conjord *conjord_intersecciona(conjord *a, conjord *b)
19300 | {
19301 |     conjord *cj_c = conjord_cria("C");
19302 |     conjord *cj_d = conjord_cria("D");
19303 |     conjord *cj_e = conjord_cria("E");
19304 |
19305 |     // cj_c = conjord_subtrai(conjord_une(conjord_une(cj_c, a), b), ;
19306 |     conjord *cj_a = conjord_cria("A");
19307 |     conjord *aub = conjord_une(conjord_une(cj_a, a), b);
19308 |     conjord *amb = conjord_subtrai(conjord_une(cj_a, a), b);
19309 |     conjord *bma = conjord_subtrai(conjord_une(cj_a, b), a);
19310 |     conjord *ambubma = conjord_une(amb, bma);
19311 |     return conjord_subtrai(aub, ambubma);
19312 | }
19313 |
19314 | static void conjord_une_helper(conjord *c, rb_n *n)
19315 | @@ -234,7 +235,6 @@ static void conjord_subtrai_helper(conjord *c, rb_n *n)
19316 | conjord *conjord_subtrai(conjord *a, conjord *b)
19317 | {
19318 |     conjord *cj_c = conjord_cria("C");
19319 |     cj_c->pr = &rb_s;
19320 |     conjord_une_helper(cj_c, a->pr);
19321 |     conjord_subtrai_helper(cj_c, b->pr);
19322 |     return cj_c;
19323 | }
19324 | diff --git a/src/testa_item_3.c b/src/testa_item_3.c
19325 | index 178608e..87431ce 100644
19326 | --- a/src/testa_item_3.c
19327 | +++ b/src/testa_item_3.c
19328 | @@ -73,7 +73,7 @@ void testa_item_3(const char *m)
19329 |     printf("\n");
19330 |     conjord_imprime_venn(cj_b, pos0);
19331 |     printf("\n");
19332 |     conjord_imprime_venn(conjord_une(cj_a, cj_b), pos0);
19333 |     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b), pos0);
19334 |     INFO("item 3.1 operacoes com conjuntos");
19335 |     CM;
19336 | }
19337 |
19338 | commit 486d5a7efdd52f75117ce8a4de53fa00ec0929ec
19339 | Author: Joilnen <joilnen@gmail.com>
19340 | Date: Sat Jan 8 15:59:18 2022 -0300
19341 |
19342 | todos os itens parecem estar satisfeitos
19343 |
19344 | diff --git a/src/testa_item_3.c b/src/testa_item_3.c
19345 | index 87431ce..31e34fc 100644
19346 | --- a/src/testa_item_3.c
19347 | +++ b/src/testa_item_3.c
19348 | @@ -73,6 +73,24 @@ void testa_item_3(const char *m)
19349 |     printf("\n");
19350 |     conjord_imprime_venn(cj_b, pos0);
19351 |     printf("\n");
19352 |     conjord_imprime_venn(conjord_une(cj_a, cj_b), pos0);
19353 |     +
19354 |     INFO("item 3.1 operacoes com conjuntos");
19355 |     CM;
19356 |
19357 |     LTEL;
19358 |     conjord_imprime_venn(cj_a, pos0);
19359 |     printf("\n");
19360 |     conjord_imprime_venn(cj_b, pos0);
19361 |     printf("\n");
19362 |     conjord_imprime_venn(conjord_subtrai(cj_a, cj_b), pos0);
19363 |     +
19364 |     INFO("item 3.1 operacoes com conjuntos");
19365 |     CM;

```



```

19363 +
19364 +     LTELA;
19365 +     conjord_imprime_venn(cj_a, pos0);
19366 +     printf("\n");
19367 +     conjord_imprime_venn(cj_b, pos0);
19368 +     printf("\n");
19369 +     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b), pos0);
19370 +     INFO(" * item 3.1 operacoes com conjuntos");
19371 +     CM;
19372
19373 commit 22b55cf6650d3b4d7d16f1e755a4a936340d75fd
19374 Author: Joilnen <joilnen@gmail.com>
19375 Date: Sat Jan 8 16:21:43 2022 -0300
19376
19377     info no teste final na arvore
19378
19379 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
19380 index 31e34fc..7d50af4 100644
19381 --- a/src/testa_item_3.c
19382 +++ b/src/testa_item_3.c
19383 @@ -74,7 +74,7 @@ void testa_item_3(const char *m)
19384     conjord_imprime_venn(cj_b, pos0);
19385     printf("\n");
19386     conjord_imprime_venn(conjord_une(cj_a, cj_b), pos0);
19387 -     INFO(" * item 3.1 operacoes com conjuntos");
19388 +     INFO(" * item 3.1 uniao de conjuntos");
19389     CM;
19390
19391     LTELA;
19392 @@ -83,7 +83,7 @@ void testa_item_3(const char *m)
19393     conjord_imprime_venn(cj_b, pos0);
19394     printf("\n");
19395     conjord_imprime_venn(conjord_subtrai(cj_a, cj_b), pos0);
19396 -     INFO(" * item 3.1 operacoes com conjuntos");
19397 +     INFO(" * item 3.1 subtracao de conjuntos");
19398     CM;
19399
19400     LTELA;
19401 @@ -92,7 +92,13 @@ void testa_item_3(const char *m)
19402     conjord_imprime_venn(cj_b, pos0);
19403     printf("\n");
19404     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b), pos0);
19405 -     INFO(" * item 3.1 operacoes com conjuntos");
19406 +     INFO(" * item 3.1 interseccao com conjuntos");
19407     CM;
19408 +
19409 +     LTELA;
19410 +     tela_checklist(1 << I3_1);
19411 +     tela_checklist(1 << I3_2);
19412 +     tela_checklist(1 << I3_3);
19413 +     CM;
19414 }
19415
19416 diff --git a/src/testa_rb.c b/src/testa_rb.c
19417 index d218910..c9bbae7 100644
19418 --- a/src/testa_rb.c
19419 +++ b/src/testa_rb.c
19420 @@ -12,9 +12,20 @@
19421
19422 void tela_testa_rb(const char *m)
19423 {
19424 +     int l = 5, c = 24, f;
19425 +     char mens[6][48] =
19426 +     {
19427 +         S_UNDERL"serah efetuado os testes:"S_NORM,
19428 +         "insercao e remocao na arvore",
19429 +         "nao foi pedido mas me ajudou a",
19430 +         "checar a correcao da arvore"
19431 +     };
19432     LTELA;
19433     printf("Testa rb\n");
19434     for (f = 0; f < 4; f++)
19435         CKLSITEM(l++, c, 0, mens[f]);
19436     INFO_FILE(__FILE__);
19437     CM;
19438 }
19439
19440 /**
19441
19442 commit 6f6086039095bdf0b9edd3a663bc20017c5b8166
19443 Author: Joilnen <joilnen@gmail.com>
19444 Date: Sat Jan 8 16:24:55 2022 -0300
19445
19446     consertando uma mensagem errada
19447
19448 diff --git a/src/testa_rb.c b/src/testa_rb.c
19449 index c9bbae7..d8e2a09 100644
19450 --- a/src/testa_rb.c
19451 +++ b/src/testa_rb.c
19452 @@ -16,7 +16,7 @@ void tela_testa_rb(const char *m)
19453     char mens[6][48] =
19454     {
19455         S_UNDERL"serah efetuado os testes:"S_NORM,
19456 -         "insercao e remocao na arvore",
19457 +         "insersao e remocao na arvore",
19458         "nao foi pedido mas me ajudou a",
19459         "checar a correcao da arvore"
19460     };
19461
19462 commit 116562333b47be7503932a0e77e4193e850a9027
19463 Author: Joilnen <joilnen@gmail.com>
19464 Date: Sat Jan 8 16:40:37 2022 -0300
19465
19466     limpando codigo nao utilizado
19467
19468 diff --git a/src/testa_rb.c b/src/testa_rb.c
19469 index d8e2a09..77edb91 100644
19470 --- a/src/testa_rb.c
19471 +++ b/src/testa_rb.c
19472 @@ -111,263 +111,6 @@ void testa_arvore_rb(const char *m)
19473     CM;
19474 }
19475
19476 LTELA;
19477
19478 /**

```

```

19479 - * cria noh raiz, utiliza a tecnica de macro pois simplifica e deixa
19480 - * uma interface mais intuitiva ao usuario da biblioteca
19481 - * @parametro 0 nome variavel do noh raiz
19482 - * @parametro 1 chave do noh raiz
19483 - * @parametro 2 dado criado neste noh, pode ser de qqr tipo
19484 - */
19485 -// rb_n *conjunto_a = rb_cr(10, &mt0);
19486 -//
19487 -// LTELA;
19488 -// rb_pr(conjunto_a, TCOLOR);
19489 -// printf("\n");
19490 -// printf("= arvore =====\n");
19491 -// conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19492 -// printf("= =====\n");
19493 -// printf("%s", m);
19494 -// getc(stdin);
19495 -//
19496 -// meu_tipo mt8;
19497 -// mt8.nome = (char *) malloc(sizeof("nome para teste") + 1);
19498 -// strncpy(mt8.nome, "nome para teste", sizeof("nome para teste") + 1);
19499 -// mt8.telefone = (char *) malloc(sizeof("999999999") + 1);
19500 -// strncpy(mt8.telefone, "999999999", sizeof("999999999") + 1);
19501 -//
19502 -// rb_n *novo_noh = (rb_n *) malloc (sizeof(rb_n));
19503 -// novo_noh->k = 9;
19504 -// novo_noh->d = &mt8;
19505 -//
19506 -// rb_insert(&conjunto_a, novo_noh);
19507 -//
19508 -// meu_tipo mt1;
19509 -// mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
19510 -// strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
19511 -// mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
19512 -// strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
19513 -//
19514 -// novo_noh = (rb_n *) malloc (sizeof(rb_n));
19515 -// novo_noh->k = 9;
19516 -// novo_noh->d = &mt1;
19517 -//
19518 -// rb_insert(&conjunto_a, novo_noh);
19519 -//
19520 -// LTELA;
19521 -// rb_pr(conjunto_a, TCOLOR);
19522 -// printf("\n");
19523 -// printf("= arvore =====\n");
19524 -// conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19525 -// printf("= =====\n");
19526 -// printf("%s", m);
19527 -// getc(stdin);
19528 -//
19529 -// meu_tipo mt2;
19530 -// mt2.nome = (char *) malloc(sizeof("nome para teste") + 1);
19531 -// strncpy(mt2.nome, "nome para teste", sizeof("nome para teste") + 1);
19532 -// mt2.telefone = (char *) malloc(sizeof("999999999") + 1);
19533 -// strncpy(mt2.telefone, "999999999", sizeof("999999999") + 1);
19534 -//
19535 -// novo_noh = (rb_n *) malloc (sizeof(rb_n));
19536 -// novo_noh->k = 11;
19537 -// novo_noh->d = &mt2;
19538 -// rb_insert(&conjunto_a, novo_noh);
19539 -//
19540 -// LTELA;
19541 -// rb_pr(conjunto_a, TCOLOR);
19542 -// printf("\n");
19543 -// printf("= arvore =====\n");
19544 -// conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19545 -// printf("= =====\n");
19546 -// printf("%s", m);
19547 -// getc(stdin);
19548 -//
19549 -// meu_tipo mt3;
19550 -// mt3.nome = (char *) malloc(sizeof("nome para teste") + 1);
19551 -// strncpy(mt3.nome, "nome para teste", sizeof("nome para teste") + 1);
19552 -// mt3.telefone = (char *) malloc(sizeof("999999999") + 1);
19553 -// strncpy(mt3.telefone, "999999999", sizeof("999999999") + 1);
19554 -//
19555 -// novo_noh = (rb_n *) malloc (sizeof(rb_n));
19556 -// novo_noh->k = 8;
19557 -// novo_noh->d = &mt3;
19558 -// rb_insert(&conjunto_a, novo_noh);
19559 -//
19560 -// LTELA;
19561 -// rb_pr(conjunto_a, TCOLOR);
19562 -// printf("\n");
19563 -// printf("= arvore =====\n");
19564 -// conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19565 -// printf("= =====\n");
19566 -// printf("%s", m);
19567 -// getc(stdin);
19568 -//
19569 -// meu_tipo mt4;
19570 -// mt4.nome = (char *) malloc(sizeof("nome para teste") + 1);
19571 -// strncpy(mt4.nome, "nome para teste", sizeof("nome para teste") + 1);
19572 -// mt4.telefone = (char *) malloc(sizeof("999999999") + 1);
19573 -// strncpy(mt4.telefone, "999999999", sizeof("999999999") + 1);
19574 -//
19575 -// novo_noh = (rb_n *) malloc (sizeof(rb_n));
19576 -// novo_noh->k = 6;
19577 -// novo_noh->d = &mt4;
19578 -// rb_insert(&conjunto_a, novo_noh);
19579 -//
19580 -// LTELA;
19581 -// rb_pr(conjunto_a, TCOLOR);
19582 -// printf("\n");
19583 -// printf("= arvore =====\n");
19584 -// conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19585 -// printf("= =====\n");
19586 -// printf("%s", m);
19587 -// getc(stdin);
19588 -//
19589 -// meu_tipo mt5;
19590 -// mt5.nome = (char *) malloc(sizeof("nome para teste") + 1);
19591 -// strncpy(mt5.nome, "nome para teste", sizeof("nome para teste") + 1);
19592 -// mt5.telefone = (char *) malloc(sizeof("999999999") + 1);
19593 -// strncpy(mt5.telefone, "999999999", sizeof("999999999") + 1);
19594 -//

```

```

19595 -//      novo_noh = (rb_n *) malloc (sizeof(rb_n));
19596 -//      novo_noh->k = 5;
19597 -//      novo_noh->d = &mt5;
19598 -//      rb_insert(&conjunto_a, novo_noh);
19599 -//
19600 -//      LTELA;
19601 -//      rb_pr(conjunto_a, TCOLOR);
19602 -//      printf("\n");
19603 -//      printf("= arvore =====\n");
19604 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19605 -//      printf("= =====\n");
19606 -//      printf("%s", m);
19607 -//
19608 -//      neu_tipo mt6;
19609 -//      mt6.nome = (char *) malloc(sizeof("nome para teste") + 1);
19610 -//      strncpy(mt6.nome, "nome para teste", sizeof("nome para teste") + 1);
19611 -//      mt6.telefone = (char *) malloc(sizeof("999999999") + 1);
19612 -//      strncpy(mt6.telefone, "999999999", sizeof("999999999") + 1);
19613 -//
19614 -//      novo_noh = (rb_n *) malloc (sizeof(rb_n));
19615 -//      novo_noh->k = 4;
19616 -//      novo_noh->d = &mt6;
19617 -//      rb_insert(&conjunto_a, novo_noh);
19618 -//
19619 -//      LTELA;
19620 -//      rb_pr(conjunto_a, TCOLOR);
19621 -//      printf("\n");
19622 -//      printf("= arvore =====\n");
19623 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19624 -//      printf("= =====\n");
19625 -//      printf("%s", m);
19626 -//      getc(stdin);
19627 -//
19628 -//      neu_tipo mt7;
19629 -//      mt7.nome = (char *) malloc(sizeof("nome para teste") + 1);
19630 -//      strncpy(mt7.nome, "nome para teste", sizeof("nome para teste") + 1);
19631 -//      mt7.telefone = (char *) malloc(sizeof("999999999") + 1);
19632 -//      strncpy(mt7.telefone, "999999999", sizeof("999999999") + 1);
19633 -//
19634 -//      novo_noh = (rb_n *) malloc (sizeof(rb_n));
19635 -//      novo_noh->k = 3;
19636 -//      novo_noh->d = &mt7;
19637 -//      rb_insert(&conjunto_a, novo_noh);
19638 -//
19639 -//      LTELA;
19640 -//      rb_pr(conjunto_a, TCOLOR);
19641 -//      printf("\n");
19642 -//      printf("= arvore =====\n");
19643 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19644 -//      printf("= =====\n");
19645 -//      printf("%s", m);
19646 -//      getc(stdin);
19647 -//
19648 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 3));
19649 -//      printf("\n");
19650 -//      LTELA;
19651 -//      rb_pr(conjunto_a, TCOLOR);
19652 -//      printf("\n");
19653 -//      printf("= arvore =====\n");
19654 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19655 -//      printf("= =====\n");
19656 -//      printf("%s", m);
19657 -//      getc(stdin);
19658 -//
19659 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 11));
19660 -//      LTELA;
19661 -//      rb_pr(conjunto_a, TCOLOR);
19662 -//      printf("\n");
19663 -//      printf("= arvore =====\n");
19664 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19665 -//      printf("= =====\n");
19666 -//      printf("%s", m);
19667 -//      getc(stdin);
19668 -//
19669 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 4));
19670 -//      LTELA;
19671 -//      rb_pr(conjunto_a, TCOLOR);
19672 -//      printf("\n");
19673 -//      printf("= arvore =====\n");
19674 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19675 -//      printf("= =====\n");
19676 -//      printf("%s", m);
19677 -//      getc(stdin);
19678 -//
19679 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 8));
19680 -//      LTELA;
19681 -//      rb_pr(conjunto_a, TCOLOR);
19682 -//      printf("\n");
19683 -//      printf("= arvore =====\n");
19684 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19685 -//      printf("= =====\n");
19686 -//      printf("%s", m);
19687 -//      getc(stdin);
19688 -//
19689 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 9));
19690 -//      LTELA;
19691 -//      rb_pr(conjunto_a, TCOLOR);
19692 -//      printf("\n");
19693 -//      printf("= arvore =====\n");
19694 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19695 -//      printf("= =====\n");
19696 -//      printf("%s", m);
19697 -//      getc(stdin);
19698 -//      rb_pr(conjunto_a, TCOLOR);
19699 -//      printf("\n");
19700 -//
19701 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 5));
19702 -//      LTELA;
19703 -//      rb_pr(conjunto_a, TCOLOR);
19704 -//      printf("\n");
19705 -//      printf("= arvore =====\n");
19706 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19707 -//      printf("= =====\n");
19708 -//      printf("%s", m);
19709 -//      getc(stdin);
19710 -//

```

```

19711 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 10));
19712 -//      LTEL;
19713 -//      rb_pr(conjunto_a, TCOLOR);
19714 -//      printf("\n");
19715 -//      printf("= arvore =====\n");
19716 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19717 -//      printf("= =====\n");
19718 -//      printf("%s", m);
19719 -//      getc(stdin);
19720 -//
19721 -//      rb_delete(&conjunto_a, rb_search(conjunto_a, 6));
19722 -//      LTEL;
19723 -//      printf("nada\n");
19724 -//      printf("= arvore =====\n");
19725 -//      conjord_imprime_rb(conjunto_a, 1, TCOLOR);
19726 -//      printf("= =====\n");
19727 -//      rb_pr(conjunto_a, TCOLOR);
19728 -//      printf("\n");
19729 -
19730 -    /**
19731 -     * testando apenas rb no momento
19732 -     conjord *conj_a = conjord_cria();
19733 -     printf("eh vazio %d", conjord_eh_vazio(conj_a));
19734 -     */
19735 -}
19736
19737 /**
19738 commit 8ea985f270e97d3ab18d8035e9621883e8f5eff7
19739 Author: Joilnen <joilnen@gmail.com>
19740 Date: Sun Jan 9 06:29:26 2022 -0300
19741
19742     removendo parametro desnecessario
19743
19744 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19745 index 0dccc039..6602160 100644
19746 --- a/src/conjunto_ordenado.c
19747 +++ b/src/conjunto_ordenado.c
19748 @@ -168,7 +168,7 @@ static void conjord_imprime_venn_helper(rb_n *n)
19749     conjord_imprime_venn_helper(n->r);
19750 }
19751
19752 -void conjord_imprime_venn(conjord *c, int pos)
19753 +void conjord_imprime_venn(conjord *c)
19754 {
19755     if (c && c->r)
19756     {
19757 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
19758 index e788918..ffa7160 100644
19759 --- a/src/conjunto_ordenado.h
19760 +++ b/src/conjunto_ordenado.h
19761 @@ -127,7 +127,7 @@ int conjord_seta_imprime_elemento(void (*i)(void *n));
19762
19763 void conjord_imprime_conjunto(conjord *c);
19764
19765 -void conjord_imprime_venn(conjord *c, int pos);
19766 +void conjord_imprime_venn(conjord *c);
19767
19768
19769 /**
19770 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
19771 index 7d50af4..aadf98d 100644
19772 --- a/src/testa_item_3.c
19773 +++ b/src/testa_item_3.c
19774 @@ -69,29 +69,29 @@ void testa_item_3(const char *m)
19775     LTEL;
19776     adiciona_venn(cj_a, pos1, elem0);
19777     adiciona_venn(cj_b, pos2, elem1);
19778     conjord_imprime_venn(cj_a, pos0);
19779     conjord_imprime_venn(cj_a);
19780     printf("\n");
19781     conjord_imprime_venn(cj_b, pos0);
19782     conjord_imprime_venn(cj_b);
19783     printf("\n");
19784     conjord_imprime_venn(conjord_une(cj_a, cj_b), pos0);
19785     conjord_imprime_venn(conjord_une(cj_a, cj_b));
19786     INFO("item 3.1 uniao de conjuntos");
19787     CM;
19788
19789     LTEL;
19790     conjord_imprime_venn(cj_a, pos0);
19791     conjord_imprime_venn(cj_a);
19792     printf("\n");
19793     conjord_imprime_venn(cj_b, pos0);
19794     conjord_imprime_venn(cj_b);
19795     printf("\n");
19796     conjord_imprime_venn(conjord_subtrai(cj_a, cj_b), pos0);
19797     conjord_imprime_venn(conjord_subtrai(cj_a, cj_b));
19798     INFO("item 3.1 subtracao de conjuntos");
19799     CM;
19800
19801     LTEL;
19802     conjord_imprime_venn(cj_a, pos0);
19803     conjord_imprime_venn(cj_a);
19804     printf("\n");
19805     conjord_imprime_venn(cj_b, pos0);
19806     conjord_imprime_venn(cj_b);
19807     printf("\n");
19808     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b), pos0);
19809     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b));
19810     INFO("item 3.1 interseccao com conjuntos");
19811     CM;
19812
19813
19814 commit b81c6a1e2353669e3b891a47fe65f105428a034e
19815 Author: Joilnen <joilnen@gmail.com>
19816 Date: Sun Jan 9 06:54:24 2022 -0300
19817
19818     deletando recursivamente os dados dos nohs da arvore
19819
19820 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19821 index 6602160..40fb827 100644
19822 --- a/src/conjunto_ordenado.c
19823 +++ b/src/conjunto_ordenado.c
19824 @@ -29,7 +29,8 @@ void conjord_destroi(conjord *c)
19825     return;
19826     if (!c->r)

```

```

19827     return;
19828 -   free(c->pr);
19829 +   conjord_esvazia(c);
19830 +   // free(c->pr);
19831   free(c);
19832 }
19833
19834
19835 commit 68b80179418c9d2c3d7a131cdde64670c5419c05
19836 Author: Joilnen <joilnen@gmail.com>
19837 Date: Mon Jan 10 11:40:42 2022 -0300
19838
19839     apenas limpando o codigo
19840
19841 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
19842 index 925553b..815cfdb 100644
19843 --- a/src/testa_item_1.c
19844 +++ b/src/testa_item_1.c
19845 @@ -62,12 +62,12 @@ void testa_item_1(const char *m)
19846     CM;
19847
19848     LTELA;
19849 -   conjord_imprime_arvore_rb(cj_a, 1);
19850 +   conjord_imprime_arvore_rb(cj_a, 0);
19851     INFO("* item 4 mostra da arvore atual");
19852     CM;
19853
19854     LTELA;
19855 -   tela_checklist((1 << I4));
19856 +   tela_checklist(1 << I4);
19857     CM;
19858
19859     LTELA;
19860 @@ -76,7 +76,7 @@ void testa_item_1(const char *m)
19861     CM;
19862
19863     LTELA;
19864 -   tela_checklist((1 << I1_1));
19865 +   tela_checklist(1 << I1_1);
19866     CM;
19867
19868     conjord_destroi(cj_a);
19869
19870 commit 656484f2f5dd5218eda052e794dbaaacc3b2d53a
19871 Author: Joilnen <joilnen@gmail.com>
19872 Date: Mon Jan 10 11:58:22 2022 -0300
19873
19874     implementado checagem se elemento existe no conjord
19875
19876 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19877 index 40fb827..cb26a7b 100644
19878 --- a/src/conjunto_ordenado.c
19879 +++ b/src/conjunto_ordenado.c
19880 @@ -241,5 +241,8 @@ conjord_subtrai(conjord *a, conjord *b)
19881     return cj_c;
19882 }
19883
19884 -
19885 +int conjord_existe_elemento(conjord *a, int chave)
19886 +{
19887 +   return rb_search(a->pr, chave) ? 1: 0;
19888 +}
19889
19890
19891 commit 72803c65324b042f3f5c89821e3ffde945063bfb
19892 Author: Joilnen <joilnen@gmail.com>
19893 Date: Mon Jan 10 12:03:18 2022 -0300
19894
19895     adaptando checagem de vazio para usar sentinela
19896
19897 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19898 index cb26a7b..5cc561d 100644
19899 --- a/src/conjunto_ordenado.c
19900 +++ b/src/conjunto_ordenado.c
19901 @@ -80,21 +80,12 @@ int conjord_cardinalidade(conjord *c)
19902     return -1;
19903 }
19904
19905 -/**
19906 - * imprime elementos do conjunto subitem 1 do item 2
19907 - * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
19908 - */
19909 -void conjord_imprime()
19910 -{
19911 -}
19912 -
19913 -
19914 -/**
19915 - * testa se o conjunto estah vazio
19916 - */
19917 -int conjord_eh_vazio(const conjord *a)
19918 -{
19919 -   return !a->pr;
19920 +   return a->pr == &rb_s;
19921 }
19922
19923 static void conjord_esvazia_helper(rb_n *r)
19924 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
19925 index ffa7160..cdc8fe3 100644
19926 --- a/src/conjunto_ordenado.h
19927 +++ b/src/conjunto_ordenado.h
19928 @@ -39,12 +39,6 @@ int conjord_insere_elemento(conjord *a, int chave, void *elemento);
19929 -
19930 -void conjord_deleta_elemento(conjord *a, int chave);
19931 -
19932 -/**
19933 - * imprime elementos do conjunto subitem 1 do item 2
19934 - * "imprimir os elementos de um conjunto, em ordem crescente das chaves"
19935 - */
19936 -void conjord_imprime();
19937 -
19938 -/**
19939 - * retorna a cardinalidade do conjunto
19940 - */
19941
19942 commit 0853d6514a22918fc5a0b0e5b58c65f5ee82dec4

```

```

19943 Author: Joilnen <joilnen@gmail.com>
19944 Date: Mon Jan 10 12:21:54 2022 -0300
19945
19946     adequando ao estilo de escrita do código escolhido olhar na doc
19947
19948 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
19949 index 5cc561d..8317cdf 100644
19950 --- a/src/conjunto_ordenado.c
19951 +++ b/src/conjunto_ordenado.c
19952 @@ -25,9 +25,9 @@ conjord *conjord_cria(char *nome)
19953 */
19954 void conjord_destrói(conjord *c)
19955 {
19956     if(!c)
19957 +     if(!c)
19958         return;
19959 -     if(!c->pr)
19960 +     if(!c->pr)
19961         return;
19962     conjord_esvazia(c);
19963     // free(c->pr);
19964 @@ -39,12 +39,12 @@ void conjord_destrói(conjord *c)
19965 */
19966 int conjord_insere_elemento(conjord *a, int chave, void *elemento)
19967 {
19968 -     if (a->pr)
19969 +     if (a->pr != &rb_s)
19970     {
19971         rb_n *n = (rb_n *) malloc(sizeof(rb_n));
19972         n->k = chave;
19973         n->d = elemento;
19974 -         if(!rb_insert(&a->pr, n))
19975 +         if (!rb_insert(&a->pr, n))
19976             a->cardinalidade++;
19977         else
19978             return -1;
19979 @@ -63,7 +63,7 @@ void conjord_deleta_elemento(conjord *a, int chave)
19980 {
19981     rb_n *t = rb_search(a->pr, chave);
19982
19983 -     if(t)
19984 +     if (t)
19985     {
19986         rb_delete(&a->pr, t);
19987         a->cardinalidade--;
19988 diff --git a/src/red_black.c b/src/red_black.c
19989 index 447027f..6349dc7 100644
19990 --- a/src/red_black.c
19991 +++ b/src/red_black.c
19992 @@ -155,7 +155,7 @@ static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
19993     u->p->l = v;
19994     else
19995         u->p->r = v;
19996 -     if(v)
19997 +     if (v)
19998         v->p = u->p;
19999     return 0;
20000 @@ -378,9 +378,9 @@ void rb_pr(rb_n *t, int s, int e)
20001     rb_pr(t->r, s, e);
20002
20003     printf("%*c", s - dist, ' ');
20004 -     if(t->c == BLACK)
20005 +     if (t->c == BLACK)
20006     {
20007         if(e == TCOLOR)
20008 +         if (e == TCOLOR)
20009             printf("\033[44m\033[37m%d\033[0m\n", t->k);
20010         else
20011             printf("%d\n", t->k);
20012 @@ -388,7 +388,7 @@ void rb_pr(rb_n *t, int s, int e)
20013     }
20014     else
20015     {
20016         if(e == TCOLOR)
20017 +         if (e == TCOLOR)
20018             printf("\033[41m\033[37m%d\033[0m\n", t->k);
20019         else
20020             printf("\033[7m%d\033[0m\n", t->k, t->c);
20021 @@ -411,7 +411,7 @@ void rb_pr2(rb_n *t, int s, int e)
20022     // printf(" ");
20023     // printf("%*s", s, "");
20024     printf("%*c", s - dist, ' ');
20025 -     if(t->c == BLACK)
20026 +     if (t->c == BLACK)
20027     {
20028         if(e == TCOLOR)
20029             printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
20030 diff --git a/src/testa.h b/src/testa.h
20031 index 302e359..dcff849 100644
20032 --- a/src/testa.h
20033 +++ b/src/testa.h
20034 @@ -10,6 +10,7 @@
20035 @@ -10,6 +10,7 @@
20036 #include "testa_item_2.h"
20037 #include "testa_item_3.h"
20038 #include "testa_rb.h"
20039 #include "checklist.h"
20040
20041 #endif
20042
20043 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
20044 index 815cfd8..ac7e322 100644
20045 --- a/src/testa_item_1.c
20046 +++ b/src/testa_item_1.c
20047 @@ -2,8 +2,7 @@
20048 #include <string.h>
20049 #include <time.h>
20050 #include "conjunto_ordenado.h"
20051 #include "testa_item_1.h"
20052 #include "checklist.h"
20053 #include "testa.h"
20054
20055 static void imprime_meu_tipo(void *);
20056
20057 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
20058 index c5c36d7..1de9e99 100644

```

```

20059 --- a/src/testa_item_2.c
20060 +++ b/src/testa_item_2.c
20061 @@ -3,7 +3,6 @@
20062 #include <time.h>
20063 #include "conjunto_ordenado.h"
20064 #include "testa.h"
20065 -#include "checklist.h"
20066
20067 /*
20068  * aluno: joilnen leite
20069 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
20070 index aadf98d..4eab32f 100644
20071 --- a/src/testa_item_3.c
20072 +++ b/src/testa_item_3.c
20073 @@ -3,7 +3,6 @@
20074 #include <time.h>
20075 #include "conjunto_ordenado.h"
20076 #include "testa.h"
20077 -#include "checklist.h"
20078
20079 /*
20080  * aluno: joilnen leite
20081 */
20082 @@ -104,7 +103,7 @@ void testa_item_3(const char *m)
20083
20084 void imprime_meu_tipo(void *n)
20085 {
20086     if(n)
20087         printf("%c", ((meu_tipo3 *)n)->letra);
20088 }
20089
20090 diff --git a/src/testa_rb.c b/src/testa_rb.c
20091 index 77adb91..98b0574 100644
20092 --- a/src/testa_rb.c
20093 +++ b/src/testa_rb.c
20094 @@ -28,12 +28,6 @@ void tela_testa_rb(const char *m)
20095 }
20096
20097
20098 -/**
20099  * funcao de impressao do dipo especifico criado pelo o usuario da biblioteca
20100  * @noh rb
20101  */
20102 -static void imprime_meu_tipo(void *);
20103 -
20104 /**
20105  * funcao main de teste separada do codigo da biblioteca como especificado
20106  */
20107 @@ -47,44 +41,13 @@ void testa_arvore_rb(const char *m)
20108     char *telefone;
20109     meu_tipo;
20110
20111     /**
20112     * criacao do dado para o tipo do usuario
20113     */
20114     meu_tipo mt0;
20115     mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
20116     strncpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
20117     mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
20118     strncpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
20119
20120     /**
20121     * seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao
20122     */
20123     conjord_setar_imprime_elemento(imprime_meu_tipo);
20124
20125     conjord *cj_a = conjord_cria("A");
20126     conjord_inserir_elemento(cj_a, 10, &mt0);
20127     // LTELA;
20128     // conjord_imprime_arvore_rb(cj_a, 1);
20129     // CM;
20130
20131     meu_tipo mt1;
20132     mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
20133     strncpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
20134     mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
20135     strncpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
20136
20137     conjord_inserir_elemento(cj_a, 11, &mt1);
20138     // LTELA;
20139     // conjord_imprime_arvore_rb(cj_a, 1);
20140     // CM;
20141
20142     int ctest;
20143     conjord *cj_b = conjord_cria("B");
20144     const int size_n = 21;
20145     int test_n[size_n];
20146     for (ctest = 0; ctest < size_n; ++ctest)
20147     {
20148         if(ctest < 10) test_n[ctest] = ctest + 11;
20149         if (ctest < 10) test_n[ctest] = ctest + 11;
20150         else test_n[ctest] = ctest - 10;
20151     }
20152
20153 @@ -113,13 +76,4 @@ void testa_arvore_rb(const char *m)
20154     LTELA;
20155 }
20156
20157 -/**
20158  * definicao da funcao de impressao estipulado pelo cliente, aqui apenas um
20159  * exemplo
20160  */
20161 -void imprime_meu_tipo(void *n)
20162 -{
20163     -}
20164
20165
20166
20167
20168 commit 600b1df008af9c670179151f861b7d0f07370b52
20169 Author: Joilnen <joilnen@gmail.com>
20170 Date: Mon Jan 10 12:30:40 2022 -0300
20171
20172     corrigindo uma palavra
20173
20174 diff --git a/src/testa_rb.c b/src/testa_rb.c

```

```

20175 index 98b0574..470e325 100644
20176 --- a/src/testa_rb.c
20177 +++ b/src/testa_rb.c
20178 @@ -16,7 +16,7 @@ void tela_testa_rb(const char *m)
20179     char mens[6][48] =
20180     {
20181         S_UNDERL"serah efetuado os testes:"S_NORM,
20182         "insercao e remocao na arvore",
20183         "insercao e remocao na arvore",
20184         "nao foi pedido mas me ajudou a",
20185         "checar a correcao da arvore"
20186     };
20187
20188 commit 9793c61f0a2ff9873af542d37c4dfe166e80340b
20189 Author: Joilnen <joilnen@gmail.com>
20190 Date: Mon Jan 10 13:00:35 2022 -0300
20191
20192     removendo funcoes desnecessarias
20193
20194 diff --git a/src/red_black.c b/src/red_black.c
20195 index 6349dc7..8aee293 100644
20196 --- a/src/red_black.c
20197 +++ b/src/red_black.c
20198 @@ -309,21 +309,6 @@ int rb_delete(rb_n **t, rb_n *z)
20199     return 0;
20200 }
20201
20202 -rb_n* rb_intersec(rb_n *a, rb_n *b)
20203 -{
20204     return NULL;
20205 -}
20206
20207 -rb_n* rb_une(rb_n *a, rb_n *b)
20208 -{
20209     return NULL;
20210 -}
20211
20212 -rb_n* rb_subtrai(rb_n *a, rb_n *b)
20213 -{
20214     return NULL;
20215 -}
20216
20217 int rb_destroi(rb_n **a)
20218 {
20219     return 0;
20220
20221 diff --git a/src/red_black.h b/src/red_black.h
20222 index 98705dc..4a3331 100644
20223 --- a/src/red_black.h
20224 +++ b/src/red_black.h
20225 @@ -94,30 +94,6 @@ int rb_delete(rb_n **t, rb_n *z);
20226
20227 int rb_exite_elem(rb_n *a, int k);
20228
20229 /**
20230  * @param a noh raiz da arvore
20231  * @param b noh raiz da arvore a ser operada
20232  * @return retorna ponteiro para uma nova arvore fruto da interseccao
20233  * das duas arvores @param a e @param b
20234  */
20235 -rb_n* rb_intersec(rb_n *a, rb_n *b);
20236
20237 /**
20238  * @param a noh raiz da arvore
20239  * @param b noh raiz da arvore a ser operada
20240  * retorna ponteiro para uma nova arvore fruto da uniao
20241  * das duas arvores @param a e @param b
20242  */
20243 -rb_n* rb_une(rb_n *a, rb_n *b);
20244
20245 /**
20246  * @param a noh raiz da arvore
20247  * @param b noh raiz da arvore a ser operada
20248  * retorna ponteiro para uma nova arvore fruto da subtracao
20249  * das duas arvores @param a e @param b
20250  */
20251 -rb_n* rb_subtrai(rb_n *a, rb_n *b);
20252
20253 /**
20254  * deleta, e desaloca recursos de memoria
20255  * para a arvore apontada por @param a
20256  */
20257 commit df48de679915aea4349b67c31376e75939fa77cf
20258 Author: Joilnen <joilnen@gmail.com>
20259 Date: Tue Jan 11 01:36:43 2022 -0300
20260
20261     testa arvore com numeros randomicos unicos, podendo ser testado em loop
20262
20263 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
20264 index 8317cdf..9914c53 100644
20265 --- a/src/conjunto_ordenado.c
20266 +++ b/src/conjunto_ordenado.c
20267 @@ -118,11 +118,11 @@ void conjord_imprime_arvore_rb(conjord *c, int s)
20268     rb_pr(c->pr, s, TCOLOR);
20269 }
20270
20271 -void conjord_imprime_lista(int *l, int s)
20272 +void conjord_imprime_lista(int *l, int b, int e)
20273 {
20274     int i;
20275     for (i = 0; i < s; i++)
20276         printf(S_LIST, l[i] + 1, l[i]);
20277     for (i = b; i < e; i++)
20278         printf(S_LIST, i + 1, l[i]);
20279 }
20280
20281 void (*conjord_imprime_elemento)(void *n);
20282 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
20283 index cdc8fe3..ec09580 100644
20284 --- a/src/conjunto_ordenado.h
20285 +++ b/src/conjunto_ordenado.h
20286 @@ -110,7 +110,7 @@ void conjord_imprime_rb(rb_n *t, int s, int e);
20287
20288 void conjord_imprime_arvore_rb(conjord *c, int s);
20289
20290 -void conjord_imprime_lista(int *l, int s);
20291 +void conjord_imprime_lista(int *l, int s, int e);

```



```

20291 extern void (*conjord_imprime_elemento)(void *n);
20292 extern void* (*conjord_retorna_elemento)(void *n);
20293 @@ -139,6 +139,7 @@ void conjord_imprime_venn(conjord *c);
20294 #define INFO2(x) printf("%s%s", S_INFO2, x, S_NORM)
20295 #define INFO_ST(x) printf("%s%s%s", S_INV, S_INFO_ST, x, S_NORM)
20296 #define CM printf("%s", m); fflush(stdout); getc(stdin)
20297 #define CMR(x) printf("%s", S_CMR"q + * tecla enter *"); fflush(stdout); x = getc(stdin)
20298 #define CKLSITEM(l,c,b,x) \
20299     if (b) printf(S_CHKLS_DONE, l, c, x); \
20300     else printf(S_CHKLS, l, c, x)
20301 diff --git a/src/jcurses.h b/src/jcurses.h
20302 index 98b5ec3..f18d7ef 100644
20303 --- a/src/jcurses.h
20304 +++ b/src/jcurses.h
20305 @@ -16,6 +16,7 @@
20306 #define LIMPATELA "\033[H\033[J"
20307 #define S_CM "\033[24;62H"
20308 #define S_CMR "\033[24;58H"
20309 #define S_UNDERL "\033[21m"
20310 #define S_AZUL "\033[44m\033[37m"
20311 #define S_VERM "\033[41m\033[37m"
20312 diff --git a/src/testa_rb.c b/src/testa_rb.c
20313 index 470e325..c6a0645 100644
20314 --- a/src/testa_rb.c
20315 +++ b/src/testa_rb.c
20316 @@ -8,6 +8,7 @@
20317 #include "conjunto_ordenado.h"
20318 #include <stdio.h>
20319 #include <string.h>
20320 #include <time.h>
20321 void testa_testa_rb(const char *m)
20322 @@ -28,6 +29,21 @@ void testa_testa_rb(const char *m)
20323 }
20324 +static int pega_unico(int *l, int s)
20325 +{
20326 +    srand(time(NULL));
20327 +    int i = rand() % 99 + 1;
20328 +    int t;
20329 +    for (t = 0; t < s; ++t)
20330 +    {
20331 +        if (i == l[t])
20332 +            t = 0;
20333 +        i = rand() % 99 + 1;
20334 +    }
20335 +    return i;
20336 +}
20337 +/**
20338 + * funcao main de teste separada do codigo da biblioteca como especificado
20339 + */
20340 @@ -45,35 +61,38 @@ void testa_arvore_rb(const char *m)
20341     conjord *cj_b = conjord_cria("B");
20342     const int size_n = 21;
20343     int test_n[size_n];
20344     for (ctest = 0; ctest < size_n; ++ctest)
20345     {
20346         char sai = 10;
20347         while (sai == 10)
20348         {
20349             if (ctest < 10) test_n[ctest] = ctest + 11;
20350             else test_n[ctest] = ctest - 10;
20351         }
20352         for (ctest = 0; ctest < size_n; ++ctest)
20353             test_n[ctest] = pega_unico(test_n, ctest);
20354     }
20355     LATELA;
20356     conjord_imprime_lista(test_n, size_n);
20357     CM;
20358     for (ctest = 0; ctest < size_n; ++ctest)
20359     {
20360         conjord_insere_elemento(cj_b, test_n[ctest], NULL);
20361         LATELA;
20362         conjord_imprime_lista(&test_n[ctest + 1], size_n - ctest - 1);
20363         printf("\033[H");
20364         conjord_imprime_arvore_rb(cj_b, 1);
20365         conjord_imprime_lista(test_n, 0, size_n);
20366         CM;
20367     }
20368     for (ctest = 0; ctest < size_n; ++ctest)
20369     {
20370         conjord_insere_elemento(cj_b, test_n[ctest], NULL);
20371         LATELA;
20372         conjord_imprime_lista(test_n, ctest + 1, size_n);
20373         printf("\033[H");
20374         conjord_imprime_arvore_rb(cj_b, 1);
20375         CM;
20376     }
20377     for (ctest = 0; ctest < size_n; ++ctest)
20378     {
20379         // conjord_imprime_arvore_rb(cj_b, 1);
20380         for (ctest = 0; ctest < size_n; ++ctest)
20381         {
20382             LATELA;
20383             conjord_deleta_elemento(cj_b, test_n[ctest]);
20384             conjord_imprime_lista(test_n, 0, ctest + 1);
20385             printf("\033[H");
20386             conjord_imprime_arvore_rb(cj_b, 1);
20387             CM;
20388         }
20389         LATELA;
20390         conjord_imprime_lista(test_n, ctest + 1);
20391         conjord_deleta_elemento(cj_b, test_n[ctest]);
20392         printf("\033[H");
20393         conjord_imprime_arvore_rb(cj_b, 1);
20394         CM;
20395     }
20396     CM;
20397     CMR(sai);
20398 }
20399 LATELA;
20400 }

```

```

20407
20408
20409
20410 commit 01f945bfa612db99b834c3c5646fd4a80761775b
20411 Author: Joilnen <joilnen@gmail.com>
20412 Date: Tue Jan 11 02:58:54 2022 -0300
20413
20414     consertando geracao randomico unico
20415
20416 diff --git a/src/testa_rb.c b/src/testa_rb.c
20417 index c6a0645..5617e8a 100644
20418 --- a/src/testa_rb.c
20419 +++ b/src/testa_rb.c
20420 @@ -37,8 +37,10 @@ static int pega_unico(int *l, int s)
20421 {
20422     for (t = 0; t < s; ++t)
20423     {
20424         if (i == l[t])
20425         {
20426             t = 0;
20427             i = rand() % 99 + 1;
20428         }
20429     }
20430
20431     return i;
20432
20433 commit 9e7db2fa741433532b3337a8c50132493cc46fec
20434 Author: Joilnen <joilnen@gmail.com>
20435 Date: Tue Jan 11 11:34:18 2022 -0300
20436
20437     consertando a impressao quando nao ha intesercao
20438
20439 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
20440 index 9914c53..ef4d273 100644
20441 --- a/src/conjunto_ordenado.c
20442 +++ b/src/conjunto_ordenado.c
20443 @@ -166,7 +166,8 @@ void conjord_imprime_venn(conjord *c)
20444 {
20445     printf("%s = {", c->nome);
20446     conjord_imprime_venn_helper(c->pr);
20447     printf("\033[D\033[D");
20448     if (c->cardinalidade)
20449     printf("\033[D\033[D");
20450     printf("}");
20451 }
20452
20453 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
20454 index ec09580..3917c96 100644
20455 --- a/src/conjunto_ordenado.h
20456 +++ b/src/conjunto_ordenado.h
20457 @@ -133,7 +133,7 @@ void conjord_imprime_venn(conjord *c);
20458 * foi primeiro implementado com um loop
20459 * mas pensando melhor dessa forma eh mais rapido
20460 */
20461 #define LTELTA printf(LIMPA_TELA)
20462 #define LTELTA printf(S_LTELTA)
20463 #define TIPO_DE_TERMINAL TCOLOR
20464 #define INFO(x) printf("%s%s", S_INFO, x, S_NORM)
20465 #define INFO2(x) printf("%s%s", S_INFO2, x, S_NORM)
20466 diff --git a/src/jcurses.h b/src/jcurses.h
20467 index f18d7ef..724ae59 100644
20468 --- a/src/jcurses.h
20469 +++ b/src/jcurses.h
20470 @@ -14,7 +14,7 @@
20471 * primaria que atende as necessidades apenas dessa avaliacao
20472 */
20473
20474 #define LIMPA_TELA "\033[H\033[J"
20475 #define S_LTELTA "\033[H\033[J"
20476 #define S_CM "\033[24;62H"
20477 #define S_CMR "\033[24;58H"
20478 #define S_UNDERL "\033[21m"
20479 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
20480 index 4eab32f..ba9966d 100644
20481 --- a/src/testa_item_3.c
20482 +++ b/src/testa_item_3.c
20483 @@ -63,7 +63,7 @@ void testa_item_3(const char *m)
20484 conjord *cj_b = conjord_cria("B");
20485 int imp, pos0 = 0, pos1 = 1, pos2 = 2;
20486 char elem0[] = "ABCDEFGH";
20487 char elem1[] = "DEFGHI";
20488 char elem1[] = "GHIJKL";
20489
20490 LTELTA;
20491 adiciona_venn(cj_a, pos1, elem0);
20492 diff --git a/src/testa_rb.c b/src/testa_rb.c
20493 index 5617e8a..149abad 100644
20494 --- a/src/testa_rb.c
20495 +++ b/src/testa_rb.c
20496 @@ -71,6 +71,7 @@ void testa_arvore_rb(const char *m)
20497
20498 LTELTA;
20499 conjord_imprime_lista(test_n, 0, size_n);
20500 INFO("insere/remove 20 elem. chave unica entre 1 a 99");
20501 CM;
20502 for (ctest = 0; ctest < size_n; ++ctest)
20503 {
20504 @@ -79,6 +80,7 @@ void testa_arvore_rb(const char *m)
20505 conjord_imprime_lista(test_n, ctest + 1, size_n);
20506 printf("\033[H");
20507 conjord_imprime_arvore_rb(cj_b, 1);
20508 INFO("insere/remove 20 elem. chave unica entre 1 a 99");
20509 CM;
20510 }
20511
20512 @@ -90,11 +92,13 @@ void testa_arvore_rb(const char *m)
20513 conjord_imprime_lista(test_n, 0, ctest + 1);
20514 printf("\033[H");
20515 conjord_imprime_arvore_rb(cj_b, 1);
20516 INFO("insere/remove 20 elem. chave unica entre 1 a 99");
20517 CM;
20518 }
20519 LTELTA;
20520 CMR(sai);
20521 }
20522 + LTELTA;

```

```

20523 }
20524
20525
20526
20527 commit 502d78db93582a2d9faf75c213ca8bbd27d6680f
20528 Author: Joilnen <joilnen@gmail.com>
20529 Date: Wed Jan 12 04:43:35 2022 -0300
20530
20531 atualizando documentacao
20532
20533 diff --git a/doc/tex/code/inc.c b/doc/tex/code/inc.c
20534 index 96c00d1..31411fd 100644
20535 --- a/doc/tex/code/inc.c
20536 +++ b/doc/tex/code/inc.c
20537 @@ -11,7 +11,7 @@ int rb_insert(rb_n **t, rb_n *z)
20538     * ser alterada nesta função e desejamos que estas alterações
20539     * persistam
20540     */
20541 -    rb_n *y = &rb_s;
20542 +    rb_n *y = &rb_s;
20543     rb_n *x = *t;
20544
20545     /**
20546 diff --git a/doc/tex/red-black-tree.pdf b/doc/tex/red-black-tree.pdf
20547 new file mode 100644
20548 index 0000000..9966f82
20549 Binary files /dev/null and b/doc/tex/red-black-tree.pdf differ
20550 diff --git a/doc/tex/red-black-tree.tex b/doc/tex/red-black-tree.tex
20551 new file mode 100644
20552 index 0000000..cd8afac
20553 --- /dev/null
20554 +++ b/doc/tex/red-black-tree.tex
20555 @@ -0,0 +1,61 @@
20556 +% Red-black tree
20557 +% Author: Madit
20558 +\documentclass{article}
20559 +\usepackage{tikz}
20560 +%%%<
20561 +\usepackage{verbatim}
20562 +\usepackage[active,tightpage]{preview}
20563 +\PreviewEnvironment{tikzpicture}
20564 +\setlength{\PreviewBorder}{10pt}%
20565 +%%%>
20566 +\begin{comment}
20567 +:Title: Red-black tree
20568 +:Tags: Trees;Graphs
20569 +:Author: Madit
20570 +:Slug: red-black-tree
20571 +
20572 +:A red-black tree is a special type of binary tree, used in computer science
20573 +:to organize pieces of comparable data, such as text fragments or numbers.
20574 +(Wikipedia)
20575 +\end{comment}
20576 +\usetikzlibrary{arrows}
20577 +
20578 +\tikzset{
20579 +    treenode/.style = {align=center, inner sep=0pt, text centered,
20580 +        font=\sffamily},
20581 +    arn_n/.style = {treenode, circle, white, font=\sffamily\bfseries, draw=black,
20582 +        fill=black, text width=1.5em},% arbre rouge noir, noeud noir
20583 +    arn_r/.style = {treenode, circle, red, draw=red,
20584 +        text width=1.5em, very thick},% arbre rouge noir, noeud rouge
20585 +    arn_x/.style = {treenode, rectangle, draw=black,
20586 +        minimum width=0.5em, minimum height=0.5em}% arbre rouge noir, nil
20587 +}
20588 +
20589 +\begin{document}
20590 +\begin{tikzpicture}[->,>=stealth',level/.style={sibling distance = 5cm/#1,
20591 +    level distance = 1.5cm}]
20592 +\node [arn_n] {33}
20593 +    child{ node [arn_r] {15}
20594 +        child{ node [arn_n] {10}
20595 +            child{ node [arn_r] {5} edge from parent node[above left]
20596 +                {$x$}} %for a named pointer
20597 +            child{ node [arn_x] {} }
20598 +        }
20599 +        child{ node [arn_n] {20}
20600 +            child{ node [arn_r] {18}}
20601 +            child{ node [arn_x] {} }
20602 +        }
20603 +    }
20604 +    child{ node [arn_r] {47}
20605 +        child{ node [arn_n] {38}
20606 +            child{ node [arn_r] {36}}
20607 +            child{ node [arn_r] {39}}
20608 +        }
20609 +        child{ node [arn_n] {51}
20610 +            child{ node [arn_r] {49}}
20611 +            child{ node [arn_x] {} }
20612 +        }
20613 +    }
20614 +}
20615 +\end{tikzpicture}
20616 +\end{document}
20617 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
20618 index fa90c6d..5053a33 100644
20619 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
20620 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
20621 index fe33786..e8d8ae5 100644
20622 --- a/doc/tex/relatorio.tex
20623 +++ b/doc/tex/relatorio.tex
20624 @@ -36,11 +36,34 @@
20625     tabsize=2
20626 }
20627
20628 +% \tikzset{
20629 +%     treenode/.style = {align=center, inner sep=0pt, text centered,
20630 +%         font=\sffamily},
20631 +%     arn_n/.style = {treenode, circle, white, font=\sffamily\bfseries, draw=black,
20632 +%         fill=black, text width=1.5em},% arbre rouge noir, noeud noir
20633 +%     arn_r/.style = {treenode, circle, red, draw=red,
20634 +%         text width=1.5em, very thick},% arbre rouge noir, noeud rouge
20635 +%     arn_x/.style = {treenode, rectangle, draw=black,
20636 +%         minimum width=0.5em, minimum height=0.5em}% arbre rouge noir, nil
20637 +% }
20638 +

```

```

20639 +\tikzset{
20640 +   treenode/.style = {align=center, inner sep=0pt, text centered,
20641 +     font=\sffamily},
20642 +   arn_n/.style = {treenode, circle, white, font=\sffamily\bfseries, draw=black,
20643 +     fill=black, text width=1.5em},% arbre rouge noir, noeud noir
20644 +   arn_r/.style = {treenode, circle, white, font=\sffamily\bfseries, draw=red, fill=red,
20645 +     text width=1.5em},% arbre rouge noir, noeud rouge
20646 +   arn_x/.style = {treenode, rectangle, draw=black, fill=black,
20647 +     minimum width=0.5em, minimum height=0.5em}% arbre rouge noir, nil
20648 +}
20649 +
20650 +\lstset{
20651 +   style=mystyle,
20652 +   language=C,
20653 +   inputencoding=utf8,
20654 +   texcl=true
20655 +   language=C
20656 +   inputencoding=utf8,
20657 +   texcl=true,
20658 +   escapeinside={(!){!}}
20659 +}
20660
20661 % \renewcommand{\headrulewidth}{0pt} \renewcommand{\footrulewidth}{0pt}
20662 @@ -56,7 +79,7 @@
20663 % ]{DejaVu Sans}
20664
20665 \title {
20666 - \Large{\textbf{RELATÓRIO SOBRE O CÓDIGO FONTE DESTE PROJETO, CONJUNTO ORDENADO}} \\\
20667 + \Large{\textbf{RELATÓRIO SOBRE O CÓDIGO FONTE DO PROJETO, CONJUNTO ORDENADO}} \\\
20668 \large{\textbf{projeto este que faz uso da estrutura de dados Red Black}} \\\
20669 \large{UFES Centro Universitário Norte do Espírito Santo}
20670 \author{JOILNEN LEITE \\\ \href{mailto:joilnen.leite@edu.ufes.br}}
20671 @@ -90,16 +113,250 @@ da atividade conjunto ordenado \\\
20672 \indent \textbf{Palavras-chave: } fontes, C, Red Black, Conjunto\\\
20673 \small
20674 \section {Introdução}
20675 -\noindent Esta biblioteca é composta por quatro arquivos, \\\
20676 -\enf{conjunto\_ordenado.h}\\\
20677 -\enf{conjunto\_ordenado.c}\\\
20678 -\enf{red\_black.h}\\\
20679 -\enf{red\_black.c}\\\
20680 +\noindent Esta biblioteca é composta pelo os seguintes arquivos, \\\
20681 +\begin{itemize}
20682 + \item conjunto\_ordenado.h
20683 + \item conjunto\_ordenado.c
20684 + \item red\_black.h
20685 + \item red\_black.c
20686 + \item jcourses.h
20687 +\end{itemize}
20688 +
20689 +Completamente separado do código da biblioteca temos alguns arquivos que implementam
20690 +testes, estes são:
20691 +\begin{itemize}
20692 + \item main.c
20693 + \item testa_item_1.c
20694 + \item testa_item_2.c
20695 + \item testa_item_3.c
20696 + \item testa_rb.c
20697 +\end{itemize}
20698 +
20699 +Todos os arquivos estão listados nos anexos na sua íntegra.
20700 +
20701 +Como foi implementado um número grande de testes, estes foram separados em quatro
20702 +arquivos, \enf{testa_item_1.c, testa_item_2.c, testa_item_3.c, testa_rb.c}
20703 +e tem suas funções chamadas sequencialmente dentro da função \enf{main}, no arquivo \enf{main.c}
20704 +
20705 +Seguiremos neste relatório uma abordagem \enf{top-down} onde partiremos das estruturas
20706 +e funções manipuladas pelo o código cliente em direção as estruturas e funções que implementam
20707 +e operam na estrutura de dados, \enf{árvore red-black}, que é totalmente ocultada do cliente
20708 +ou seja poderíamos reimplementar as funcionalidade com outras esturas de dados e manter
20709 +a interface compatível com a existente.
20710 +
20711 +O estilo do código fonte neste trabalho é o mais tradicional, chaves abrem e fecham do mesmo lado
20712 +nas funções e instruções escritas em mais de uma linha,
20713 +entre cada instrução e seus operandos há sempre espaços, com exceção das funções e seus parênteses,
20714 +os eapços dividem visialmente os tokens, como em arrays, em equações, símbolo
20715 +de ponteiro alinhado à variável, todos os comentários seguem ANSI C, /* */.
20716 +
20717 +Na leitura da documentação nos comentários vale a pena ressaltar que todos estão em ASCII,
20718 +por isso não tem acentuação e a descrição dos parâmetros são antecedidas com \enf{@param}
20719 +que é tag utilizada pelo o sistema que gera documentação apartir do código fonte,
20720 +documentação esta constante nos anexos.
20721 +
20722 +\section {Primeiras Estruturas}
20723 +\subsection {conjord}
20724 +A primeira estrutura que veremos aqui é a \enf{conjord} ela representa um conjunto ordenando
20725 +é o nosso envólucro sobre a estrutura da árvore,
20726 +
20727 +\lstinputlisting [linrange={11-31}, firstnumber=11, caption = {Fragmento do conjunto_ordenado.h}]{../src/
conjunto_ordenado.h}
20728 +
20729 +Aqui temos a função que cria um conjunto ordenando, um tipo conjord
20730 +
20731 +\lstinputlisting [firstnumber=9, linrange={9-23}, caption = {Fragmento do conjunto_ordenado.c}]{../src/
conjunto_ordenado.c}
20732 +
20733 +Apesar do código ser bem simples, temos aqui detalhadamente, a criação do tipo cojord
20734 +, o ponteiro pr aponta para raiz da árvore, interna a este conjunto, como
20735 +o conjunto tá vazio ele aponta para o nó sentinela(rb_s - red black sentinela),
20736 +idx é um indice para um array de raízes que é usado apenas pra controle e acesso
20737 +rápido a todas as árvores criadas, nome é uma identificação opcional e cardinalidade
20738 +zero por que o conjunto está vazio.
20739 +
20740 +Aqui o exemplo do seu uso num caso de teste
20741 +
20742 +\lstinputlisting [firstnumber=54, linrange={54-56}, caption = {Fragmento do testa_item_1.
c}]{../src/testa_item_1.
c}
20743 +
20744 +Sigamos para a função de teste a seguir \enf{adiciona_lista_de_elementos} ela vai exemplificar
20745 +como essa criação se dá até a árvore
20746 +
20747 +\lstinputlisting [firstnumber=9, linrange={9-42}, caption = {Fragmento do testa_item_1.c}]{../src/testa_item_1.c}
20748 +
20749 +Nesse fragmento vemos o tipo definido pelo o usuário será o tipo genérico
20750 +onde apontaremos o ponteiro \enf{void *} pra ele,
20751 +apartir da linha 28, temos a criação de um tipo composto do tipo do cliente, aqui como

```

```

20752 +uma estrutura com dois ponteiros para char, sequencia finalizada como '\textbackslash0', que é como
20753 +representamos strings em C, são criados 20 elementos neste loop, com chave randômica
20754 +de 1 a 99, a função \en{conjord_inser_e elemento} na linha 33, tendo como parâmetros
20755 +o conjunto criado, a chave e o objeto do cliente, criará um novo elemento neste conjunto,
20756 +esta função retorna código de erro, zero para elemento inserido, e não zero caso,
20757 +o elemento já exista na árvore, então neste \en{if}, nos certificamos que apenas
20758 +teremos elementos com chaves diferentes nos 20 elementos ou nesta árvore, analizaremos
20759 +esta função a seguir,
20760 +
20761 +\lstinputlisting [firstnumber=39, linerange={39-65}, caption = {Fragmento do conjunto_ordenado.c}]{../src/
conjunto_ordenado.c}
20762 +o \en{if}, na linha 47, adiciona um elemento na árvore já existente, pois note que \en{pr} diferente de \en{rb_s},
o que significa que já existe árvore criada para este \en{conjord},
20763 +caso contrário é criada a raiz da árvore e apontado o \en{pr} para ela, \en{rb_cr},
20764 +cria uma árvore nova retorna o nó raiz tendo uma chave e um tipo qualquer(\en{void *}),
20765 +cardinalidade é incrementado, e no caso da inserção em árvore pre-existente,
20766 +ele só é incrementado caso a função \en{rb_insert}, retorne código de inserido, zero,
20767 +temos como retorno dessa função de inserção no conjunto, zero para inserido, e -1,
20768 +caso o contrário, é assim que nos certificamos de ter inserido um elemento realmente
20769 +como no código de teste visto anteriormente, note que usamos ponteiro duplo na raiz,
20770 +por que em algumas operações teremos que modificar esse ponteiro dentro de funções chamadas,
20771 +como em rotações e deleções.
20772 +
20773 +\subsection {Estruturas da Red Black}
20774 +A função \en{conjord_inser_e elemento} chama no seu interior \en{rb_insert}, que insere
20775 +o dado genérico apontado por \en{elemento} e sua \en{chave}, na árvore apontada por \en{pr}
20776 +do conjord \en{a}, o nó \en{rb_n *n} é alocado, populado com chave e elemento e passado
20777 +por referência para \en{rb_insert} caso esta retorne zero a inserção foi executada corretamente,
20778 +então incrementamos o contador de cardinalidade, caso contrário a função retorna código de erro -1.
20779 +
20780 +Analisaremos agora \en{rb_cr} e \en{rb_insert}, respectivamente as funções que cria uma árvore
20781 +red black e insere um novo nó.
20782 +\lstinputlisting [firstnumber=25, linerange={25-41}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
20783 +A mesma ideia que vimos na função \en{conjord_inser_e elemento}, alocamos aqui um novo nó,
20784 +populamos com chave e o dado genérico, também setamos a cor preta, pois é raiz, apontamos
20785 +ponteiros, pai, e filhos para nó sentinela \en{rb_s}
20786 +
20787 +Antes de passarmos para o \en{rb_insert}, vamos dar uma olha na estrutura nó \en{rb_n}
20788 +\lstinputlisting [firstnumber=33, linerange={33-50}, caption = {Fragmento do red_black.h}, a
documentação de comentário é explícita.
20789 +
20790 +Então agora a função \en{rb_insert} que insere um novo nó na árvore, detalharemos agora
20791 +o funcionamento desta função
20792 +\lstinputlisting [firstnumber=43, linerange={43-145}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
20793 +
20794 +A variável \en{x} é uma variável que itera sobre os nós da árvore ela começa sendo igual
20795 +ao nó raiz, \en{y} é uma variável que aponta para pai, pois a lógica dessa rotina é ir rastreando
20796 +nó e seu pai, como a condição inicial é o nó raiz, a \en{x} aponta para raiz e a \en{y} aponta
20797 +para o nó sentinela, consideremos os casos possíveis, caso um, o nó raiz não existe ou seja é
20798 +igual a sentinela e estamos incluindo o primeiro nó que se tornará o raiz, neste caso
20799 +o primeiro \en{while} não é executado pois a \en{x} apontará para \en{rb_s} (sentinela),
20800 +como \en{y} rastreia o pai, ela aponta também para sentinela, e fazemos o nó incluído apontar
20801 +o ponteiro pai para o sentinela através \en{z->p = y}, caso neste ponto a \en{y} continua
20802 +apontando para sentinela significa que temos apenas um nó sendo tratado, logo fazemos este nó
20803 +raiz na linha \en{*t = z}, posteriormente fazemos ponteiros pai e filhos apontarem para sentinela
20804 +e colorimos de vermelho, neste ponto você pode perguntar, mas se ele é o raiz não deveria ser
20805 +preto, verdade, é que essa linha é feita com vistas aos casos mais comuns que são a inserção
20806 +de nós não raiz, que é tratado no \en{while} seguinte, a cor será alterado para preto, no
20807 +final da função pelo o ponteiro \en{t} na linha \en{(*t)->c = BLACK}, ou seja ao final
20808 +da função ela sempre pinta o nó raiz de preto.
20809 +
20810 +Caso dois, neste o nó inserido não é o primeiro, o primeiro \en{while} itera pelo nó raiz,
20811 +através dos ponteiros para o(s) filho(s) a direita ou a esquerda de acordo com a comparação
20812 +das chaves, dos nós visitados e do nó sendo incluído, note que neste ponto, é feito
20813 +a crítica se as chaves são iguais, como a especificação proíbe chaves iguais, ele retorna
20814 +código de erro após o \en{else} ou finaliza quando a \en{x} apontar para o sentinela
20815 +que neste caso representa um nó folha, a \en{y} apontará para o pai deste nó, então
20816 +fora é o ponto que será incluído o novo nó, na linha \en{z->p = y}, enquadro o primeiro
20817 +\en{if} setou as variáveis \en{x} e \en{y}, o segundo \en{if}, inclui o novo nó
20818 +\en{z} no lado esquerdo ou direito de \en{y}, aponta os filhos desde nó para sentinela
20819 +e colore com vermelho, que é a ação mais comum.
20820 +
20821 +Todas as inclusões entram numa etapa, o segundo \en{while}, que chamei de \en{ajuste}
20822 +na bibliografia é referido como \en{fixup}, esta parte do código faz o ajuste de coloração
20823 +e rotaciona nós para manter as propriedades da árvore red-black obedecidas.
20824 +O primeiro \en{if} dentro deste segundo \en{while}.
20825 +
20826 +Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \en{z}
20827 +é o nó a esquerda, ou seja filho a esquerda do avô, fazemos \en{y} apontar para o nó
20828 +pai a direita. O diagrama abaixo representa esta
20829 +
20830 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
+ level distance = 1cm}]
20831 + \node [arn_n] {}
20832 + \node [arn_r] {}
20833 + \node [arn_r] {}
20834 + \node [arn_r] {}
20835 + \node [arn_x] {}
20836 + \node [arn_x] {}
20837 + \node [arn_x] {}
20838 + \node [arn_x] {}
20839 + \node [arn_r] {}
20840 + \node [arn_r] {}
20841 + \node [arn_x] {}
20842 + \node [arn_x] {}
20843 + \node [arn_x] {}
20844 + \node [arn_x] {}
20845 + \node [arn_x] {}
20846 + \node [arn_x] {}
20847 + \node [arn_x] {}
20848 + \node [arn_x] {}
20849 + \node [arn_x] {}
20850 + \node [arn_x] {}
20851 + \node [arn_x] {}
20852 + \node [arn_x] {}
20853 + \node [arn_x] {}
20854 + \node [arn_x] {}
20855 + \node [arn_x] {}
20856 + \node [arn_x] {}
20857 + \node [arn_x] {}
20858 + \node [arn_x] {}
20859 + \node [arn_x] {}
20860 + \node [arn_x] {}
20861 + \node [arn_x] {}
20862 + \node [arn_x] {}
20863 + \node [arn_x] {}
20864 + \node [arn_x] {}

```

```

20865 +%
20866 +%
20867 +\begin{tikzpicture}[<->,>=stealth',level/.style={sibling distance = 2.5cm/#1,
20868 + level distance = 1cm}]
20869 +\node [arn_r] {}
20870 + child{ node [arn_n] {}
20871 + child{ node [arn_r] {z}
20872 + child{ node [arn_x] {}}}
20873 + child{ node [arn_x] {}}}
20874 + }
20875 + child{ node [arn_x] {}}}
20876 + }
20877 + child{ node [arn_n] {}
20878 + child { node [arn_x] {} }
20879 + child { node [arn_x] {} }
20880 + }
20881 +;
20882 +\end{tikzpicture}
20883 +\colorbox{BurntOrange}{após o \en{else} nas linhas, 107-117 }
20884 +\begin{tikzpicture}[<->,>=stealth',level/.style={sibling distance = 2.5cm/#1,
20885 + level distance = 1cm}]
20886 +\node [arn_n] {}
20887 + child{ node [arn_n] {}
20888 + child{ node [arn_r] {z}
20889 + child{ node [arn_x] {}}}
20890 + child{ node [arn_x] {}}}
20891 + }
20892 + child{ node [arn_x] {}}}
20893 + }
20894 + child{ node [arn_n] {}
20895 + child { node [arn_x] {} }
20896 + child { node [arn_x] {} }
20897 + }
20898 +;
20899 +\end{tikzpicture}
20900 +\\
20901 +\\
20902 +\\
20903 +\\
20904 +\\
20905 +\\
20906 +%%
20907 +%%
20908 +%% daqui pra frente eh antigo
20909 +%%
20910 +%%
20911 +\en{conjunto_ordenado.h} é o arquivo cabeçalho que deverá ser incluído no código
20912 +fonte do cliente, \en{conjunto_ordenado.c} onde estão implementados as funções,
20913 +\en{red_black.h} e \en{red_black.c}, cabeçalho e implementações da estrutura
20914 +de dados, árvore red-black e \en{jurses.h} um arquivo com macros auxiliares
20915 +de impressao e formatação de saída na tela.
20916 +
20917 +sendo necessário apenas para seu uso a inclusão via, \\
20918 +0 TAD \en{conjord} é a abstração mais alta, representando conjunto, a qual o código
20919 +cliente tem acesso e suas respectivas funções.
20920 +\en{\#include "conjunto_ordenado.h"} onde se encontra declarado, sendo definido em \\
20921 +\en{"conjunto_ordenado.c"}, o TAD \footnote{Tipo Abstrato de Dados}
20922 +\en{conjord} que oculta do código cliente e manipula internamente a estrutura de dados
20923 +red black \en{"struct rb_n *"} aqui neste trabalho, declarada em \en{red_black.h} e definida em
20924 +red black \en{"struct rb_n"} aqui neste trabalho, declarada em \en{red_black.h} e definida em
20925 +\en{red_black.c}
20926 +
20927 +Além da convencional inclusão e compilação, o usuário da biblioteca, pessoa
20928 +@@ -109,7 +366,7 @@ que escreve o programa cliente, aqui neste trabalho exemplificado pelo os arquiv
20929 +deverá escolher definir seu tipo dado, através de uma struct ou utilizar um tipo de dado nativo da
20930 +linguagem.
20931 +\renewcommand{\lstlistingname}{Fonte}
20932 +\begin{lstlisting}[language=c, caption={Exemplo do uso da biblioteca conjunto_ordenado}]
20933 +\begin{lstlisting}[caption={Exemplo do uso da biblioteca conjunto_ordenado}]
20934 +\include "conjunto_ordenado.h"
20935 +
20936 +int main()
20937 +@@ -147,7 +404,7 @@ com \en{insere_elemento}
20938 + \ \
20939 +
20940 +\noindent Estrutura principal
20941 +\begin{lstlisting}[language=c, caption={Nó Red Black}]
20942 +\begin{lstlisting}[caption={Nó Red Black}]
20943 +/**
20944 + * composicao do noh da RB
20945 + * @rb_n noh red black
20946 +@@ -209,9 +466,10 @@ int seta_imprime_elemento(void (*i)(rb_n *n));
20947 +
20948 +\end{lstlisting}
20949 +
20950 +-\lstinputlisting [language=c] {code/inc.c}
20951 +-\lstinputlisting [language=c] {code/del.c}
20952 +-\lstinputlisting [language=c] {code/utills.c}
20953 +\lstinputlisting [linerange={10-15}, firstnumber=10]{.../src/testa_item_1.c}
20954 +Variavel rb_n com nome y \ref{14}
20955 +% \lstinputlisting {code/del.c}
20956 +% \lstinputlisting {code/utills.c}
20957 +
20958 +\section {TAD}
20959 +\begin{figure}[H]
20960 +diff --git a/src/checklist.c b/src/checklist.c
20961 +index 189eb62..399bf26 100644
20962 +--- a/src/checklist.c
20963 ++++ b/src/checklist.c
20964 +@@ -3,6 +3,9 @@
20965 + #include "conjunto_ordenado.h"
20966 + #include "checklist.h"
20967 +
20968 +/** mostra itens da avaliacao
20969 + * @param i especifica que itens estao selecionados
20970 + */
20971 +void tela_checklist(int i)
20972 +{
20973 + int l = 5, c = 24, bo = 1UL;
20974 +diff --git a/src/checklist.h b/src/checklist.h
20975 +index 730b616..aaf3a24 100644
20976 +--- a/src/checklist.h
20977 ++++ b/src/checklist.h
20978 +@@ -5,8 +5,12 @@
20979 + * aluno: joilnen leite
20980 + */

```

```

20981
20982 /** mostra itens da avaliacao
20983 * * @param i especifica que itens estao selecionados
20984 * */
20985 void tela_checklist(int i);
20986
20987 /** itens da avaliacao mostrados na tela */
20988 #define ITEM1_1 "1.1 listar ordenado por chave"
20989 #define ITEM1_2 "1.2 cardinalidade do conjunto"
20990 #define ITEM1_3 "1.3 verificar se conj. vazio"
20991 @@ -22,6 +26,7 @@ void tela_checklist(int i);
20992
20993 #define ITEM4 "4 imprimir arvore rubro negra"
20994
20995 /** usado na operacao de bits na mostra de itens da avaliacao */
20996 enum {I1_1, I1_2, I1_3, I1_4, I2_1, I2_2, I2_3, \
20997       I3_1, I3_2, I3_3, I4};
20998
20999 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
21000 index ef4d273..79e54c4 100644
21001 --- a/src/conjunto_ordenado.c
21002 +++ b/src/conjunto_ordenado.c
21003 @@ -8,6 +8,8 @@ static unsigned int r_counter = 0;
21004
21005 /**
21006  * cria conjunto ordenado e retorna o ponteiro para ele
21007  * * @param nome string que nomeia conjunto, caso sem nome,
21008  * * NULL
21009  */
21010 conjord *conjord_cria(char *nome)
21011 {
21012 @@ -36,6 +38,9 @@ void conjord_destroi(conjord *c)
21013
21014 /**
21015  * insere elemento no conjunto ordenado
21016  * * @param a conjunto ordenando
21017  * * @param chave
21018  * * @param ponteiro para tipo do cliente
21019  */
21020 int conjord_insere_elemento(conjord *a, int chave, void *elemento)
21021 {
21022 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
21023 index 3917c96..52406e9 100644
21024 --- a/src/conjunto_ordenado.h
21025 +++ b/src/conjunto_ordenado.h
21026 @@ -9,15 +9,21 @@
21027 #include "red_black.h"
21028
21029 /**
21030  * estrutura da TAD explicitamente
21031  * * estrutura do tipo abstrado de dados
21032  * * esta estrutura oculta o uso da red black
21033  * * suas funcoes e funcionamento interno do cliente
21034  * * expando apenas funcoes e operacoes com semantica
21035  * * relativa ao uso do cliente aqui nesta avaliacao
21036  * * conjunto ordenado as operacoes e funcoes
21037  * * exigidas no documento da avaliacao para este TAD
21038  */
21039 #ifndef conjord_h
21040 #define conjord_h
21041 * relativa ao uso do cliente
21042 * * @param pr ponteiro para para raiz da arvore embarcada
21043 * * @param idx indice para um array de ponteiros para as raizes,
21044 * * eh utilizado internamente para se ter acessor rapido a todas
21045 * * as raizes criadas
21046 * * (controle do numero de conjuntos)
21047 * * @param nome opcional para o conjunto
21048 * * @param cardinalidade do conjunto
21049 */
21050 typedef struct conjord
21051 {
21052     rb_n *pr;
21053     unsigned int idx;
21054     char *nome;
21055 } conjord;
21056 @@ -26,6 +32,8 @@ typedef struct conjord {
21057
21058 /**
21059  * cria conjunto ordenado e retorna o ponteiro para ele
21060  * * @param nome string que nomeia conjunto, caso sem nome,
21061  * * NULL
21062  */
21063 conjord *conjord_cria(char *nome);
21064
21065 diff --git a/src/red_black.c b/src/red_black.c
21066 index 8aee293..8386a31 100644
21067 --- a/src/red_black.c
21068 +++ b/src/red_black.c
21069 @@ -41,7 +41,9 @@ rb_n* rb_cr(int key, void *dado)
21070 }
21071
21072 /**
21073  * inclusao baseado no livro do cormen
21074  * * inclusao de um novo noh na arvore rb
21075  * * @param t eh a noh raiz
21076  * * @param z eh o noh a ser incluido
21077  */
21078 int rb_insert(rb_n **t, rb_n *z)
21079 {
21080 diff --git a/src/red_black.h b/src/red_black.h
21081 index a4a3331..11f2828 100644
21082 --- a/src/red_black.h
21083 +++ b/src/red_black.h
21084 @@ -36,7 +36,6 @@ typedef enum tipo_screen {TMONO, TCOLOR} tipo_screen;
21085
21086 * @param k chave (k de key)
21087 * @param c cor (c de color ou cor, 0-black, 1-red)
21088 * @param d dado (d de data ou dados)
21089 * @param np numero de nohs pretos ateh a folha
21090 * @param l left, ponteiro para filho a esquerda
21091 * @param r right, ponteiro para filho a direita
21092 * @param p parent, ponteiro para noh pai
21093 @@ -45,7 +44,6 @@ typedef struct rb_n {
21094     int k;
21095     unsigned char c:1;
21096     void *d;
21097     unsigned int np;
21098     struct rb_n *l, *r, *p;
21099 } rb_n;

```



```

21097 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
21098 index ac7e322..ba28c67 100644
21099 --- a/src/testa_item_1.c
21100 +++ b/src/testa_item_1.c
21101 @@ -6,6 +6,11 @@
21102
21103 static void imprime_meu_tipo(void *);
21104
21105 /**
21106  * tipo definido pelo usuario
21107  * @param nome string em c
21108  * @param telefone string em c
21109  */
21110 typedef struct {
21111     char *nome;
21112     char *telefone;
21113 @@ -17,10 +22,6 @@ static void adiciona_lista_de_elementos(conjord *c)
21114     srand(time(NULL));
21115     for (i = 0; i < 20;)
21116     {
21117         /**
21118         * definicao de um tipo exemplo pelo o usuario
21119         */
21120         /**
21121         * criacao do dado para o tipo do usuario
21122         */
21123 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
21124 index ba9966d..4eab32f 100644
21125 --- a/src/testa_item_3.c
21126 +++ b/src/testa_item_3.c
21127 @@ -63,7 +63,7 @@ void testa_item_3(const char *m)
21128     conjord *cj_b = conjord_cria("B");
21129     int imp, pos0 = 0, pos1 = 1, pos2 = 2;
21130     char elem0[] = "ABCDEF";
21131     char elem1[] = "GHIJKL";
21132     char elem1[] = "DEFGHI";
21133
21134     LTELA;
21135     adiciona_venn(cj_a, pos1, elem0);
21136
21137 commit d7fe00babe1044669953bb1bc15c63e7f90ffe77
21138 Author: Joilnen <joilnen@gmail.com>
21139 Date: Wed Jan 12 04:45:45 2022 -0300
21140
21141 deleta arquivos n utilizado
21142
21143 diff --git a/doc/tex/red-black-tree.pdf b/doc/tex/red-black-tree.pdf
21144 deleted file mode 100644
21145 index 9966f82..0000000
21146 Binary files a/doc/tex/red-black-tree.pdf and /dev/null differ
21147 diff --git a/doc/tex/red-black-tree.tex b/doc/tex/red-black-tree.tex
21148 deleted file mode 100644
21149 index cd8afac..0000000
21150 --- a/doc/tex/red-black-tree.tex
21151 +++ /dev/null
21152 @@ -1,61 +0,0 @@
21153 -% Red-black tree
21154 -% Author: Madit
21155 -\documentclass{article}
21156 -\usepackage{tikz}
21157 -%%<
21158 -\usepackage{verbatim}
21159 -\usepackage[active,tightpage]{preview}
21160 -\PreviewEnvironment{tikzpicture}
21161 -\setlength{\PreviewBorder}{10pt}%
21162 -%%>
21163 -\begin{comment}
21164 -:Title: Red-black tree
21165 -:Tags: Trees;Graphs
21166 -:Author: Madit
21167 -:Slug: red-black-tree
21168 -
21169 -A red-black tree is a special type of binary tree, used in computer science
21170 -to organize pieces of comparable data, such as text fragments or numbers.
21171 -(Wikipedia)
21172 -\end{comment}
21173 -\usetikzlibrary{arrows}
21174 -
21175 -\tikzset{
21176 -    treenode/.style = {align=center, inner sep=0pt, text centered,
21177 -        font=\sffamily},
21178 -    arn_n/.style = {treenode, circle, white, font=\sffamily\bfseries, draw=black,
21179 -        fill=black, text width=1.5em},% arbre rouge noir, noeud noir
21180 -    arn_r/.style = {treenode, circle, red, draw=red,
21181 -        text width=1.5em, very thick},% arbre rouge noir, noeud rouge
21182 -    arn_x/.style = {treenode, rectangle, draw=black,
21183 -        minimum width=0.5em, minimum height=0.5em}% arbre rouge noir, nil
21184 -}
21185
21186 -\begin{document}
21187 -\begin{tikzpicture}[>,>=stealth',level/.style={sibling distance = 5cm/#1,
21188 -    level distance = 1.5cm}]
21189 -\node [arn_n] {33}
21190 -    child{ node [arn_r] {15}
21191 -        child{ node [arn_n] {10}
21192 -            child{ node [arn_r] {5} edge from parent node[above left]
21193 -                {$x$}} %for a named pointer
21194 -            child{ node [arn_x] {} }
21195 -        }
21196 -        child{ node [arn_n] {20}
21197 -            child{ node [arn_r] {18}}
21198 -            child{ node [arn_x] {} }
21199 -        }
21200 -    }
21201 -    child{ node [arn_r] {47}
21202 -        child{ node [arn_n] {38}
21203 -            child{ node [arn_r] {36}}
21204 -            child{ node [arn_r] {39}}
21205 -        }
21206 -        child{ node [arn_n] {51}
21207 -            child{ node [arn_r] {49}}
21208 -            child{ node [arn_x] {} }
21209 -        }
21210 -    }
21211 -}
21212

```



```

21213 -;
21214 -\end{tikzpicture}
21215 -\end{document}
21216
21217 commit 9bc43552baa8951043b0c272fb2edae2403d8d50
21218 Author: Joilnen <joilnen@gmail.com>
21219 Date: Wed Jan 12 15:19:59 2022 -0300
21220
21221 atualizando documentacao
21222
21223 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
21224 index 5053a33..f744110 100644
21225 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
21226 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
21227 index e8d8ae5..cccf14a 100644
21228 --- a/doc/tex/relatorio.tex
21229 +++ b/doc/tex/relatorio.tex
21230 @@ -49,13 +49,13 @@
21231
21232 \tikzset{
21233   treenode/.style = {align=center, inner sep=0pt, text centered,
21234     font=\sffamily},
21235   arn_n/.style = {treenode, circle, white, font=\sffamily\bfseries, draw=black,
21236     fill=black, text width=1.5em},% arbre rouge noir, noeud noir
21237   arn_r/.style = {treenode, circle, white, font=\sffamily\bfseries, draw=red, fill=red,
21238     text width=1.5em},% arbre rouge noir, noeud rouge
21239   font=\tiny,
21240   arn_n/.style = {treenode, circle, white, draw=black,
21241     text centered, fill=black, minimum size=3mm},% arbre rouge noir, noeud noir
21242   arn_r/.style = {treenode, circle, white, draw=red, fill=red,
21243     text centered, minimum size=3mm},% arbre rouge noir, noeud rouge
21244   arn_x/.style = {treenode, rectangle, draw=black, fill=black,
21245     minimum width=0.5em, minimum height=0.5em}% arbre rouge noir, nil
21246   minimum width=0.25em, minimum height=0.25em}% arbre rouge noir, nil
21247 }
21248
21249 \lstset{
21250 @@ -195,7 +195,8 @@ teremos elementos com chaves diferentes nos 20 elementos ou nesta árvore, anali
21251 esta função a seguir,
21252
21253 \lstinputlisting [firstnumber=39, linerange={39-65}, caption = {Fragmento do conjunto_ordenado.c}]{../src/
21254 conjunto_ordenado.c}
21255 -o \en{if}, na linha 47, adiciona um elemento na árvore já existente, pois note que \en{pr} diferente de \en{rb_s},
21256 o que significa que já existe árvore criada para este \en{conjord},
21257 +o \en{if}, na linha 47, adiciona um elemento na árvore já existente, pois note que \en{pr}
21258 +diferente de \en{rb_s}, o que significa que já existe árvore criada para este \en{conjord},
21259 caso contrário é criada a raiz da árvore e apontado o \en{pr} para ela, \en{rb_cr},
21260 cria uma árvore nova retorna o nó raiz tendo uma chave e um tipo qualquer(\en{void *}),
21261 cadididade é incrementado, e no caso da inserção em árvore pre-existente,
21262 @@ -227,7 +228,10 @@ Então agora a função \en{rb_insert} que insere um novo nó na árvore, detal
21263 o funcionamento desta função
21264 \lstinputlisting [firstnumber=43, linerange={43-145}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
21265
21266 -A variável \en{x} é uma variável que itera sobre os nós da árvore ela começa sendo igual
21267 +Tendo apenas um único nó esta rotina apenas atualiza o ponteiro raiz para este nó,
21268 +na linha 76, faz o nó incluído apontar seu ponteiro pai para o sentinela, através de \en{y}
21269 +(inicializado com sentinela em 55) e atualiza o nó raiz em 86 colorindo em seguida com preto em 142.
21270 +
21271 +A primeira parte desta rotina, que inclui o primeiro \en{while} até o
21272 +primeiro \en{if}, apenas faz a inclusão do novo nó da mesma forma que na árvore binária de
21273 +busca comum, o while itera até a folha comparando as chaves, neste caso aqui, foi
21274 +modificado para aceitar apenas chaves diferentes, retornando código de erro caso contrário,
21275 +e uma vez que temos a \en{y} apontando para este nó folha, anexamos \en{z} a este nó
21276 +fazendo o apontamento do ponteiro filho neste nó folha pra \en{z}, a única grande
21277 +diferença é que no final desta parte colorimos com vermelho, o novo nó incluído.
21278 +
21279 +Todas as inclusões entram numa etapa, o segundo \en{while}, que chamei de \en{ajuste}
21280 +na bibliografia é referida como \en{fixup}, esta parte do código faz o ajuste de coloração
21281 +e rotaciona nós para manter as propriedades da árvore red-black obedecidas.
21282 +
21283 +Veamos como esta parte da rotina trata os três casos em que podem ser quebradas
21284 +as propriedades da árvore na inclusão e como esta parte do código resolve as inconsistências.
21285 +
21286 +Considerando a priori apenas as inclusões que ocorrerão a esquerda, nós anexados ao ponteiro
21287 +que aponta para o filho esquerdo.
21288 +
21289 +O primeiro caso é tratado no segnode \en{if} do segundo \en{while}, o primeiro \en{if}
21290 +decide se está sendo incluído pelo o lado esquerdo e da linha 99-106, temos o tratamento
21291 +do \en{caso 1}, nó pai vermelho, e nó tio vermelho, segue um diagrama deste caso em
21292 +nome da clereza.
21293 +
21294 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
21295 + level distance = .5cm}]
21296 +\node [arn_n] {}
21297 +  child{ node [arn_r] {}
21298 +    child{ node [arn_n] {}
21299 +      child { node [arn_x] {} }
21300 +      child { node [arn_x] {} }
21301 +    }
21302 +    child{ node [arn_n] {}
21303 +      child{ node [arn_r] {}
21304 +        child { node [arn_x] {} }
21305 +        child { node [arn_x] {} }
21306 +      }
21307 +      child { node [arn_x] {} }
21308 +    }
21309 +    child{ node [arn_r] {}
21310 +      child { node [arn_x] {} }
21311 +      child { node [arn_x] {} }
21312 +    }
21313 +  }
21314 +  child{ node [arn_n] {}
21315 +    child { node [arn_x] {} }
21316 +    child{ node [arn_r] {}
21317 +      child { node [arn_x] {} }
21318 +      child { node [arn_x] {} }
21319 +    }
21320 +  }
21321 +;
21322 +\end{tikzpicture}
21323 +\colorbox{BurntOrange}{\scriptsize após o \en{if} nas linhas, 99-106 }
21324 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
21325 + level distance = .5cm}]
21326 +\node [arn_n] {}

```

```

21327 + child{ node [arn_r] {}
21328 + child{ node [arn_n] {}
21329 + child { node [arn_x] {} }
21330 + child { node [arn_x] {} }
21331 + }
21332 + child{ node [arn_r] {z}
21333 + child{ node [arn_n] {}
21334 + child{ node [arn_r] {}
21335 + child { node [arn_x] {} }
21336 + child { node [arn_x] {} }
21337 + }
21338 + child { node [arn_x] {} }
21339 + }
21340 + child{ node [arn_n] {}
21341 + child { node [arn_x] {} }
21342 + child { node [arn_x] {} }
21343 + }
21344 + }
21345 + }
21346 + child{ node [arn_n] {y}
21347 + child { node [arn_x] {} }
21348 + child{ node [arn_r] {}
21349 + child { node [arn_x] {} }
21350 + child { node [arn_x] {} }
21351 + }
21352 + }
21353 +;
21354 +\end{tikzpicture} \\\
21355 +Note que saímos do \enf{caso 1} e caímos no \enf{caso 2}, nó pai do nó agora
21356 +apontado por \enf{z}, vermelho e nó tio preto \enf{y}, lembre-se que estamos dentro de um
21357 +\enf{while} em loop enquanto a cor do pai de \enf{z} for vermelha, este segundo caso
21358 +é tratado no \enf{if} 109-113, onde há uma rotação a esquerda, vamos aproveitar para
21359 +analisarmos o que faz esta função.
21360 +\letinputlisting [firstnumber=319, linerange={319-342}, caption = {Fragmento do red_black.c}]{.../src/red_black.c}
21361 +Esta função \enf{rb_rot_esq}, basicamente diminui a altura do lado direito e aumenta
21362 +do lado esquerdo reagrupando os nós, vejamos detalhadamente como.
21363 +Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
21364 +é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
21365 +tio a direita. O diagrama abaixo representa esta
21366 +Detalhando, variável \enf{x} é uma variável que itera sobre os nós da árvore ela começa sendo igual
21367 +ao nó raiz, \enf{y} é uma variável que aponta para pai, pois a lógica dessa rotina é ir rastrando
21368 +nó e seu pai, como a condição inicial é o nó raiz, a \enf{x} aponta para raiz e a \enf{y} aponta
21369 +para o nó sentinela, consideremos os casos possíveis, caso um, o nó raiz não existe ou seja é
21370 @@ -254,7 +356,7 @@ fora é o ponto que será incluído o novo nó, na linha \enf{z->p = y}, enquad
21371 \enf{z} no lado esquerdo ou direito de \enf{y}, aponta os filhos desde nó para sentinela
21372 e colore com vermelho, que é a ação mais comum.
21373
21374 -Todas as inclusões entram numa etapa, o segundo \enf{while}, que chamei de \enf{ajuste}
21375 +Todas as inclusões entram numa etapa, o segundo \enf{while}, que chamei de \enf{ajuste}
21376 na bibliografia é referiad como \enf{fixup}, esta parte do código faz o ajuste de coloração
21377 e rotaciona nós para manter as propriedades da árvore red-black obedecidas.
21378 O primeiro \enf{if} dentro deste segundo \enf{while}.
21379 @@ 263,40 +365,6 @@ Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \
21380 é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
21381 tio a direita. O diagrama abaixo representa esta
21382
21383 -\begin{tikzpicture}[>=>stealth',level/.style={sibling distance = 2.5cm/#1,
21384 - level distance = 1cm}]
21385 -\node [arn_n] {}
21386 - child{ node [arn_r] {}
21387 - child{ node [arn_r] {z}
21388 - child{ node [arn_x] {} }
21389 - child{ node [arn_x] {} }
21390 - }
21391 - child{ node [arn_x] {} }
21392 - }
21393 - child{ node [arn_r] {}
21394 - child { node [arn_x] {} }
21395 - child { node [arn_x] {} }
21396 - }
21397 -;
21398 -\end{tikzpicture}
21399 -\colorbox{BurntOrange}{após o \enf{if} nas linhas, 100-106 }
21400 -\begin{tikzpicture}[>=>stealth',level/.style={sibling distance = 2.5cm/#1,
21401 - level distance = 1cm}]
21402 -\node [arn_n] {}
21403 - child{ node [arn_n] {}
21404 - child{ node [arn_r] {z}
21405 - child{ node [arn_x] {} }
21406 - child{ node [arn_x] {} }
21407 - }
21408 - child{ node [arn_x] {} }
21409 - }
21410 - child{ node [arn_n] {}
21411 - child { node [arn_x] {} }
21412 - child { node [arn_x] {} }
21413 - }
21414 -;
21415 -\end{tikzpicture}
21416 -\\
21417 %
21418 %
21419 %
21420 diff --git a/src/red_black.c b/src/red_black.c
21421 index 8386a31..981d8a0 100644
21422 --- a/src/red_black.c
21423 +++ b/src/red_black.c
21424 @@ -316,6 +316,12 @@ int rb_destroi(rb_n **a)
21425     return 0;
21426 }
21427
21428 /**
21429  * rotaciona a esquerda trocando as referências
21430  * entre nós
21431  * @param t nó raiz (ou da subarvore)
21432  * @param x nó que serah rotacionado
21433  */
21434 int rb_rot_esq(rb_n **t, rb_n *x)
21435 {
21436     rb_n *y = x->r;
21437
21438 commit 2cabd6ac2adad32d7174b1f3fb69fafc456ead66
21439 Author: Joilnen <joilnen@gmail.com>
21440 Date: Thu Jan 13 06:15:49 2022 -0300
21441
21442 documentacao

```

```

21443 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
21444 index f744110..fbede5f 100644
21445 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
21446 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
21447 index cccf14a..ab2d3b6 100644
21448 --- a/doc/tex/relatorio.tex
21449 +++ b/doc/tex/relatorio.tex
21450 @@ -9,6 +9,9 @@
21451 % \fancyfoot{}
21452 \usepackage{listingsutf8}
21453 \usepackage{xcolor}
21454 \usepackage{lmodern}
21455 +\usepackage[T1]{fontenc}
21456 +\usepackage{ae}
21457 +\usepackage{aer}
21458 % \definecolor{codegreen}{rgb}{0,0.6,0}
21459 \definecolor{codegreen}{rgb}{.6,.3,0}
21460 @@ -16,6 +19,8 @@
21461 \definecolor{codepurple}{rgb}{0.58,0,0.82}
21462 \definecolor{backcolour}{rgb}{0.95,0.95,0.92}
21463 +\DeclareEmphSequence{\bfseries, \mdseries}
21464 +
21465 \lstdefinestyle{mystyle}{
21466     backgroundcolor=\color{backcolour},
21467     commentstyle=\color{codegreen},
21468     @@ -175,7 +180,7 @@ zero por que o conjunto está vazio.
21469
21470 Aqui o exemplo do seu uso num caso de teste
21471
21472 -\lstinputlisting [firstnumber=54, linerange={54-56}, caption = {Fragmento do testa_item_1.c}]{../src/testa_item_1.
21473 c}
21474 +\lstinputlisting [firstnumber=55, linerange={55-56}, caption = {Fragmento do testa_item_1.c}]{../src/testa_item_1.
21475 c}
21476
21477 Sigamos para a função de teste a seguir \en{adiciona_lista_de_elementos} ela vai exemplificar
21478 como essa criação se dá até a árvore
21479 @@ -250,7 +255,7 @@ as propriedades da árvore na inclusão e como esta parte do código resolve as
21480 Considerando a priori apenas as inclusões que ocorrerão a esquerda, nós anexados ao ponteiro
21481 que aponta para o filho esquerdo.
21482
21483 -0 primeiro caso é tratado no segnode \en{if} do segundo \en{while}, o primeiro \en{if}
21484 +0 primeiro caso é tratado no segundo \en{if} do segundo \en{while}, o primeiro \en{if}
21485 decide se está sendo incluído pelo o lado esquerdo e da linha 99-106, temos o tratamento
21486 do \en{caso 1}, não pai vermelho, e não tio vermelho, segue um diagrama deste caso em
21487 nome da clareza.
21488 @@ -325,7 +330,86 @@ apontado por \en{z}, vermelho e não tio preto \en{y}, lembre-se que estamos de
21489 analisarmos o que faz esta função.
21490 \lstinputlisting [firstnumber=319, linerange={319-342}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
21491 Esta função \en{rb_rot_esq}, basicamente diminui a altura do lado direito e aumenta
21492 do lado esquerdo reagrupando os nós, vejamos detalhadamente como.
21493 +do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
21494 +\begin{table}[!ht]
21495 +\scriptsize
21496 +\begin{tabular}{|l|l|}
21497 +\hline
21498 +327 & guarda referência do filho a direita \\
21499 +\hline
21500 +328 & faz filho a direita igual a neto a esquerda (diminuindo a altura direita) \\
21501 +\hline
21502 +329 & se filho a direita (antigo neto a esquerda) não é sentinela \\
21503 +\hline
21504 +330 & atualiza pai de filho a direita (antigo neto a esquerda)\\
21505 +\hline
21506 +331 & atualiza pai do nó movido\\
21507 +\hline
21508 +332 & testa se nó anterior era raiz \\
21509 +\hline
21510 +333 & atualiza nó raiz \\
21511 +\hline
21512 +334 & se não é raiz testa se é nó a esquerda\\
21513 +\hline
21514 +335 & atualiza como nó a esquerda\\
21515 +\hline
21516 +336 & se nenhuma das opções anteriores\\
21517 +\hline
21518 +337 & atualiza como nó a direita\\
21519 +\hline
21520 +338 & atualiza filho a esquerda de y igual x, note que o nó x que era pai passa a ser filho\\
21521 +\hline
21522 +339 & atualiza pai de x igual de y \\
21523 +\hline
21524 +341 & retorna inteiro para uso em testes (não utilizado neste trabalho, mas a função tem esta interface)\\
21525 +\hline
21526 +\end{tabular}
21527 +\end{table}
21528 +
21529 +Um diagrama em nome da clareza \\
21530 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
21531 + level distance = .5cm}]
21532 +\node [arn_n] {x}
21533 +\child { node [arn_r] {1}
21534 +\child { node [arn_x] {} }
21535 +\child { node [arn_x] {} }
21536 +}
21537 +\child { node [arn_n] {y}
21538 +\child { node [arn_r] {5}
21539 +\child { node [arn_x] {} }
21540 +\child { node [arn_x] {} }
21541 +}
21542 +\child { node [arn_r] {8}
21543 +\child { node [arn_x] {} }
21544 +\child { node [arn_x] {} }
21545 +}
21546 +}
21547 +;
21548 +\end{tikzpicture}
21549 +\colorbox{BurntOrange}{\scriptsize após \en{rb_rot_esq} }
21550 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
21551 + level distance = .5cm}]
21552 +\node [arn_n] {y}
21553 +\child { node [arn_n] {x}
21554 +\child { node [arn_r] {1}
21555 +\child { node [arn_x] {} }
21556 +\child { node [arn_x] {} }

```

```

21557 +         }
21558 +         child{ node [arn_r] {5}
21559 +             child { node [arn_x] {} } }
21560 +             child { node [arn_x] {} } }
21561 +         }
21562 +     }
21563 +     child{ node [arn_r] {8}
21564 +         child { node [arn_x] {} } }
21565 +         child { node [arn_x] {} } }
21566 +     }
21567 +;
21568 +\end{tikzpicture} \\
21569 +\\
21570 +\\
21571 +\\
21572 +\\
21573 Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
21574 é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
21575 tio a direita. O diagrama abaixo representa esta
21576
21577 commit d0bac1f2bb0af97adbfff86a1dc98de537db953e
21578 Author: Joilnen <joilnen@gmail.com>
21579 Date: Thu Jan 13 06:22:40 2022 -0300
21580
21581 documentacao
21582
21583 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
21584 index fbde5f..1b84da1 100644
21585 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
21586 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
21587 index ab2d3b6..c1002df 100644
21588 --- a/doc/tex/relatorio.tex
21589 +++ b/doc/tex/relatorio.tex
21590 @@ -370,7 +370,7 @@ Um diagrama em nome da clareza \\
21591 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
21592 level distance = .5cm}]
21593 \node [arn_n] {x}
21594 - child{ node [arn_r] {1}
21595 + child{ node [arn_n] {1}
21596     child { node [arn_x] {} } }
21597     child { node [arn_x] {} } }
21598 }
21599 @@ -400,7 +400,7 @@ Um diagrama em nome da clareza \\
21600     child { node [arn_x] {} } }
21601 }
21602 - child{ node [arn_r] {8}
21603 + child{ node [arn_n] {8}
21604     child { node [arn_x] {} } }
21605     child { node [arn_x] {} } }
21606 }
21607 }
21608
21609 commit ee355fee21e11c6de623bcb7a546e8e21f69c944
21610 Author: Joilnen <joilnen@gmail.com>
21611 Date: Thu Jan 13 06:50:00 2022 -0300
21612
21613 documentacao
21614
21615 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
21616 index 1b84da1..9326675 100644
21617 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
21618 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
21619 index c1002df..6aeef3f 100644
21620 --- a/doc/tex/relatorio.tex
21621 +++ b/doc/tex/relatorio.tex
21622 @@ -227,7 +227,7 @@ populamos com chave e o dado genérico, também setamos a cor preta, pois é ra
21623 ponteiros, pai, e filhos para nó sentinela \enf{rb_s}
21624
21625 Antes de passarmos para o \enf{rb_insert}, vamos dar uma olha na estrutura nó \enf{rb_n}
21626 -\lstinputlisting [firstnumber=33, linerange={33-50}, caption = {Fragmento do red_black.h}, a
21627 documentação de comentário é explícita.
21628 +\lstinputlisting [firstnumber=33, linerange={33-48}, caption = {Fragmento do red_black.h}, a
21629 documentação de comentário é explícita.
21630
21631 Então agora a função \enf{rb_insert} que insere um novo nó na árvore, detalharemos agora
21632 o funcionamento desta função
21633 @@ -366,7 +366,7 @@ do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
21634 \end{table}
21635
21636 -Um diagrama em nome da clareza \\
21637 +\noindent Um diagrama em nome da clareza \\
21638 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
21639 level distance = .5cm}]
21640 \node [arn_n] {x}
21641
21642 commit 0f7051f5cb4ecd4c43fc83cd710fb71644cadbe1
21643 Author: Joilnen <joilnen@gmail.com>
21644 Date: Thu Jan 13 17:53:26 2022 -0300
21645
21646 formatando melhor saída no caso de teste 3
21647
21648 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
21649 index 4eab32f..088ba21 100644
21650 --- a/src/testa_item_3.c
21651 +++ b/src/testa_item_3.c
21652 @@ -68,28 +68,37 @@ void testa_item_3(const char *m)
21653     LTELA;
21654     adiciona_venn(cj_a, pos1, elem0);
21655     adiciona_venn(cj_b, pos2, elem1);
21656 +     printf("\033[8;30H");
21657 +     conjord_imprime_venn(cj_a);
21658 +     printf("\n");
21659 +     printf("\033[9;30H");
21660 +     conjord_imprime_venn(cj_b);
21661 +     printf("\n");
21662 +     printf("\033[10;30H");
21663 +     conjord_imprime_venn(conjord_une(cj_a, cj_b));
21664 +     INFO(" * item 3.1 uniao de conjuntos");
21665 +     CM;
21666
21667     LTELA;
21668 +     printf("\033[8;30H");
21669 +     conjord_imprime_venn(cj_a);
21670 +     printf("\n");
21671 +     printf("\033[9;30H");

```

```

21671     conjord_imprime_venn(cj_b);
21672     printf("\n");
21673 +   printf("\033[10;30H");
21674     conjord_imprime_venn(conjord_subtrai(cj_a, cj_b));
21675     INFO("* item 3.1 subtracao de conjuntos");
21676     CM;
21677
21678     LTEL;
21679 +   printf("\033[8;30H");
21680     conjord_imprime_venn(cj_a);
21681     printf("\n");
21682 +   printf("\033[9;30H");
21683     conjord_imprime_venn(cj_b);
21684     printf("\n");
21685 +   printf("\033[10;30H");
21686     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b));
21687     INFO("* item 3.1 interseccao com conjuntos");
21688     CM;
21689
21690 commit 749107752ff4d5e566a0a69c9cf559647bd81cd8
21691 Author: Joilnen <joilnen@gmail.com>
21692 Date:   Fri Jan 14 01:05:47 2022 -0300
21693
21694     melhorando a documentacao
21695
21696 diff --git a/doc/tex/packages.tex b/doc/tex/packages.tex
21697 index 5e4756a..54ac1c3 100644
21698 --- a/doc/tex/packages.tex
21699 +++ b/doc/tex/packages.tex
21700 @@ -1,13 +1,15 @@
21701 \usepackage{syntax}
21702 \usepackage[brazilian]{babel}
21703 \usepackage[utf8]{inputenc}
21704 +\usepackage[light,condensed,math]{iwona}
21705 +\usepackage[T1]{fontenc}
21706 % \usepackage[usenames, dvipsnames]{xcolor}
21707 \usepackage[usenames, dvipsnames]{color}
21708 % \usepackage{tgtermes} times roman font similar
21709 \usepackage{hyperref}
21710 \usepackage[indentfirst}
21711 \usepackage{bbding}
21712 -\usepackage{pifont}
21713 +% \usepackage{pifont}
21714 \usepackage{makeidx}
21715 \makeindex
21716 \hypersetup{
21717 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
21718 index 9326675..46ee4e7 100644
21719 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
21720 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
21721 index 6e6ee3f..c94b32c 100644
21722 --- a/doc/tex/relatorio.tex
21723 +++ b/doc/tex/relatorio.tex
21724 @@ -9,9 +9,6 @@
21725 % \fancyfoot{}
21726 \usepackage{listingsutf8}
21727 \usepackage{xcolor}
21728 -\usepackage{lmodern}
21729 -\usepackage[T1]{fontenc}
21730 -\usepackage{aesupp}
21731
21732 % \definecolor{codegreen}{rgb}{0,0.6,0}
21733 \definecolor{codegreen}{rgb}{.6,.3,0}
21734 @@ -20,6 +17,7 @@
21735 \definecolor{backcolour}{rgb}{0.95,0.95,0.92}
21736
21737 \DeclareEmphSequence{\bfseries, \mdseries}
21738 +% \renewcommand{\baselinestretch}{0.5}
21739
21740 \lstdefinestyle{mystyle}{
21741     backgroundcolor=\color{backcolour},
21742 @@ -38,7 +36,7 @@
21743     showstringspaces=false,
21744     showtabs=false,
21745     frame=single,
21746     -   tabsize=2,
21747     +   tabsize=2,
21748 }
21749
21750 % \tikzset{
21751 @@ -78,16 +76,18 @@
21752 \graphicspath{{fig/}}
21753
21754 \begin{document}
21755 +\normalfont
21756 % \setmainfont[
21757 %     Ligatures=TeX,
21758 %     Numbers=OldStyle, Proportional}
21759 % ]{DejaVu Sans}
21760
21761 \title {
21762 -   \Large{\textbf{RELATÓRIO SOBRE O CÓDIGO FONTE DO PROJETO, CONJUNTO ORDENADO}} \\\
21763 -   \Large{\textbf{projeto este que faz uso da estrutura de dados Red Black}} \\\
21764 -   \Large{\textbf{Relatório sobre o Código Fonte}} \\\
21765 +   \Large{\textbf{do Projeto, Conjunto Ordenado}} \\\
21766 +   \Large{\textbf{Uso da estrutura de dados, Árvore Red Black}} \\\
21767   \large {UFES Centro Universitário Norte do Espírito Santo}
21768 -   \author{JOILNEN LEITE \\\ \href{mailto:joilnen.leite@edu.ufes.br}}
21769 +   \author{Joilnen Leite \\\ \href{mailto:joilnen.leite@edu.ufes.br}}
21770   {\color{blue}joilnen.leite@edu.ufes.br}}
21771   % \footnotesize{Joilnen Leite}
21772   % \footnotesize{UFES Centro Universitário Norte do Espírito Santo}
21773 @@ -120,21 +120,21 @@ da atividade conjunto ordenado \\\
21774 \section {Introdução}
21775 \noindent Esta biblioteca é composta pelo os seguintes arquivos, \\\
21776 \begin{itemize}
21777 -   \item conjunto\_ordenado.h
21778 -   \item conjunto\_ordenado.c
21779 -   \item red\_black.h
21780 -   \item red\_black.c
21781 -   \item jcurses.h
21782 +   \item [\blacksquare$] conjunto\_ordenado.h
21783 +   \item [\blacksquare$] conjunto\_ordenado.c
21784 +   \item [\blacksquare$] red\_black.h
21785 +   \item [\blacksquare$] red\_black.c
21786 +   \item [\blacksquare$] jcurses.h

```

```

21787 \end{itemize}
21788
21789 Completamente separado do código da biblioteca temos alguns arquivos que implementam
21790 testes, estes são:
21791 \begin{itemize}
21792 - \item main.c
21793 - \item testa_item_1.c
21794 - \item testa_item_2.c
21795 - \item testa_item_3.c
21796 - \item testa_rb.c
21797 + \item [\blacksquare$] main.c
21798 + \item [\blacksquare$] testa_item_1.c
21799 + \item [\blacksquare$] testa_item_2.c
21800 + \item [\blacksquare$] testa_item_3.c
21801 + \item [\blacksquare$] testa_rb.c
21802 \end{itemize}
21803
21804 Todos os arquivos estão listados nos anexos na sua íntegra.
21805 @@ -167,7 +167,7 @@ A primeira estrutura que veremos aqui é a \en{conjord} ela representa um conju
21806
21807 \lstinputlisting [linarange={11-31}, firstnumber=11, caption = {Fragmento do conjunto_ordenado.h}]{../../src/
21808 conjunto_ordenado.h}
21809
21809 -Aqui temos a função que cria um conjunto ordenando, um tipo conjord
21810 +\noindent Aqui temos a função que cria um conjunto ordenando, um tipo conjord
21811
21812 \lstinputlisting [firstnumber=9, linerange={9-23}, caption = {Fragmento do conjunto_ordenado.c}]{../../src/
21813 conjunto_ordenado.c}
21814
21814 @@ -178,11 +178,11 @@ idx é um índice para um array de raízes que é usado apenas pra controle e aces
21815 rápido a todas as árvores criadas, nome é uma identificação opcional e cardinalidade
21816 zero por que o conjunto está vazio.
21817
21818 -Aqui o exemplo do seu uso num caso de teste
21819 +\noindent Aqui o exemplo do seu uso num caso de teste
21820
21821 \lstinputlisting [firstnumber=55, linerange={55-56}, caption = {Fragmento do testa_item_1.c}]{../../src/testa_item_1.
21822 c}
21823
21823 -Sigamos para a função de teste a seguir \en{adiciona_lista_de_elementos} ela vai exemplificar
21824 +\noindent Sigamos para a função de teste a seguir \en{adiciona_lista_de_elementos} ela vai exemplificar
21825 como essa criação se dá até a árvore
21826
21827 \lstinputlisting [firstnumber=9, linerange={9-42}, caption = {Fragmento do testa_item_1.c}]{../../src/testa_item_1.c}
21828 @@ -366,7 +366,66 @@ do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
21829 \end{tabular}
21830 \end{table}
21831
21832 -\noindent Um diagrama em nome da clareza \\
21833 +\noindent Um diagrama em nome da clareza, clro que aqui está representado apenas os nós
21834 +que são representativos de uma rotação a esquerda e não árvores completas.\\
21835 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
21836 + level distance = .5cm}]
21837 + \node [arn_n] {x}
21838 + \node [arn_n] {x}
21839 + \node [arn_n] {x}
21840 + \node [arn_n] {x}
21841 + \node [arn_n] {x}
21842 + \node [arn_n] {x}
21843 + \node [arn_n] {x}
21844 + \node [arn_n] {x}
21845 + \node [arn_n] {x}
21846 + \node [arn_n] {x}
21847 + \node [arn_n] {x}
21848 + \node [arn_n] {x}
21849 + \node [arn_n] {x}
21850 + \node [arn_n] {x}
21851 + \node [arn_n] {x}
21852 + \node [arn_n] {x}
21853 + \node [arn_n] {x}
21854 + \node [arn_n] {x}
21855 + \node [arn_n] {x}
21856 + \node [arn_n] {x}
21857 + \node [arn_n] {x}
21858 + \node [arn_n] {x}
21859 + \node [arn_n] {x}
21860 + \node [arn_n] {x}
21861 + \node [arn_n] {x}
21862 + \node [arn_n] {x}
21863 + \node [arn_n] {x}
21864 + \node [arn_n] {x}
21865 + \node [arn_n] {x}
21866 + \node [arn_n] {x}
21867 + \node [arn_n] {x}
21868 + \node [arn_n] {x}
21869 + \node [arn_n] {x}
21870 + \node [arn_n] {x}
21871 + \node [arn_n] {x}
21872 + \node [arn_n] {x}
21873 + \node [arn_n] {x}
21874 + \node [arn_n] {x}
21875 + \node [arn_n] {x}
21876 + \node [arn_n] {x}
21877 + \node [arn_n] {x}
21878 + \node [arn_n] {x}
21879 + \node [arn_n] {x}
21880 + \node [arn_n] {x}
21881 + \node [arn_n] {x}
21882 + \node [arn_n] {x}
21883 + \node [arn_n] {x}
21884 + \node [arn_n] {x}
21885 + \node [arn_n] {x}
21886 + \node [arn_n] {x}
21887 + \node [arn_n] {x}
21888 + \node [arn_n] {x}
21889 + \node [arn_n] {x}
21890 + \node [arn_n] {x}
21891 + \node [arn_n] {x}
21892 + \node [arn_n] {x}
21893 + \node [arn_n] {x}
21894 + \node [arn_n] {x}
21895 + \node [arn_n] {x}
21896 + \node [arn_n] {x}
21897 + \node [arn_n] {x}
21898 + \node [arn_n] {x}
21899 + \node [arn_n] {x}

```

```

21900 -\\
21901 -\\
21902 -\\
21903 -\\
21904 Aqui temos a parte mais importante da inclusão, se o nó pai do não incluído \enf{z}
21905 é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
21906 tio a direita. O diagrama abaixo representa esta
21907
21908 commit dd5fcbf3498c6e0ea845cb10d729ff937993a35a
21909 Author: Joilnen <joilnen@gmail.com>
21910 Date: Fri Jan 14 01:15:44 2022 -0300
21911
21912 atualizacao de repertorio
21913
21914 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
21915 index 46ee4e7..620617c 100644
21916 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
21917 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
21918 index c84b32c..1ee70a9 100644
21919 --- a/doc/tex/relatorio.tex
21920 +++ b/doc/tex/relatorio.tex
21921 @@ -174,7 +174,7 @@ A primeira estrutura que veremos aqui é a \enf{conjord} ela representa um conju
21922 Apesar do código ser bem simples, temos aqui detalhadamente, a criação do tipo conjord
21923 , o ponteiro pr aponta para raiz da árvore, interna a este conjunto, como
21924 o conjunto tá vazio ele aponta para o nó sentinela(rb_s - red black sentinela),
21925 -idx é um índice para um array de raízes que é usado apenas pra controle e acesso
21926 +idx é um índice para um array de raízes que é usado apenas para controle e acesso
21927 rápido a todas as árvores criadas, nome é uma identificação opcional e cardinalidade
21928 zero por que o conjunto está vazio.
21929
21930 @@ -188,7 +188,7 @@ como essa criação se dá até a árvore
21931 \lstinputlisting [firstnumber=9, linerange={9-42}, caption = {Fragmento do testa_item_1.c}]{.../src/testa_item_1.c}
21932
21933 Nesse fragmento vemos o tipo definido pelo o usuário será o tipo genérico
21934 -onde apontaremos o ponteiro \enf{void *} pra ele,
21935 +onde apontaremos o ponteiro \enf{void *} para ele,
21936 apartir da linha 28, temos a criação de um tipo composto do tipo do cliente, aqui como
21937 uma estrutura com dois ponteiros para char, sequência finalizada como '\textbackslash0', que é como
21938 representamos strings em C, são criados 20 elementos neste loop, com chave randômica
21939 @@ -242,7 +242,7 @@ primeiro \enf{if}, apenas faz a inclusão do novo nó da mesma forma que na árv
21940 busca comum, o while itera até a folha comparando as chaves, neste caso aqui, foi
21941 modificado para aceitar apenas chaves diferentes, retornando código de erro caso contrário,
21942 e uma vez que temos a \enf{y} apontando para este nó folha, anexamos \enf{z} a este nó
21943 -fazendo o apontamento do ponteiro filho neste nó folha pra \enf{z}, a única grande
21944 +fazendo o apontamento do ponteiro filho neste nó folha para \enf{z}, a única grande
21945 diferença é que no final desta parte colorimos com vermelho, o novo nó incluído.
21946
21947 Todas as inclusões entram numa etapa, o segndo \enf{while}, que chamei de \enf{ajuste}
21948
21949 commit 5901d2b83393fb16607bbbdcae3f9a8c24836245
21950 Author: Joilnen <joilnen@gmail.com>
21951 Date: Fri Jan 14 13:10:25 2022 -0300
21952
21953 documentacao
21954
21955 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
21956 index 620617c..5d71172 100644
21957 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
21958 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
21959 index 1ee70a9..59a8375 100644
21960 --- a/doc/tex/relatorio.tex
21961 +++ b/doc/tex/relatorio.tex
21962 @@ -9,6 +9,25 @@
21963 % \fancyfoot{}
21964 \usepackage{listingsutf8}
21965 \usepackage{xcolor}
21966 +\usepackage{booktabs}
21967 +\usepackage[most]{tcolorbox}
21968 +
21969 +\newtcolorbox[blend into=tables]{mytable}[2][\%
21970 + enhanced,
21971 + float,
21972 + every float=\centering,
21973 + capture=hbox,
21974 + title = #2,
21975 + attach boxed title to top left={%
21976 + xshift=5mm,
21977 + yshift=-\tcboxedtitleheight/2,
21978 + yshifttext=-1mm},
21979 + boxed title style={colback=blue!50, sharp corners},
21980 + colframe = blue!50,
21981 + colback = blue!20,
21982 + % overlay = {\node[text=white, fill=red] at (frame.east)
21983 + % {\clubsuit$};},
21984 + #1}
21985
21986 % \definecolor{codegreen}{rgb}{0,0.6,0}
21987 \definecolor{codegreen}{rgb}{.6,.3,0}
21988 @@ -53,9 +72,9 @@
21989 \tikzset{
21990 treenode/.style = {align=center, inner sep=0pt, text centered,
21991 font=\tiny},
21992 - arn_n/.style = {treenode, circle, white, draw=black,
21993 + arn_n/.style = {treenode, circle, yellow, draw=black, font=\scriptsize,
21994 text centered, fill=black, minimum size=3mm},% arbre rouge noir, noeud noir
21995 - arn_r/.style = {treenode, circle, white, draw=red, fill=red,
21996 + arn_r/.style = {treenode, circle, black, draw=red, fill=red, font=\scriptsize,
21997 text centered, minimum size=3mm},% arbre rouge noir, noeud rouge
21998 arn_x/.style = {treenode, rectangle, draw=black, fill=black,
21999 minimum width=0.25em, minimum height=0.25em}% arbre rouge noir, nil
22000 @@ -258,8 +277,8 @@ que aponta para o filho esquerdo.
22001 O primeiro caso é tratado no segundo \enf{if} do segundo \enf{while}, o primeiro \enf{if}
22002 decide se está sendo incluído pelo o lado esquerdo e da linha 99-106, temos o tratamento
22003 do \enf{caso 1}, nó pai vermelho, e nó tio vermelho, segue um diagrama deste caso em
22004 -nome da clereza.
22005
22006 +nome da clereza.\\
22007 +\begin{minipage}{20cm}
22008 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22009 level distance = .5cm}]
22010 \node [arn_n] {}
22011 @@ -322,7 +341,8 @@ nome da clereza.
22012 }
22013 }
22014 ;
22015 -\end{tikzpicture} \\

```



```

22016 +\end{tikzpicture}
22017 +\end{minipage}
22018 Note que saímos do \enf{caso 1} e caímos no \enf{caso 2}, nó pai do nó agora
22019 apontado por \enf{z}, vermelho e nó tio preto \enf{y}, lembre-se que estamos dentro de um
22020 \enf{while} em loop enquanto a cor do pai de \enf{z} for vermelha, este segundo caso
22021 @@ -331,7 +351,8 @@ analisarmos o que faz esta função.
22022 \lstinputlisting [firstnumber=319, linerange={319-342}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
22023 Esta função \enf{rb_rot_esq}, basicamente diminui a altura do lado direito e aumenta
22024 do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
22025 -\begin{table}[!ht]
22026 +
22027 +\begin{mytable}[colback=black!5]{rotação à esquerda}
22028 \scriptsize
22029 \begin{tabular}{|l|l|}
22030 \hline
22031 @@ -364,7 +385,42 @@ do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
22032 341 & retorna inteiro para uso em testes (não utilizado neste trabalho, mas a função tem esta interface)\\
22033 \hline
22034 \end{tabular}
22035 -\end{table}
22036 +\end{mytable}
22037 +
22038 +% \begin{table}[!ht]
22039 +% \scriptsize
22040 +% \begin{tabular}{|l|l|}
22041 +% \hline
22042 +% 327 & guarda referência do filho a direita \\
22043 +% \hline
22044 +% 328 & faz filho a direita igual a neto a esquerda (diminuindo a altura direita) \\
22045 +% \hline
22046 +% 329 & se filho a direita (antigo neto a esquerda) não é sentinela \\
22047 +% \hline
22048 +% 330 & atualiza pai de filho a direita (antigo neto a esquerda)\\
22049 +% \hline
22050 +% 331 & atualiza pai do nó movido\\
22051 +% \hline
22052 +% 332 & testa se nó anterior era raiz \\
22053 +% \hline
22054 +% 333 & atualiza nó raiz \\
22055 +% \hline
22056 +% 334 & se não é raiz testa se é nó a esquerda\\
22057 +% \hline
22058 +% 335 & atualiza como nó a esquerda\\
22059 +% \hline
22060 +% 336 & se nenhuma das opções anteriores\\
22061 +% \hline
22062 +% 337 & atualiza como nó a direita\\
22063 +% \hline
22064 +% 338 & atualiza filho a esquerda de y igual x, note que o nó x que era pai passa a ser filho\\
22065 +% \hline
22066 +% 339 & atualiza pai de x igual de y \\
22067 +% \hline
22068 +% 341 & retorna inteiro para uso em testes (não utilizado neste trabalho, mas a função tem esta interface)\\
22069 +% \hline
22070 +% \end{tabular}
22071 +% \end{table}
22072
22073 \noindent Um diagrama em nome da clareza, clro que aqui está representado apenas os nós
22074 que são representativos de uma rotação a esquerda e não árvores completas.\\
22075 diff --git a/doc/tex/table.lot b/doc/tex/table.lot
22076 new file mode 100644
22077 index 00000000..6a84ce8
22078 --- /dev/null
22079 +++ b/doc/tex/table.lot
22080 @@ -0,0 +1,3 @@
22081 +\contentsline {table}{\numberline {1}{\ignorespaces First table}}{1}{}%
22082 +\contentsline {colorbox}{\numberline {2}{\ignorespaces Second table}}{1}{}%
22083 +\contentsline {colorbox}{\numberline {3}{\ignorespaces Trird table}}{1}{}%
22084 diff --git a/doc/tex/table.pdf b/doc/tex/table.pdf
22085 new file mode 100644
22086 index 00000000..f7e3592
22087 Binary files /dev/null and b/doc/tex/table.pdf differ
22088 diff --git a/doc/tex/table.tex b/doc/tex/table.tex
22089 new file mode 100644
22090 index 00000000..4a270ee
22091 --- /dev/null
22092 +++ b/doc/tex/table.tex
22093 @@ -0,0 +1,60 @@
22094 +\documentclass{article}
22095 +\usepackage{booktabs}
22096 +\usepackage{most}{tcolorbox}
22097 +
22098 +\newtcolorbox[blend into=tables]{mytable}[2][]{%
22099 +  enhanced,
22100 +  float,
22101 +  every float=\centering,
22102 +  capture=hbox,
22103 +  title = #2,
22104 +  attach boxed title to top left={%
22105 +    xshift=5mm
22106 +    yshift=-\tcboxedtitleheight/2,
22107 +    yshifttext=-1mm},
22108 +  boxed title style={colback=blue!50, sharp corners},
22109 +  colframe = blue!50,
22110 +  colback = blue!20,
22111 +  % overlay = {\node[text=white, fill=red] at (frame.east)
22112 +  %   {\clubsuit$};},
22113 +  #1}
22114 +
22115 +\begin{document}
22116 +\listoftables
22117 +
22118 +\begin{table}
22119 +\centering
22120 +\begin{tabular}{|r|l|}
22121 +\hline
22122 + 7C0 & hexadecimal \\
22123 + 3700 & octal \\
22124 + 1111000000 & binary \\
22125 + \hline \hline
22126 + 1984 & decimal \\
22127 + \hline
22128 +\end{tabular}
22129 +\caption{First table}
22130 +\end{table}
22131 +

```



```

22132 +\begin{mytable}{Second table}
22133 +\begin{tabular}{rl}
22134 + 7C0 & \hexadecimal \\
22135 + 3700 & \octal \\
22136 + 11111000000 & \binary \\
22137 + \midrule
22138 + 1984 & \decimal \\
22139 +\end{tabular}
22140 +\end{mytable}
22141 +
22142 +\begin{mytable}[colback=red!5]{Trird table}
22143 +\begin{tabular}{l*{6}{c}r}
22144 +Team & P & W & D & L & F & A & Pts \\
22145 +\midrule
22146 +Manchester United & 6 & 4 & 0 & 2 & 10 & 5 & 12 \\
22147 +Celtic & 6 & 3 & 0 & 3 & 8 & 9 & 9 \\
22148 +Benfica & 6 & 2 & 1 & 3 & 7 & 8 & 7 \\
22149 +FC Copenhagen & 6 & 2 & 1 & 3 & 5 & 8 & 7 \\
22150 +\end{tabular}
22151 +\end{mytable}
22152 +
22153 +\end{document}
22154
22155 commit 934bdf72c946071e5b7e1f05a296e6b37554e91b
22156 Author: Joilnen <joilnen@gmail.com>
22157 Date: Fri Jan 14 14:24:24 2022 -0300
22158
22159 melhorando a formatacao, mudando o sublinhado dos titulos por bold,
22160 pra funcionar eh compativel com vt100
22161
22162 diff --git a/src/checklist.c b/src/checklist.c
22163 index 399bfb26..c61a11b 100644
22164 --- a/src/checklist.c
22165 +++ b/src/checklist.c
22166 @@ -12,7 +12,7 @@ void tela_checklist(int i)
22167 static unsigned int cklst_estado = 0;
22168 cklst_estado |= i ;
22169
22170 - CKLSITEM(1++, c + 11, 0, "avaliacao");
22171 + CKLSITEM(1++, c + 11, 0, S_UNDERL"avaliacao"S_NORM);
22172 CKLSITEM(1++, c, 0, " ");
22173 CKLSITEM(1++, c, cklst_estado & bo, ITEM1_1); bo <= 1;
22174 CKLSITEM(1++, c, cklst_estado & bo, ITEM1_2); bo <= 1;
22175 diff --git a/src/jcurses.h b/src/jcurses.h
22176 index 724ae59..aa58c52 100644
22177 --- a/src/jcurses.h
22178 +++ b/src/jcurses.h
22179 @@ -17,7 +17,7 @@
22180 #define S_LTELA "\033[H\033[J"
22181 #define S_CM "\033[24;62H"
22182 #define S_CMR "\033[24;58H"
22183 -#define S_UNDERL "\033[21m"
22184 +#define S_UNDERL "\033[1m"
22185 #define S_AZUL "\033[44m\033[37m"
22186 #define S_VERM "\033[41m\033[37m"
22187 #define S_INV "\033[7m"
22188 diff --git a/src/testa_rb.c b/src/testa_rb.c
22189 index 149abad..52bbd2c 100644
22190 --- a/src/testa_rb.c
22191 +++ b/src/testa_rb.c
22192 @@ -14,15 +14,16 @@
22193 void tela_testa_rb(const char *m)
22194 {
22195 int l = 5, c = 24, f;
22196 - char mens[6][48] =
22197 + char mens[5][48] =
22198 {
22199 S_UNDERL"serah efetuado os testes:"S_NORM,
22200 "insercao e remocao na arvore",
22201 "nao foi pedido mas me ajudou a",
22202 - "checar a correcao da arvore",
22203 + "checar a correcao da arvore",
22204 + "chaves randomicas de 01 a 99"
22205 };
22206 LTELA;
22207 - for (f = 0; f < 4; f++)
22208 + for (f = 0; f < 5; f++)
22209 CKLSITEM(1++, c, 0, mens[f]);
22210 INFO_FILE(__FILE__);
22211 CM;
22212
22213 commit 8bd521ac056bc1b2d26793862c18a64cc051759f
22214 Author: Joilnen <joilnen@gmail.com>
22215 Date: Fri Jan 14 15:07:27 2022 -0300
22216
22217 melhorando a info sobre o codigo executado, nome do arquivo e funcao
22218
22219 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
22220 index 52406e9..ca7663e 100644
22221 --- a/src/conjunto_ordenado.h
22222 +++ b/src/conjunto_ordenado.h
22223 @@ -151,7 +151,7 @@ void conjord_imprime_venn(conjord *c);
22224 #define CKLSITEM(l,c,b,x) \
22225 if (b) printf(S_CHKLIST_DONE, l, c, x); \
22226 else printf(S_CHKLIST, l, c, x)
22227 -#define INFO_FILE(x) printf("%s%s", S_INFO_ST, x, S_NORM)
22228 +#define INFO_FILE printf("%s%s", S_INFO_ST, __FILE__, S_INFO_ST_1, __FUNCTION__, S_NORM); fflush(stdout);
22229 #define INFO_CARD(x) printf("%s", S_CARD, x, S_NORM)
22230
22231 #endif
22232 diff --git a/src/jcurses.h b/src/jcurses.h
22233 index aa58c52..5b2940c 100644
22234 --- a/src/jcurses.h
22235 +++ b/src/jcurses.h
22236 @@ -23,7 +23,8 @@
22237 #define S_INV "\033[7m"
22238 #define S_NORM "\033[0m"
22239 #define S_INFO "\033[24;0H\033[7m"
22240 -#define S_INFO_ST "\033[0;65H"
22241 +#define S_INFO_ST "\033[1;65H"
22242 +#define S_INFO_ST_1 "\033[2;65H"
22243 #define S_CARD "\033[0;60H"
22244 #define S_CHKLIST "\033[%d;%dH\033[0m"
22245 #define S_CHKLIST_DONE "\033[46m\033[30m\033[%d;%dH\033[0m"
22246 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
22247 index ba28c67..d28ca84 100644

```

```

22248 --- a/src/testa_item_1.c
22249 +++ b/src/testa_item_1.c
22250 @@ -93,7 +93,7 @@ void tela_item_1(const char *m)
22251     CKLSITEM(1++, c, 0, "e ordenacao dos elementos do conjunto");
22252     CKLSITEM(1++, c, 0, "o tipo de dado do usuario aqui eh uma");
22253     CKLSITEM(1++, c, 0, "estrutura com duas strings, nome e telefone");
22254 - INFO_FILE(__FILE__);
22255 + INFO_FILE;
22256     CM;
22257 }
22258
22259 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
22260 index 1da9e99..af21392 100644
22261 --- a/src/testa_item_2.c
22262 +++ b/src/testa_item_2.c
22263 @@ -97,7 +97,7 @@ void tela_item_2(const char *m)
22264     };
22265     for (f = 0; f < 8; f++)
22266         CKLSITEM(1++, c, 0, mens[f]);
22267 - INFO_FILE(__FILE__);
22268 + INFO_FILE;
22269     CM;
22270 }
22271
22272 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
22273 index 088ba21..5bd780b 100644
22274 --- a/src/testa_item_3.c
22275 +++ b/src/testa_item_3.c
22276 @@ -77,6 +77,7 @@ void testa_item_3(const char *m)
22277     printf("\033[10;30H");
22278     conjord_imprime_venn(conjord_une(cj_a, cj_b));
22279     INFO("= item 3.1 uniao de conjuntos");
22280 + INFO_FILE;
22281     CM;
22282
22283     LTEL;
22284 @@ -89,6 +90,7 @@ void testa_item_3(const char *m)
22285     printf("\033[10;30H");
22286     conjord_imprime_venn(conjord_subtrai(cj_a, cj_b));
22287     INFO("= item 3.1 subtracao de conjuntos");
22288 + INFO_FILE;
22289     CM;
22290
22291     LTEL;
22292 @@ -101,6 +103,7 @@ void testa_item_3(const char *m)
22293     printf("\033[10;30H");
22294     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b));
22295     INFO("= item 3.1 interseccao com conjuntos");
22296 + INFO_FILE;
22297     CM;
22298
22299     LTEL;
22300
22301 commit e844e278d872ead6fbbb2536c048d46429841f7f
22302 Author: Joilnen <joilnen@gmail.com>
22303 Date: Fri Jan 14 15:17:34 2022 -0300
22304
22305     dicionando autoria na primeira tela
22306
22307 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
22308 index d28ca84..8d49570 100644
22309 --- a/src/testa_item_1.c
22310 +++ b/src/testa_item_1.c
22311 @@ -93,6 +93,8 @@ void tela_item_1(const char *m)
22312     CKLSITEM(1++, c, 0, "e ordenacao dos elementos do conjunto");
22313     CKLSITEM(1++, c, 0, "o tipo de dado do usuario aqui eh uma");
22314     CKLSITEM(1++, c, 0, "estrutura com duas strings, nome e telefone");
22315 + CKLSITEM(1++, c, 0, "");
22316 + CKLSITEM(1++, c, 0, "          autor: joilnen leite");
22317     INFO_FILE;
22318     CM;
22319 }
22320
22321 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
22322 index 5bd780b..c6d0004 100644
22323 --- a/src/testa_item_3.c
22324 +++ b/src/testa_item_3.c
22325 @@ -30,7 +30,6 @@ static void adiciona_venn(conjord *c, int pos, char *letras)
22326     if (!conjord_insere_elemento(c, *i, mt0))
22327         ++i;
22328     }
22329 }
22330
22331 void tela_item_3(const char *m)
22332
22333 commit e8be6243a11ac1ae644fad11c0069c80e288c9e4
22334 Author: Joilnen <joilnen@gmail.com>
22335 Date: Sun Jan 16 15:48:31 2022 -0300
22336
22337     documentacao
22338
22339 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
22340 index 5d71172..cecdf1e 100644
22341 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
22342 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
22343 index 59a8375..d2361cd 100644
22344 --- a/doc/tex/relatorio.tex
22345 +++ b/doc/tex/relatorio.tex
22346 @@ -93,6 +93,7 @@
22347 \newcommand{\enf}[1]{\emph{\textbf{#1}}}
22348
22349 \graphicspath{{fig/}}
22350 +\renewcommand{\baselinestretch}{.7}
22351
22352 \begin{document}
22353 \normalfont
22354 @@ -114,27 +115,11 @@
22355 \date{}
22356 }
22357 \maketitle
22358 -% \thispagestyle{empty}
22359 -% \begin{center}
22360 -% \Large{\textit{\textbf{Resenha Critica do ensaio,\ "Ética e Pesquisa: Autonomia e Heteronomia na Prática Científica}}}
22361 -% \footnote{Artigo escrito por Antonio Joaquim Severino}
22362 -% \ \

```

```

22363 -% \end{center}
22364 -% \hrule
22365 -% \/\
22366 -% \/\
22367 -% \normalsize{\textit{\textbf{Autor: Joilnen Leite}}}\
22368 -% \normalsize{\textit{\textbf{Disciplina: Metodologia Científica}}}\
22369 -% \normalsize{\textit{\textbf{Curso: Ciência da Computação}}}\
22370 -% \normalsize{\textit{\textbf{Data: 24 de junho de 2020}}}\
22371 -% \/\
22372 -% \hrule
22373 -% \tableofcontents
22374 -% \vspace*{2cm}
22375 -\textbf{Resumo: } Relatório básico sobre o conteúdo e processo de desenvolvimento
22376 +
22377 +\noindent \textbf{Resumo: } Relatório básico sobre o conteúdo e processo de desenvolvimento
22378 da atividade conjunto ordenado \/\
22379 \/\
22380 -\indent \textbf{Palavras-chave: } fontes, C, Red Black, Conjunto\
22381 +\noindent \textbf{Palavras-chave: } fontes, C, Red Black, Conjunto\
22382 \small
22383 \section{Introdução}
22384 \noindent Esta biblioteca é composta pelo os seguintes arquivos, \/\
22385 @@ -146,7 +131,7 @@ da atividade conjunto ordenado \/\
22386 \item [\blacksquare] jcurses.h
22387 \end{itemize}
22388
22389 -Completamente separado do código da biblioteca temos alguns arquivos que implementam
22390 +\noindent Completamente separado do código da biblioteca temos alguns arquivos que implementam
22391 testes, estes são:
22392 \begin{itemize}
22393 \item [\blacksquare] main.c
22394 @@ -165,7 +150,7 @@ e tem suas funções chamadas sequencialmente dentro da função \enf{main}, no
22395 Seguiremos neste relatório uma abordagem \enf{top-down} onde partiremos das estruturas
22396 e funções manipuladas pelo o código cliente em direção as estruturas e funções que implementam
22397 e operam na estrutura de dados, \enf{árvore red-black}, que é totalmente ocultada do cliente
22398 -ou seja poderíamos reimplementar as funcionalidade com outras esturas de dados e manter
22399 +ou seja poderíamos reimplementar as funcionalidades com outras esturas de dados e manter
22400 a interface compatível com a existente.
22401
22402 O estilo do código fonte neste trabalho é o mais tradicional, chaves abrem e fecham do mesmo lado
22403 @@ -182,7 +167,7 @@ documentação esta constante nos anexos.
22404 \section{Primeiras Estruturas}
22405 \subsection{conjord}
22406 A primeira estrutura que veremos aqui é a \enf{conjord} ela representa um conjunto ordenando
22407 -é o nosso envólucro sobre a estrutura da árvore,
22408 +é o nosso envólucro sobre a estrutura da \enf{árvore red-black}.
22409
22410 \lstinputlisting [linrange={11-31}, firstnumber=11, caption = {Fragmento do conjunto_ordenado.h}]{../src/
conjunto_ordenado.h}
22411
22412 @@ -221,12 +206,12 @@ esta função a seguir,
22413 \lstinputlisting [firstnumber=39, linerange={39-65}, caption = {Fragmento do conjunto_ordenado.c}]{../src/
conjunto_ordenado.c}
22414 o \enf{if}, na linha 47, adiciona um elemento na árvore já existente, pois note que \enf{pr}
22415 diferente de \enf{rb_s}, o que significa que já existe árvore criada para este \enf{conjord},
22416 -caso contrário é criada a raiz da árvore e apontado o \enf{pr} para ela, \enf{rb_cr},
22417 +caso contrário é criada a raiz da árvore e apontado o \enf{pr} para ela, \enf{rb_cr} linha 60,
22418 cria uma árvore nova retorna o nó raiz tendo uma chave e um tipo qualquer(\enf{void *}),
22419 cardinalidade é incrementado, e no caso da inserção em árvore pre-existente,
22420 ele só é incrementado caso a função \enf{rb_insert}, retorne código de inserido, zero,
22421 temos como retorno dessa função de inserção no conjunto, zero para inserido, e -1,
22422 -caso o contrário, é assim que nos certificamos de ter inserido um elemento realmente
22423 +caso o contrário, é assim que nos certificamos de ter inserido um elemento realmente de chave única
22424 como no código de teste visto anteriormente, note que usamos ponteiro duplo na raiz,
22425 por que em algumas operações teremos que modificar esse ponteiro dentro de funções chamadas,
22426 como em rotações e delegações.
22427 @@ -243,18 +228,20 @@ red black e insere um novo nó.
22428 \lstinputlisting [firstnumber=26, linerange={25-41}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
22429 A mesma ideia que vimos na função \enf{conjord_insere_elemento}, alocamos aqui um novo nó,
22430 populamos com chave e o dado genérico, também setamos a cor preta, pois é raiz, apontamos
22431 -ponteiros, pai, e filhos para nó sentinela \enf{rb_s}
22432 +ponteiros, pai, e filhos para nó sentinela \enf{rb_s}, por fim retornamos o nó raiz criado.
22433
22434 Antes de passarmos para o \enf{rb_insert}, vamos dar uma olha na estrutura nó \enf{rb_n}
22435 -\lstinputlisting [firstnumber=33, linerange={33-48}, caption = {Fragmento do red_black.h}]{../src/red_black.h}, a
22436 documentação de comentário é explícita.
22437 +\lstinputlisting [firstnumber=33, linerange={33-48}, caption = {Fragmento do red_black.h}]{../src/red_black.h},
22438 +trata-se da estrutura que abstrai um nó da árvore, comentário no código fonte descreve cada parâmetro.
22439
22440 -Então agora a função \enf{rb_insert} que insere um novo nó na árvore, detalharemos agora
22441 +Então a função \enf{rb_insert} que insere um novo nó na árvore, detalharemos agora
22442 o funcionamento desta função
22443 \lstinputlisting [firstnumber=43, linerange={43-145}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
22444
22445 Tendo apenas um único nó esta rotina apenas atualiza o ponteiro raiz para este nó,
22446 na linha 76, faz o nó incluído apontar seu ponteiro pai para o sentinela, através de \enf{y}
22447 -(inicializado com sentinela em 55) e atualiza o nó raiz em 86 colorindo em seguida com preto em 142.
22448 +(inicializado com sentinela em 55) e atualiza o nó raiz em 86 colorindo em seguida colore em 142,
22449 +pois toda raiz é preta na \enf{árvore red-black}.
22450
22451 A primeira parte desta rotina, que inclui o primeiro \enf{while} até o
22452 primeiro \enf{if}, apenas faz a inclusão do novo nó da mesma forma que na árvore binária de
22453 @@ -264,19 +251,19 @@ e uma vez que temos a \enf{y} apontando para este nó folha, anexamos \enf{z} a
22454 fazendo o apontamento do ponteiro filho neste nó folha para \enf{z}, a única grande
22455 diferença é que no final desta parte colorimos com vermelho, o novo nó incluído.
22456
22457 -Todas as inclusões entram numa etapa, o segundo \enf{while}, que chamei de \enf{ajuste}
22458 -na bibliografia é referida como \enf{fixup}, esta parte do código faz o ajuste de coloração
22459 -e rotaciona nós para manter as propriedades da árvore red-black obedecidas.
22460 +Todas as inclusões entram numa etapa, o segundo \enf{while}, que chamei de \enf{ajuste}
22461 +na bibliografia é referida como \enf{fixup}, esta parte do código faz o ajuste de coloração
22462 +e rotaciona nós, caso necessário, para manter as propriedades da árvore red-black obedecidas.
22463
22464 Vejamos como esta parte da rotina trata os três casos em que podem ser quebradas
22465 -as propriedades da árvore na inclusão e como esta parte do código resolve as inconsistências.
22466 +as propriedades da árvore na inclusão e como esta parte do código resolve estas inconsistências.
22467
22468 Considerando a priori apenas as inclusões que ocorrerão a esquerda, nós anexados ao ponteiro
22469 que aponta para o filho esquerdo.
22470
22471 O primeiro caso é tratado no segundo \enf{if} do segundo \enf{while}, o primeiro \enf{if}
22472 -decide se está sendo incluído pelo o lado esquerdo e da linha 99-106, temos o tratamento
22473 -do \enf{caso 1}, nó pai vermelho, e nó tio vermelho, segue um diagrama deste caso em
22474 +decide se está sendo incluído pelo o lado esquerdo, linhas 99-106, temos o tratamento
22475 +do \enf{caso 1}, nó pai vermelho, e nó tio vermelho, segue um diagrama deste caso em
22476 nome da clreza.\

```

```

22476 \begin{minipage}{20cm}
22477 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22478 @0 -345,7 +332,7 @0 nome da clerezza.\}
22479 \end{minipage}
22480 Note que saímos do \enf{caso 1} e caímos no \enf{caso 2}, nó pai do nó agora
22481 apontado por \enf{z}, vermelho e nó tio preto \enf{y}, lembre-se que estamos dentro de um
22482 -\enf{while} em loop enquanto a cor do pai de \enf{z} for vermelha, este segundo caso
22483 +\enf{while} em loop enquanto a cor do pai de \enf{z} for vermelha. Este segundo caso
22484 é tratado no \enf{if} 109-113, onde há uma rotação a esquerda, vamos aproveitar para
22485 analisarmos o que faz esta função.
22486 \listinputlisting [firstnumber=319, linerange={319-342}, caption = {Fragmento do red_black.c}] {../../src/red_black.c}
22487 @0 -422,7 +409,7 @0 do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\}
22488 % \end{tabular}
22489 % \end{table}
22490
22491 -\noindent Um diagrama em nome da clerezza, clro que aqui está representado apenas os nós
22492 +\noindent Um diagrama em nome da clerezza, claro que aqui está representado apenas os nós
22493 que são representativos de uma rotação a esquerda e não árvores completas.\}
22494 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22495 level distance = .5cm}]
22496 @0 -464,11 +451,11 @0 que são representativos de uma rotação a esquerda e não árvores completas.\}
22497 ;
22498 \end{tikzpicture} \}
22499 Continuando a análise da função \enf{rb_insert}, do ponto que interrompemos para entendermos o \enf{rb_rot_esq},
22500 -após estar rotação a esquerda a árvore neste ponto passa para o \enf{caso 3} e a será feito uma rotação a direita
22501 +após esta rotação a esquerda a árvore neste ponto passa para o \enf{caso 3} e a será feito uma rotação a direita
22502 \enf{z} tem o pai e o avô colorido com preto e vermelho respectivamente e o avô rotacionado a direita.
22503 Repetindo aqui o fragmento do código anterior \enf{rb_insert}, trata o \enf{caso 3} linhas 114-116.
22504 \listinputlisting [firstnumber=97, linerange={97-118}, caption = {Fragmento do red_black.c}] {../../src/red_black.c}
22505 -\begin{table}[!ht]
22506 +\begin{mytable}[colback=black!5]{caso 3}
22507 \scriptsize
22508 \begin{tabular}{|l|l|}
22509 \hline
22510 @0 -476,51 +463,10 @0 Repetindo aqui o fragmento do código anterior \enf{rb_insert}, trata o \enf{caso
22511 \hline
22512 115 & colore avô de \enf{z} com vermelho \}
22513 \hline
22514 - 116 & rotaciona avô a direita \}
22515 + 116 & rotaciona avô a direita \}
22516 \hline
22517 \end{tabular}
22518 -\end{table}
22519 -\}
22520 -Um diagrama em nome da clerezza\}
22521 -\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22522 level distance = .5cm}]
22523 -\node [arn_n] {x}
22524 - child{ node [arn_n] {1}
22525 - child { node [arn_x] {} }
22526 - child { node [arn_x] {} }
22527 - }
22528 - child{ node [arn_n] {y}
22529 - child{ node [arn_r] {5}
22530 - child { node [arn_x] {} }
22531 - child { node [arn_x] {} }
22532 - }
22533 - child{ node [arn_r] {8}
22534 - child { node [arn_x] {} }
22535 - child { node [arn_x] {} }
22536 - }
22537 - }
22538 -;
22539 -\end{tikzpicture}
22540 -\colorbox{BurntOrange}{\scriptsize após \enf{rb_rot_esq} }
22541 -\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22542 level distance = .5cm}]
22543 -\node [arn_n] {y}
22544 - child{ node [arn_n] {x}
22545 - child{ node [arn_r] {1}
22546 - child { node [arn_x] {} }
22547 - child { node [arn_x] {} }
22548 - }
22549 - child{ node [arn_r] {5}
22550 - child { node [arn_x] {} }
22551 - child { node [arn_x] {} }
22552 - }
22553 - }
22554 - child{ node [arn_n] {8}
22555 - child { node [arn_x] {} }
22556 - child { node [arn_x] {} }
22557 - }
22558 -;
22559 -\end{tikzpicture} \}
22560 +\end{mytable}
22561 Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
22562 é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
22563 tio a direita. O diagrama abaixo representa esta
22564 @0 -602,6 +548,12 @0 tio a direita. O diagrama abaixo representa esta
22565 \}
22566 \}
22567 \}
22568 +\begin{enumerate}
22569 + \item \enf{z} será root
22570 + \item tio de \enf{z} vermelho
22571 + \item tio de \enf{z} preto em forma de triângulo
22572 + \item tio de \enf{z} preto em forma de linha
22573 +\end{enumerate}
22574 %%
22575 %%
22576 %% daqui pra frente eh antigo
22577
22578 commit 62f5f42841c1ce06dc91e4da71139d87b5317aa8
22579 Author: Joilnen <joilnen@gmail.com>
22580 Date: Sun Jan 16 23:55:44 2022 -0300
22581
22582 documentacao
22583
22584 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
22585 index cecdfe..eaff269 100644
22586 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
22587 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
22588 index d2361cd..925b6ff 100644
22589 --- a/doc/tex/relatorio.tex
22590 +++ b/doc/tex/relatorio.tex
22591 @0 -235,106 +235,125 @0 Antes de passarmos para o \enf{rb_insert}, vamos dar uma olha na estrutura nó \}

```

```

22592 trata-se da estrutura que abstrai um nó da árvore, comentário no código fonte descreve cada parâmetro.
22593
22594 Então a função \enfr{insert} que insere um novo nó na árvore, detalharemos agora
22595 -o funcionamento desta função
22596 +o funcionamento desta função.
22597 \lstinputlisting [firstnumber=43, linerange={43-145}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
22598
22599 -Tendo apenas um único nó esta rotina apenas atualiza o ponteiro raiz para este nó,
22600 +Consideremos estes 4 casos bases:
22601 +\begin{center}
22602 + \begin{minipage}{0.4\textwidth}
22603 + \begin{enumerate}
22604 + \setcounter{enumi}{-1}
22605 + \item \enfr{z} será root
22606 + \item tio de \enfr{z} vermelho
22607 + \item tio de \enfr{z} preto em forma de triângulo
22608 + \item tio de \enfr{z} preto em forma de linha
22609 + \end{enumerate}
22610 + \end{minipage}
22611 + \end{center}
22612 +
22613 +\enfr{Caso 0}, tendo apenas um único nó esta rotina apenas atualiza o ponteiro raiz para este nó,
22614 na linha 76, faz o nó incluído apontar seu ponteiro pai para o sentinela, através de \enfr{y}
22615 -(inicializado com sentinela em 55) e atualiza o nó raiz em 86 colorindo em seguida o nó em 142,
22616 +(inicializado com sentinela em 55) e atualiza o nó raiz em 86 colorindo em seguida com a cor preta em 142,
22617 pois toda raiz é preta na \enfr{árvore red-black}.
22618
22619 -A primeira parte desta rotina, que inclui o primeiro \enfr{while} até o
22620 +
22621 +A primeira parte desta rotina, composta do primeiro \enfr{while} até o
22622 primeiro \enfr{if}, apenas faz a inclusão do novo nó da mesma forma que na árvore binária de
22623 busca comum, o while itera até a folha comparando as chaves, neste caso aqui, foi
22624 modificado para aceitar apenas chaves diferentes, retornando código de erro caso contrário,
22625 e uma vez que temos a \enfr{y} apontando para este nó folha, anexamos \enfr{z} a este nó
22626 -fazendo o apontamento do ponteiro filho neste nó folha para \enfr{z}, a única grande
22627 -diferença é que no final desta parte colorimos com vermelho, o novo nó incluído.
22628 -fazendo o apontamento do ponteiro filho deste nó folha para \enfr{z}, a única grande
22629 +diferença é que no final desta etapa colorimos com vermelho, linha 92, o novo nó incluído.
22630
22631 -Todas as inclusões entram numa etapa, o segundo \enfr{while}, que chamei de \enfr{ajuste}
22632 +O segundo \enfr{while}, que chamei de \enfr{ajuste}
22633 na bibliografia é referida como \enfr{fixup}, esta parte do código faz o ajuste de coloração
22634 e rotaciona nós, caso necessário, para manter as propriedades da árvore red-black obedecidas.
22635
22636 -Vejamos como esta parte da rotina trata os três casos em que podem ser quebradas
22637 +Note que o while permanece em loop enquanto o pai de \enfr{z}, não a ser incluído, é também
22638 +vermelho, lembre que não podemos ter nó vermelho filho de outro nó vermelho.
22639 +
22640 +Vejamos como esta parte da rotina trata os três casos restantes, em que podem ser quebradas
22641 as propriedades da árvore na inclusão e como esta parte do código resolve estas inconsistências.
22642
22643 Considerando a priori apenas as inclusões que ocorrerão a esquerda, nós anexados ao ponteiro
22644 que aponta para o filho esquerdo.
22645
22646 -O primeiro caso é tratado no segundo \enfr{if} do segundo \enfr{while}, o primeiro \enfr{if}
22647 -decide se está sendo incluído pelo o lado esquerdo, linhas 99-106, temos o tratamento
22648 -do \enfr{caso 1}, nó pai vermelho, e nó tio vermelho, segue um diagrama deste caso em
22649 -nome da clereza.\\
22650 -\begin{minipage}{20cm}
22651 -\begin{tikzpicture}[>,>=stealth',level/.style={sibling distance = 2.5cm/#1,
22652 - level distance = .5cm}]
22653 -\node [arn_n] {}
22654 -\child{ node [arn_r] {}
22655 -\child{ node [arn_n] {}
22656 -\child { node [arn_x] {} }
22657 -\child { node [arn_x] {} }
22658 -}
22659 -\child{ node [arn_n] {}
22660 -\child{ node [arn_r] {}
22661 -\child{ node [arn_r] {z}
22662 +\enfr{Caso 1}, nó tio de \enfr{z} é vermelho, é tratado no segundo \enfr{if} do segundo
22663 +\enfr{while}, o primeiro \enfr{if} decide se está sendo incluído pelo o lado esquerdo,
22664 +linhas 99-106, temos o tratamento do \enfr{caso 1}, nó pai vermelho, e nó tio vermelho,
22665 +segue um diagrama deste caso em nome da clereza.\\
22666 +\begin{center}
22667 + \begin{minipage}{.95\textwidth}
22668 + \begin{tikzpicture}[>,>=stealth',level/.style={sibling distance = 2.5cm/#1,
22669 + level distance = .5cm}]
22670 + \node [arn_n] {}
22671 + \child{ node [arn_r] {}
22672 + \child{ node [arn_n] {}
22673 + \child { node [arn_x] {} }
22674 + \child { node [arn_x] {} }
22675 + }
22676 + \child { node [arn_x] {} }
22677 + \child{ node [arn_n] {}
22678 + \child{ node [arn_r] {}
22679 + \child{ node [arn_r] {z}
22680 + \child { node [arn_x] {} }
22681 + \child { node [arn_x] {} }
22682 + }
22683 + \child { node [arn_x] {} }
22684 + }
22685 + \child{ node [arn_r] {y}
22686 + \child { node [arn_x] {} }
22687 + \child { node [arn_x] {} }
22688 + }
22689 + }
22690 + }
22691 + \child{ node [arn_r] {y}
22692 + \child { node [arn_x] {} }
22693 + \child{ node [arn_n] {}
22694 + \child { node [arn_x] {} }
22695 + \child{ node [arn_r] {}
22696 + \child { node [arn_x] {} }
22697 + \child { node [arn_x] {} }
22698 + }
22699 + }
22700 + }
22701 + }
22702 + \child{ node [arn_n] {}
22703 + \child { node [arn_x] {} }
22704 + \child{ node [arn_r] {}
22705 + \child { node [arn_x] {} }
22706 + \child { node [arn_x] {} }
22707 + }

```

```

22708 - }
22709 -;
22710 -\end{tikzpicture}
22711 -\colorbox{BurntOrange}{\scriptsize após o \enf{if} nas linhas, 99-106 }
22712 -\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22713 - level distance = .5cm}]
22714 -\node [arn_n] {}
22715 - child{ node [arn_r] {}
22716 - child{ node [arn_n] {}
22717 - child { node [arn_x] {} }
22718 - child { node [arn_x] {} }
22719 - }
22720 - child{ node [arn_r] {z}
22721 - child{ node [arn_n] {}
22722 - child{ node [arn_r] {}
22723 - }
22724 - \end{tikzpicture}
22725 - \colorbox{BurntOrange}{\scriptsize após o \enf{if} nas linhas, 99-106 }
22726 - \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22727 - level distance = .5cm}]
22728 - \node [arn_n] {}
22729 - child{ node [arn_r] {}
22730 - child{ node [arn_n] {}
22731 - child { node [arn_x] {} }
22732 - child { node [arn_x] {} }
22733 - }
22734 - child { node [arn_x] {} }
22735 - child{ node [arn_r] {z}
22736 - child{ node [arn_n] {}
22737 - child{ node [arn_r] {}
22738 - child { node [arn_x] {} }
22739 - child { node [arn_x] {} }
22740 - }
22741 - child { node [arn_x] {} }
22742 - }
22743 - child{ node [arn_n] {}
22744 - child { node [arn_x] {} }
22745 - child { node [arn_x] {} }
22746 - }
22747 - }
22748 - }
22749 - child{ node [arn_n] {}
22750 - child { node [arn_x] {} }
22751 - child{ node [arn_n] {y}
22752 - child { node [arn_x] {} }
22753 - child{ node [arn_r] {}
22754 - child { node [arn_x] {} }
22755 - child { node [arn_x] {} }
22756 - }
22757 - }
22758 - }
22759 - }
22760 - child{ node [arn_n] {y}
22761 - child { node [arn_x] {} }
22762 - child{ node [arn_r] {}
22763 - child { node [arn_x] {} }
22764 - child { node [arn_x] {} }
22765 - }
22766 - }
22767 -;
22768 -\end{tikzpicture}
22769 -\end{minipage}
22770 +
22771 + \end{tikzpicture}
22772 + \end{minipage}
22773 + \end{center}
22774 + Note que saímos do \enf{caso 1} e caímos no \enf{caso 2}, nó pai do nó agora
22775 + apontado por \enf{z}, vermelho e nó tio preto \enf{y}, lembre-se que estamos dentro de um
22776 + \enf{while} em loop enquanto a cor do pai de \enf{z} for vermelha. Este segundo caso
22777 + é tratado no \enf{if} 109-113, onde há uma rotação a esquerda, vamos aproveitar para
22778 + analisarmos o que faz esta função.
22779 + apontado por \enf{z}, vermelho e nó tio preto \enf{y}, em formato de triângulo,
22780 + lembre-se que estamos dentro de um \enf{while} em loop enquanto a cor do pai de
22781 + \enf{z} for vermelha. Este segundo caso é tratado no \enf{if} 109-113, onde há
22782 + uma rotação a esquerda, vamos aproveitar para analisarmos o que faz esta função.
22783 + \lstinputlisting [firstnumber=319, linerange={319-342}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
22784 + Esta função \enf{rb_rot_esq}, basicamente diminui a altura do lado direito e aumenta
22785 + do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
22786 + @@ -374,82 +393,52 @@ do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
22787 + \end{tabular}
22788 + \end{mytable}
22789 -
22790 -% \begin{table}[!ht]
22791 -% \scriptsize
22792 -% \begin{tabular}{|l|l|}
22793 -% \hline
22794 -% 327 & guarda referência do filho a direita \\
22795 -% \hline
22796 -% 328 & faz filho a direita igual a neto a esquerda (diminuindo a altura direita) \\
22797 -% \hline
22798 -% 329 & se filho a direita (antigo neto a esquerda) não é sentinela \\
22799 -% \hline
22800 -% 330 & atualiza pai de filho a direita (antigo neto a esquerda)\\
22801 -% \hline
22802 -% 331 & atualiza pai do nó movido\\
22803 -% \hline
22804 -% 332 & testa se nó anterior era raiz \\
22805 -% \hline
22806 -% 333 & atualiza nó raiz \\
22807 -% \hline
22808 -% 334 & se não é raiz testa se é nó a esquerda\\
22809 -% \hline
22810 -% 335 & atualiza como nó a esquerda\\
22811 -% \hline
22812 -% 336 & se nenhuma das opções anteriores\\
22813 -% \hline
22814 -% 337 & atualiza como nó a direita\\
22815 -% \hline
22816 -% 338 & atualiza filho a esquerda de y igual x, note que o nó x que era pai passa a ser filho\\
22817 -% \hline
22818 -% 339 & atualiza pai de x igual de y \\
22819 -% \hline
22820 -% 341 & retorna inteiro para uso em testes (não utilizado neste trabalho, mas a função tem esta interface)\\
22821 -% \hline
22822 -% \end{tabular}
22823 -% \end{table}

```

```

22824 -
22825 \noindent Um diagrama em nome da clareza, claro que aqui está representado apenas os nós
22826 que são representativos de uma rotação a esquerda e não árvores completas.\\
22827 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22828 level distance = .5cm}]
22829 \node [arn_n] {x}
22830 {
22831 child{ node [arn_n] {1}
22832 child { node [arn_x] {} }
22833 child { node [arn_x] {} }
22834 }
22835 child{ node [arn_r] {5}
22836 child { node [arn_x] {} }
22837 child { node [arn_x] {} }
22838 }
22839 child{ node [arn_r] {8}
22840 child { node [arn_x] {} }
22841 child { node [arn_x] {} }
22842 }
22843 }
22844 ;
22845 \end{tikzpicture}
22846 \colorbox{BurntOrange}{\scriptsize após \enfrb_rot_esq} }
22847 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22848 level distance = .5cm}]
22849 \node [arn_n] {y}
22850 {
22851 child{ node [arn_n] {x}
22852 child{ node [arn_r] {1}
22853 child { node [arn_x] {} }
22854 child { node [arn_x] {} }
22855 }
22856 child{ node [arn_r] {5}
22857 child { node [arn_x] {} }
22858 child { node [arn_x] {} }
22859 }
22860 child{ node [arn_r] {8}
22861 child { node [arn_x] {} }
22862 child { node [arn_x] {} }
22863 }
22864 }
22865 ;
22866 \end{tikzpicture} \\
22867 +
22868 \begin{center}
22869 \begin{minipage}{.95\textwidth}
22870 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22871 level distance = .5cm}]
22872 \node [arn_n] {x}
22873 {
22874 child{ node [arn_n] {1}
22875 child { node [arn_x] {} }
22876 child { node [arn_x] {} }
22877 }
22878 child{ node [arn_r] {5}
22879 child { node [arn_x] {} }
22880 child { node [arn_x] {} }
22881 }
22882 child{ node [arn_r] {8}
22883 child { node [arn_x] {} }
22884 child { node [arn_x] {} }
22885 }
22886 }
22887 ;
22888 \end{tikzpicture}
22889 \colorbox{BurntOrange}{\scriptsize após \enfrb_rot_esq} }
22890 \begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
22891 level distance = .5cm}]
22892 \node [arn_n] {y}
22893 {
22894 child{ node [arn_n] {x}
22895 child{ node [arn_r] {1}
22896 child { node [arn_x] {} }
22897 child { node [arn_x] {} }
22898 }
22899 child{ node [arn_r] {5}
22900 child { node [arn_x] {} }
22901 child { node [arn_x] {} }
22902 }
22903 child{ node [arn_r] {8}
22904 child { node [arn_x] {} }
22905 child { node [arn_x] {} }
22906 }
22907 }
22908 ;
22909 \end{tikzpicture} \\
22910 \end{center}
22911 Continuando a análise da função \enfrb_insert, do ponto que interrompemos para entendermos o \enfrb_rot_esq,
22912 após esta rotação a esquerda a árvore neste ponto passa para o \enfrb_rot_esq e a será feito uma rotação a direita
22913 \enfrb tem o pai e o avô colorido com preto e vermelho respectivamente e o avô rotacionado a direita.
22914 @ -548,12 +537,6 @@ tio a direita. O diagrama abaixo representa esta
22915 //
22916 //
22917 \begin{enumerate}
22918 - \item \enfrb será root
22919 - \item tio de \enfrb vermelho
22920 - \item tio de \enfrb preto em forma de triângulo
22921 - \item tio de \enfrb preto em forma de linha
22922 \end{enumerate}
22923 %%
22924 %%
22925 %% daqui pra frente eh antigo
22926
22927 commit 0253450d667065c4466451d776141bfeb5ea27c0
22928 Author: Joilnen <joilnen@gmail.com>
22929 Date: Mon Jan 17 13:26:33 2022 -0300
22930
22931 documentacao e modificacao do makefile nao precisamos de -lm
22932
22933 diff --git a/doc/tex/mybib.bib b/doc/tex/mybib.bib
22934 index 124a6bf..3dce8ba 100644
22935 --- a/doc/tex/mybib.bib
22936 +++ b/doc/tex/mybib.bib
22937 @@ -1,22 +1,21 @@
22938 @book{book1,
22939 title = {Computação Gráfica: Imagem},

```



```

22940 - author = {Jonas Gomes, Luiz Velho},
22941 - publisher = {SBM},
22942 - isbn = {85-244-0088-9},
22943 - year = {1994},
22944 + title = {Introduction to Algorithms},
22945 + author = {Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein},
22946 + publisher = {MIT},
22947 + isbn = {978-0-262-03384-8},
22948 + year = {2009},
22949 series = {},
22950 edition = {},
22951 volume = {},
22952 url = {}
22953 -
22954 @book{book2,
22955 - title = {A Computational Introduction to Digital Image Processing},
22956 - author = {Alasdair McAndrew},
22957 - publisher = {Taylor and Francis},
22958 - isbn = {978-1-4822-4735-0},
22959 - year = {2016},
22960 + title = {Notas de aula e Slides},
22961 + author = {Prof. Luciana Lee},
22962 + publisher = {},
22963 + isbn = {},
22964 + year = {},
22965 series = {},
22966 - edition = {Second},
22967 + edition = {},
22968 volume = {},
22969 url = {}
22970
22971 diff --git a/doc/tex/relatorio.bbl b/doc/tex/relatorio.bbl
22972 new file mode 100644
22973 index 0000000..a5d28bf
22974 --- /dev/null
22975 +++ b/doc/tex/relatorio.bbl
22976 @@ -0,0 +1,12 @@
22977 +\begin{thebibliography}{1}
22978 +
22979 +\bibitem{book2}
22980 +Prof.-Luciana Lee.
22981 +\newblock {\em Notas de aula e Slides}.
22982 +
22983 +\bibitem{book1}
22984 +Ronald L.-Rivest Thomas H.-Cormen, Charles E.-Leiserson and Clifford Stein.
22985 +\newblock {\em Introduction to Algorithms}.
22986 +\newblock MIT, 2009.
22987 +
22988 +\end{thebibliography}
22989 diff --git a/doc/tex/relatorio.blg b/doc/tex/relatorio.blg
22990 new file mode 100644
22991 index 0000000..ffb2c2b
22992 --- /dev/null
22993 +++ b/doc/tex/relatorio.blg
22994 @@ -0,0 +1,49 @@
22995 +This is BibTeX, Version 0.99d (TeX Live 2021/Arch Linux)
22996 +Capacity: max_strings=200000, hash_size=200000, hash_prime=170003
22997 +The top-level auxiliary file: relatorio.aux
22998 +The style file: plain.bst
22999 +Database file #1: mybib.bib
23000 +Warning--empty publisher in book2
23001 +Warning--empty year in book2
23002 +You've used 2 entries,
23003 +      2118 wiz_defined-function locations,
23004 +      505 strings with 4157 characters,
23005 +and the built_in function-call counts, 483 in all, are:
23006 += -- 38
23007 +> -- 20
23008 +< -- 0
23009 ++ -- 8
23010 +- -- 6
23011 +* -- 27
23012 +.: -- 88
23013 +add.period$ -- 5
23014 +call.type$ -- 2
23015 +change.case$ -- 7
23016 +chr.to.int$ -- 0
23017 +cite$ -- 4
23018 +duplicate$ -- 24
23019 +empty$ -- 45
23020 +format.name$ -- 6
23021 +if$ -- 97
23022 +int.to.chr$ -- 0
23023 +int.to.str$ -- 2
23024 +missing$ -- 4
23025 +newline$ -- 12
23026 +num.names$ -- 4
23027 +pop$ -- 20
23028 +preamble$ -- 1
23029 +purify$ -- 7
23030 +quote$ -- 0
23031 +skip$ -- 12
23032 +stack$ -- 0
23033 +substring$ -- 10
23034 +swap$ -- 2
23035 +text.length$ -- 0
23036 +text.prefix$ -- 0
23037 +top$ -- 0
23038 +type$ -- 4
23039 +warning$ -- 2
23040 +while$ -- 4
23041 +width$ -- 3
23042 +write$ -- 19
23043 +(There were 2 warnings)
23044 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
23045 index ea1f269..8f819fd 100644
23046 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
23047 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
23048 index 925b6ff..33b80d7 100644
23049 --- a/doc/tex/relatorio.tex
23050 +++ b/doc/tex/relatorio.tex
23051 @@ -11,6 +11,66 @@
23052 \usepackage{xcolor}
23053 \usepackage{booktabs}
23054 \usepackage[most]{tcolorbox}
23055 +\usepackage[edges]{forest}

```



```

23056 +\definecolor{folderbg}{RGB}{124,166,198}
23057 +\definecolor{folderborder}{RGB}{110,144,169}
23058 +\newlength\Size
23059 +\setlength\Size{4pt}
23060 +\tikzset{
23061 + folder/.pic={%
23062 + \filldraw [draw=folderborder, top color=folderbg!50, bottom color=folderbg] (-1.05*\Size,0.2*\Size+5pt) rectangle
+ (.75*\Size,-0.2*\Size-5pt);
23063 + \filldraw [draw=folderborder, top color=folderbg!50, bottom color=folderbg] (-1.15*\Size,-\Size) rectangle
+ (1.15*\Size,\Size);
23064 + },
23065 + file/.pic={%
23066 + \filldraw [draw=folderborder, top color=folderbg!5, bottom color=folderbg!10] (-\Size,.4*\Size+5pt) coordinate (a)
+ |- (\Size,-1.2*\Size) coordinate (b) -- ++(0,1.6*\Size) coordinate (c) -- ++(-5pt,5pt) coordinate (d) -- cycle
+ (d) |- (c) ;
23067 + },
23068 +}
23069 +\forestset{
23070 + declare autowrapped toks={pic me}{},
23071 + declare boolean register={pic root},
23072 + pic root=0,
23073 + pic dir tree/.style={%
23074 + for tree={%
23075 + folder,
23076 + font=\ttfamily,
23077 + grow'=0,
23078 + },
23079 + before typesetting nodes={%
23080 + for tree={%
23081 + edge label+/.option={pic me},
23082 + },
23083 + if pic root={
23084 + tikz+={
23085 + \pic at ([xshift=\Size].west) {folder};
23086 + },
23087 + align={l}
23088 + }{ },
23089 + },
23090 + },
23091 + pic me set/.code n args=2{%
23092 + \forestset{
23093 + #1/.style={%
23094 + inner xsep=2*\Size,
23095 + pic me={pic {#2}},
23096 + }
23097 + },
23098 + },
23099 + pic me set={directory}{folder},
23100 + pic me set={file}{file},
23101 +}
23102 +
23103 +\def\Size{4pt}
23104 +\tikzset{
23105 + folder/.pic={%
23106 + \filldraw[draw=folderborder,top color=folderbg!50,bottom color=folderbg]
+ (-1.05*\Size,0.2*\Size+5pt) rectangle ++(.75*\Size,-0.2*\Size-5pt);
23107 + \filldraw[draw=folderborder,top color=folderbg!50,bottom color=folderbg]
+ (-1.15*\Size,-\Size) rectangle (1.15*\Size,\Size);
23108 + },
23109 + file/.pic={%
23110 + \filldraw [draw=folderborder, top color=folderbg!5, bottom color=folderbg!10] (-\Size,.4*\Size+5pt) coordinate (a)
+ |- (\Size,-1.2*\Size) coordinate (b) -- ++(0,1.6*\Size) coordinate (c) -- ++(-5pt,5pt) coordinate (d) -- cycle
+ (d) |- (c) ;
23111 + },
23112 +}
23113 +
23114 +}
23115 +
23116 +\newcolorbox[blend into=tables]{mytable}[2][]{%
23117 + enhanced,
23118 + @0 -22,9 +82,9 @0
23119 + xshift=5mm,
23120 + yshift=-\tcboxeditheight/2,
23121 + yshifttext=-1mm},
23122 + boxed title style={colback=blue!50, sharp corners},
23123 + colframe = blue!50,
23124 + colback = blue!20,
23125 + boxed title style={colback=blue!30, sharp corners},
23126 + colframe = gray,
23127 + colback = blue!20, boxrule = 0.5pt,
23128 + % overlay = {\node[text=white, fill=red] at (frame.east)
23129 + % {\clubsuit$}};},
23130 + #1}
23131 + @0 -123,6 +183,32 @0 da atividade conjunto ordenado \
23132 + \small
23133 + \section {Introdução}
23134 + \noindent Esta biblioteca é composta pelo os seguintes arquivos, \
23135 + \begin{tcolorbox}[boxrule = 0.5pt]
23136 + \begin{center}
23137 + \scalebox{0.6}{
23138 + \begin{forest}
23139 + pic dir tree,
23140 + pic root,
23141 + for tree={% folder icons by default; override using file for file icons
23142 + directory,
23143 + },
23144 + [conjunto_ordenado
23145 + [doc
23146 + [relatorio.pdf, file]
23147 + [spec.txt, file]
23148 + ]
23149 + [src
23150 + ]
23151 + [tests
23152 + ]
23153 + [LEIAME, file]
23154 + [makefile, file]
23155 + ]
23156 + \end{forest}
23157 + }
23158 + \end{center}
23159 + \end{tcolorbox}
23160 + \noindent Em \enf{src} estão os arquivos especificos da biblioteca\
23161 + \begin{itemize}
23162 + \item [\$blacksquare\$] conjunto\_ordenado.h
23163 + \item [\$blacksquare\$] conjunto\_ordenado.c
23164 + @0 -131,7 +217,7 @0 da atividade conjunto ordenado \
23165 + \item [\$blacksquare\$] jcurses.h

```

```

23166 \end{itemize}
23167
23168 -\noindent Completamente separado do código da biblioteca temos alguns arquivos que implementam
23169 +\noindent Completamente separado do código da biblioteca, ainda em \enf{src} temos alguns arquivos que implementam
23170 testes, estes são:
23171 \begin{itemize}
23172 \item [\blacksquare] main.c
23173 @@ -141,6 +227,11 @@ testes, estes são:
23174 \item [\blacksquare] testa_rb.c
23175 \end{itemize}
23176
23177 +\noindent Arquivo de menu auxiliar nos testes:
23178 +\begin{itemize}
23179 +\item [\blacksquare] checklist.c
23180 +\end{itemize}
23181 +
23182 Todos os arquivos estão listados nos anexos na sua íntegra.
23183
23184 Como foi implementado um número grande de testes, estes foram separados em quatro
23185 @@ -282,8 +373,9 @@ que aponta para o filho esquerdo.
23186 \enf{while}, o primeiro \enf{if} decide se está sendo incluído pelo o lado esquerdo,
23187 linhas 99-106, temos o tratamento do \enf{caso 1}, nó pai vermelho, e nó tio vermelho,
23188 segue um diagrama deste caso em nome da clereza.\\
23189 -\begin{center}
23190 +\begin{tcolorbox}[boxrule = 0.5pt]
23191 \begin{minipage}{.95\textwidth}
23192 + \centering
23193 \begin{tikzpicture}[>=>stealth',level/.style={sibling distance = 2.5cm/#1,
23194 level distance = .5cm}]
23195 \node [arn_n] {}
23196 @@ -348,7 +440,8 @@ segue um diagrama deste caso em nome da clereza.\\
23197 ;
23198 \end{tikzpicture}
23199 \end{minipage}
23200 -\end{center}
23201 +\end{tcolorbox}
23202 +
23203 Note que saímos do \enf{caso 1} e caímos no \enf{caso 2}, nó pai do nó agora
23204 apontado por \enf{z}, vermelho e nó tio preto \enf{y}, em formato de triângulo,
23205 lembre-se que estamos dentro de um \enf{while} em loop enquanto a cor do pai de
23206 @@ -396,8 +489,9 @@ do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
23207 \noindent Um diagrama em nome da clereza, claro que aqui está representado apenas os nós
23208 que são representativos de uma rotação a esquerda e não árvores completas.\\
23209
23210 -\begin{center}
23211 +\begin{tcolorbox}[boxrule = 0.5pt, colframe = gray]
23212 \begin{minipage}{.95\textwidth}
23213 + \centering
23214 \begin{tikzpicture}[>=>stealth',level/.style={sibling distance = 2.5cm/#1,
23215 level distance = .5cm}]
23216 \node [arn_n] {x}
23217 @@ -438,7 +532,7 @@ que são representativos de uma rotação a esquerda e não árvores completas.\\
23218 ;
23219 \end{tikzpicture} \\
23220 \end{minipage}
23221 -\end{center}
23222 +\end{tcolorbox}
23223 Continuando a análise da função \enf{rb_insert}, do ponto que interrompemos para entendermos o \enf{rb_rot_esq},
23224 após esta rotação a esquerda a árvore neste ponto passa para o \enf{caso 3} e a será feita uma rotação a direita
23225 \enf{z} tem o pai e o avô colorido com preto e vermelho respectivamente e o avô rotacionado a direita.
23226 @@ -669,19 +763,34 @@ Variável rb_n com nome y \ref{14}
23227 % \lstinputlisting {code/del.c}
23228 % \lstinputlisting {code/utlis.c}
23229
23230 -\section{TAD}
23231 -\begin{figure}[H]
23232 - \begin{center}
23233 - \includegraphics [scale = .4] {figs/ADT}
23234 - \caption{Diagrama TAD}
23235 - \end{center}
23236 -\end{figure}
23237 -
23238 -\section{Referências Bibliográficas}
23239 +\section{}
23240 +\nocite{*}
23241 +\bibliography{mybib.bib}{*}
23242 +\bibliographystyle{plain}
23243 +\href{https://www.geeksforgeeks.org/abstract-data-types/}{Diagrama TAD - \color{blue} https://www.geeksforgeeks.org/
23244 abstract-data-types/}
23245 +
23246 +\section{Apêndices}
23247 +\appendix
23248 +\section{builder}
23249 +\lstinputlisting [caption = {makefile}] {../../src/makefile}
23250 +\section{fontes}
23251 +\setcounter{lstlisting}{0}
23252 +\lstinputlisting [caption = {checklist.h}] {../../src/checklist.h}
23253 +\lstinputlisting [caption = {checklist.c}] {../../src/checklist.c}
23254 +\lstinputlisting [caption = {conjunto_ordenado.h}] {../../src/conjunto_ordenado.h}
23255 +\lstinputlisting [caption = {conjunto_ordenado.c}] {../../src/conjunto_ordenado.c}
23256 +\lstinputlisting [caption = {jurses.h}] {../../src/jurses.h}
23257 +\lstinputlisting [caption = {main.c}] {../../src/main.c}
23258 +\lstinputlisting [caption = {red_black.h}] {../../src/red_black.h}
23259 +\lstinputlisting [caption = {red_black.c}] {../../src/red_black.c}
23260 +\lstinputlisting [caption = {testa.h}] {../../src/testa.h}
23261 +\lstinputlisting [caption = {testa_item_1.h}] {../../src/testa_item_1.h}
23262 +\lstinputlisting [caption = {testa_item_1.c}] {../../src/testa_item_1.c}
23263 +\lstinputlisting [caption = {testa_item_2.h}] {../../src/testa_item_2.h}
23264 +\lstinputlisting [caption = {testa_item_2.c}] {../../src/testa_item_2.c}
23265 +\lstinputlisting [caption = {testa_item_3.h}] {../../src/testa_item_3.h}
23266 +\lstinputlisting [caption = {testa_item_3.c}] {../../src/testa_item_3.c}
23267 +\lstinputlisting [caption = {testa_rb.h}] {../../src/testa_rb.h}
23268 +\lstinputlisting [caption = {testa_rb.c}] {../../src/testa_rb.c}
23269
23270 \end{document}
23271
23272 diff --git a/doc/tex/t.pdf b/doc/tex/t.pdf
23273 new file mode 100644
23274 index 0000000..c3c8c30
23275 Binary files /dev/null and b/doc/tex/t.pdf differ
23276 diff --git a/doc/tex/t.tex b/doc/tex/t.tex
23277 new file mode 100644
23278 index 0000000..42126b2
23279 --- /dev/null
23280 +++ b/doc/tex/t.tex
23281 @@ -0,0 +1,28 @@

```

```

23281 +\documentclass[border=10pt,multi,tikz]{standalone}
23282 +\begin{document}
23283 +
23284 +\begin{forest}
23285 +  pic dir tree,
23286 +  pic root,
23287 +  for tree={% folder icons by default; override using file for file icons
23288 +    directory,
23289 +  },
23290 +  [system
23291 +    [config
23292 +    [lib
23293 +      [Access
23294 +      ]
23295 +      [Plugin
23296 +      ]
23297 +      [file.txt, file
23298 +      ]
23299 +    ]
23300 +  ]
23301 +  [templates
23302 +  ]
23303 +  [tests
23304 +  ]
23305 + ]
23306 +\end{forest}
23307 +\end{document}
23308 +
23309 diff --git a/doc/tex/table.pdf b/doc/tex/table.pdf
23310 index f7e3592..75557d4 100644
23311 Binary files a/doc/tex/table.pdf and b/doc/tex/table.pdf differ
23312 diff --git a/doc/tex/table.tex b/doc/tex/table.tex
23313 index 4a270ee..ea8a13 100644
23314 --- a/doc/tex/table.tex
23315 +++ b/doc/tex/table.tex
23316 @@ -22,6 +22,12 @@
23317 \begin{document}
23318 \listoftables
23319
23320 +\noindent\hrulefill\par
23321 +\noindent\makebox[\textwidth][c]{%
23322 +  \begin{minipage}{1.5\textwidth}
23323 +    \hrulefill\par
23324 +  \end{minipage}}
23325 +
23326 \begin{table}
23327 \centering
23328 \begin{tabular}{|r|l|}
23329 diff --git a/src/makefile b/src/makefile
23330 index 064af43..2ddd9d1 100644
23331 --- a/src/makefile
23332 +++ b/src/makefile
23333 @@ -13,7 +13,7 @@ D=-g
23334 .c.o:
23335 $(C) -c $(D) $<
23336 all: $(O)
23337 - $(C) -o $(E) $(O) -lm
23338 + $(C) -o $(E) $(O)
23339 clean:
23340 rm -f *.o $(E)
23341
23342
23343 commit 18975f4aa53f3ba697d915f9027294f268d43ce8
23344 Author: Joilnen <joilnen@gmail.com>
23345 Date: Mon Jan 17 16:52:17 2022 -0300
23346
23347     adequando o código ao estilo escolhido e melhorando a documentação
23348     adequando o código ao estilo escolhido e documentando
23349
23350 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
23351 index 8f819fd..7299cab 100644
23352 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
23353 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
23354 index 33b80d7..fad64d1 100644
23355 --- a/doc/tex/relatorio.tex
23356 +++ b/doc/tex/relatorio.tex
23357 @@ -32,7 +32,7 @@
23358     pic dir tree/.style={%
23359     for tree={%
23360     folder,
23361     font=\ttfamily,
23362 +    font=\normalsize,
23363     grow'=0,
23364     },
23365     before typesetting nodes={%
23366 @@ -322,14 +322,104 @@ populamos com chave e o dado genérico, também setamos a cor preta, pois é ra
23367     ponteiros, pai, e filhos para nó sentinela \enfrb_s}, por fim retornamos o nó raiz criado.
23368
23369     Antes de passarmos para o \enfrb_insert}, vamos dar uma olha na estrutura nó \enfrb_n}
23370     -\lstinputlisting [firstnumber=33, linerange={33-48}, caption = {Fragmento do red_black.h}]{../src/red_black.h},
23371     +\lstinputlisting [firstnumber=33, linerange={33-49}, caption = {Fragmento do red_black.h}]{../src/red_black.h},
23372     trata-se da estrutura que abstrai um nó da árvore, comentário no código fonte descreve cada parâmetro.
23373
23374     Então a função \enfrb_insert} que insere um novo nó na árvore, detalharemos agora
23375     -o funcionamento desta função.
23376     +o funcionamento desta função, porém antes vejamos a função de rotação que é chamada na inserção,
23377     +uma rotação a esquerda, vamos aproveitar para analisarmos o que faz esta função.
23378     +\lstinputlisting [firstnumber=319, linerange={319-342}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
23379     +Esta função \enfrb_rot_esq}, basicamente diminui a altura do lado direito e aumenta
23380     +do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
23381     +
23382     +\begin{mytable}[colback=black!5]{rotação à esquerda}
23383     +  \scriptsize
23384     +\begin{tabular}{|l|l|}
23385     +  \hline
23386     +  327 & guarda referência do filho a direita \\
23387     +  \hline
23388     +  328 & faz filho a direita igual a neto a esquerda (diminuindo a altura direita) \\
23389     +  \hline
23390     +  329 & se filho a direita (antigo neto a esquerda) não é sentinela \\
23391     +  \hline
23392     +  330 & atualiza pai de filho a direita (antigo neto a esquerda)\\
23393     +  \hline
23394     +  331 & atualiza pai do nó movido\\
23395     +  \hline
23396     +  332 & testa se nó anterior era raiz \\

```

```

23397 + \hline
23398 + 333 & atualiza nó raiz \\
23399 + \hline
23400 + 334 & se não é raiz testa se é nó a esquerda\\
23401 + \hline
23402 + 335 & atualiza como nó a esquerda\\
23403 + \hline
23404 + 336 & se nenhuma das opções anteriores\\
23405 + \hline
23406 + 337 & atualiza como nó a direita\\
23407 + \hline
23408 + 338 & atualiza filho a esquerda de y igual x, note que o nó x que era pai passa a ser filho\\
23409 + \hline
23410 + 339 & atualiza pai de x igual de y \\
23411 + \hline
23412 + 341 & retorna inteiro para uso em testes (não utilizado neste trabalho, mas a função tem esta interface)\\
23413 + \hline
23414 + \end{tabular}
23415 + \end{mytable}
23416 +
23417 + \noindent Um diagrama em nome da clareza, claro que aqui está representado apenas os nós
23418 + que são representativos de uma rotação a esquerda e não árvores completas.\\
23419 +
23420 + \begin{tcolorbox}[boxrule = 0.5pt, colframe = gray]
23421 + \begin{minipage}{.95\textwidth}
23422 + \centering
23423 + \begin{tikzpicture}[->,>stealth',level/.style={sibling distance = 2.5cm/#1,
23424 + level distance = .5cm}]
23425 + \node [arn_n] {x}
23426 + \child{ node [arn_n] {1}
23427 + \child { node [arn_x] {} }
23428 + \child { node [arn_x] {} }
23429 + }
23430 + \child{ node [arn_n] {y}
23431 + \child{ node [arn_r] {5}
23432 + \child { node [arn_x] {} }
23433 + \child { node [arn_x] {} }
23434 + }
23435 + \child{ node [arn_r] {8}
23436 + \child { node [arn_x] {} }
23437 + \child { node [arn_x] {} }
23438 + }
23439 + }
23440 + ;
23441 + \end{tikzpicture}
23442 + \colorbox{BurntOrange}{\scriptsize após \enfrb_rot_esq} }
23443 + \begin{tikzpicture}[->,>stealth',level/.style={sibling distance = 2.5cm/#1,
23444 + level distance = .5cm}]
23445 + \node [arn_n] {y}
23446 + \child{ node [arn_n] {x}
23447 + \child{ node [arn_r] {1}
23448 + \child { node [arn_x] {} }
23449 + \child { node [arn_x] {} }
23450 + }
23451 + \child{ node [arn_r] {5}
23452 + \child { node [arn_x] {} }
23453 + \child { node [arn_x] {} }
23454 + }
23455 + }
23456 + \child{ node [arn_n] {8}
23457 + \child { node [arn_x] {} }
23458 + \child { node [arn_x] {} }
23459 + }
23460 + ;
23461 + \end{tikzpicture}
23462 + \end{minipage}
23463 + \end{tcolorbox}
23464 + \\
23465 + \\
23466 + \noindent Podemos agora entender melhor o funcionamento da função \enfrb_insert}.
23467 + \lstinputlisting [firstnumber=43, linerange={43-145}, caption = {Fragmento do red_black.c}]{../../src/red_black.c}
23468 +
23469 + -Consideremos estes 4 casos bases:
23470 + -Consideremos estes 4 casos bases, sendo \enf{z} o nó a ser incluído e \enf{t}, o nó raiz:
23471 + \begin{center}
23472 + \begin{minipage}{0.4\textwidth}
23473 + \begin{enumerate}
23474 + @0 -445,192 +535,126 @@ segue um diagrama deste caso em nome da clareza.\\
23475 + Note que saímos do \enf{caso 1} e caímos no \enf{caso 2}, nó pai do nó agora
23476 + apontado por \enf{z}, vermelho e nó tio preto \enf{y}, em formato de triângulo,
23477 + lembre-se que estamos dentro de um \enf{while} em loop enquanto a cor do pai de
23478 + \enf{z} for vermelha. Este segundo caso é tratado no \enf{if} 109-113, onde há
23479 + uma rotação a esquerda, vamos aproveitar para analisarmos o que faz esta função.
23480 + \lstinputlisting [firstnumber=319, linerange={319-342}, caption = {Fragmento do red_black.c}]{../../src/red_black.c}
23481 + Esta função \enfrb_rot_esq}, basicamente diminui a altura do lado direito e aumenta
23482 + do lado esquerdo reagrupando os nós, vejamos detalhadamente como.\\
23483 + -
23484 + -\begin{mytable}[colback=black!5]{rotação à esquerda}
23485 + -\scriptsize
23486 + -\begin{tabular}{|l|l|}
23487 + - \hline
23488 + - 327 & guarda referência do filho a direita \\
23489 + - \hline
23490 + - 328 & faz filho a direita igual a neto a esquerda (diminuindo a altura direita) \\
23491 + - \hline
23492 + - 329 & se filho a direita (antigo neto a esquerda) não é sentinela \\
23493 + - \hline
23494 + - 330 & atualiza pai de filho a direita (antigo neto a esquerda)\\
23495 + - \hline
23496 + - 331 & atualiza pai do nó movido\\
23497 + - \hline
23498 + - 332 & testa se nó anterior era raiz \\
23499 + - \hline
23500 + - 333 & atualiza nó raiz \\
23501 + - \hline
23502 + - 334 & se não é raiz testa se é nó a esquerda\\
23503 + - \hline
23504 + - 335 & atualiza como nó a esquerda\\
23505 + - \hline
23506 + - 336 & se nenhuma das opções anteriores\\
23507 + - \hline
23508 + - 337 & atualiza como nó a direita\\
23509 + - \hline
23510 + - 338 & atualiza filho a esquerda de y igual x, note que o nó x que era pai passa a ser filho\\
23511 + - \hline
23512 + - 339 & atualiza pai de x igual de y \\

```

```

23513 - \hline
23514 - 341 & retorna inteiro para uso em testes (não utilizado neste trabalho, mas a função tem esta interface)\\
23515 - \hline
23516 -\end{tabular}
23517 -\end{mytable}
23518 -
23519 -\noindent Um diagrama em nome da clareza, claro que aqui está representado apenas os nós
23520 -que são representativos de uma rotação a esquerda e não árvores completas.\\
23521 -
23522 -\begin{tcolorbox}[boxrule = 0.5pt, colframe = gray]
23523 +\en{z} for vermelha. Este segundo caso é tratado no \en{if} 109-113,
23524 +\begin{tcolorbox}[boxrule = 0.5pt]
23525 \begin{minipage}{.95\textwidth}
23526 \centering
23527 \begin{tikzpicture}[>=>stealth',level/.style={sibling distance = 2.5cm/#1,
23528 \level distance = .5cm}]
23529 - \node [arn_n] {x}
23530 - child{ node [arn_n] {1}
23531 - child { node [arn_x] {} }
23532 - child { node [arn_x] {} }
23533 - }
23534 - child{ node [arn_n] {y}
23535 - child{ node [arn_r] {5}
23536 + \node [arn_n] {}
23537 + child{ node [arn_r] {}
23538 + child{ node [arn_n] {}
23539 + child { node [arn_x] {} }
23540 + child { node [arn_x] {} }
23541 + }
23542 + child{ node [arn_r] {8}
23543 + child{ node [arn_r] {z}
23544 + child{ node [arn_n] {}
23545 + child{ node [arn_r] {}
23546 + child { node [arn_x] {} }
23547 + child { node [arn_x] {} }
23548 + }
23549 + child { node [arn_x] {} }
23550 + }
23551 + child{ node [arn_n] {}
23552 + child { node [arn_x] {} }
23553 + child { node [arn_x] {} }
23554 + }
23555 + }
23556 + }
23557 + child{ node [arn_n] {y}
23558 + child { node [arn_x] {} }
23559 + child{ node [arn_r] {}
23560 + child { node [arn_x] {} }
23561 + child { node [arn_x] {} }
23562 + }
23563 + }
23564 + }
23565 - \end{tikzpicture}
23566 - \colorbox{BurntOrange}{\scriptsize após \en{rb_rot_esq} }
23567 - \end{tikzpicture}
23568 + \colorbox{BurntOrange}{\scriptsize após o \en{if} nas linhas, 99-106 }
23569 \begin{tikzpicture}[>=>stealth',level/.style={sibling distance = 2.5cm/#1,
23570 \level distance = .5cm}]
23571 - \node [arn_n] {y}
23572 - child{ node [arn_n] {x}
23573 - child{ node [arn_r] {1}
23574 + \node [arn_n] {}
23575 + child{ node [arn_r] {}
23576 + child{ node [arn_n] {}
23577 + child { node [arn_x] {} }
23578 + child { node [arn_x] {} }
23579 + }
23580 - child{ node [arn_r] {5}
23581 - child { node [arn_x] {} }
23582 - child { node [arn_x] {} }
23583 + child{ node [arn_r] {z}
23584 + child{ node [arn_n] {}
23585 + child{ node [arn_r] {}
23586 + child { node [arn_x] {} }
23587 + child { node [arn_x] {} }
23588 + }
23589 + child { node [arn_x] {} }
23590 + }
23591 + child{ node [arn_n] {}
23592 + child { node [arn_x] {} }
23593 + child { node [arn_x] {} }
23594 + }
23595 + }
23596 + }
23597 - child{ node [arn_n] {8}
23598 - child { node [arn_x] {} }
23599 + child{ node [arn_n] {y}
23600 + child { node [arn_x] {} }
23601 + child{ node [arn_r] {}
23602 + child { node [arn_x] {} }
23603 + child { node [arn_x] {} }
23604 + }
23605 + }
23606 - \end{tikzpicture} \\
23607 - \end{tikzpicture}
23608 + \end{tikzpicture}
23609 \end{minipage}
23610 \end{tcolorbox}
23611 -Continuando a análise da função \en{rb_insert}, do ponto que interrompemos para entendermos o \en{rb_rot_esq},
23612 -após esta rotação a esquerda a árvore neste ponto passa para o \en{caso 3} e a será feito uma rotação a direita
23613 -\en{z} tem o pai e o avô colorido com preto e vermelho respectivamente e o avô rotacionado a direita.
23614 -Repetindo aqui o fragmento do código anterior \en{rb_insert}, trata o \en{caso 3} linhas 114-116.
23615 -\lstinputlisting [firstnumber=97, linerange={97-118}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
23616 \begin{mytable}[colback=black!5]{caso 3}
23617 +\begin{mytable}[colback=black!5]{caso 2}
23618 \scriptsize
23619 \begin{tabular}{|l|l|}
23620 \hline
23621 - 114 & colore pai de \en{z} com preto \\
23622 + 110 & checa se \en{z} é nó a direita \\
23623 \hline
23624 - 115 & colore avô de \en{z} com vermelho \\
23625 + 111 & colore avô de \en{z} com vermelho \\
23626 \hline
23627 - 116 & rotaciona avô à direita \\
23628 \hline

```

```

23629 \end{tabular}
23630 \end{mytable}
23631 -Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
23632 -é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
23633 -tio a direita. O diagrama abaixo representa esta
23634 -Detalhando, variável \enf{x} é uma variável que itera sobre os nós da árvore ela começa sendo igual
23635 -ao nó raiz, \enf{y} é uma variável que aponta para pai, pois a lógica dessa rotina é ir rastreando
23636 -nó e seu pai, como a condição inicial é o nó raiz, a \enf{x} aponta para raiz e a \enf{y} aponta
23637 -para o nó sentinela, consideremos os casos possíveis, caso um, o nó raiz não existe ou seja é
23638 -igual a sentinela e estamos incluindo o primeiro nó que se tornará o raiz, neste caso
23639 -o primeiro \enf{while} não é executado pois a \enf{x} apontará para \enf{rb_s} (sentinela),
23640 -como \enf{y} rastreia o pai, ela aponta também para sentinela, e fazemos o nó incluído apontar
23641 -o ponteiro pai para o sentinela através \enf{z->p = y}, caso neste ponto a \enf{y} continua
23642 -apontando para sentinela significa que temos apenas um nó sendo tratado, logo fazemos este nó
23643 -raiz na linha \enf{*t = z}, posteriormente fazemos ponteiros pai e filhos apontarem para sentinela
23644 -e colorimos de vermelho, neste ponto você pode perguntar, mas se ele é o raiz não deveria ser
23645 -preto, verdade, é que essa linha é feita com vistas aos casos mais comuns que são a inserção
23646 -de nós não raiz, que é tratado no \enf{while} seguinte, a cor será alterado para preto, no
23647 -final da função pelo o ponteiro \enf{t} na linha \enf{(*t)->c = BLACK}, ou seja ao final
23648 -da função ela sempre pinta o nó raiz de preto.
23649 -
23650 -Caso dois, neste o nó inserido não é o primeiro, o primeiro \enf{while} itera pelo nó raiz,
23651 -através dos ponteiros para o(s) filho(s) a direita ou a esquerda de acordo com a comparação
23652 -das chaves, dos nós visitados e do nó sendo incluído, note que neste ponto, é feito
23653 -a crítica se as chaves são iguais, como a especificação proíbe chaves iguais, ele retorna
23654 -código de erro após o \enf{else} ou finaliza quando a \enf{x} apontar para o sentinela
23655 -que neste caso representa um nó folha, a \enf{y} apontará para o pai deste nó, estno
23656 -fora é o ponto que será incluído o novo nó, na linha \enf{z->p = y}, enquanto o primeiro
23657 -\enf{if} setou as variáveis \enf{x} e \enf{y}, o segundo \enf{if}, inclui o novo nó
23658 -\enf{z} no lado esquerdo ou direito de \enf{y}, aponta os filhos desde nó para sentinela
23659 -e colore com vermelho, que é a ação mais comum.
23660 -
23661 -Todas as inclusões entram numa etapa, o segndo \enf{while}, que chamei de \enf{ajuste}
23662 -na bibliografia é referiad como \enf{fixup}, esta parte do código raz o ajuste de coloração
23663 -e rotaciona nós para manter as propriedades da árvore red-black obedecidas.
23664 -O primeiro \enf{if} dentro deste segundo \enf{while}.
23665 -
23666 -Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
23667 -é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
23668 -tio a direita. O diagrama abaixo representa esta
23669 -
23670 -%
23671 -%
23672 -%
23673 -\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
23674 - level distance = 1cm}]
23675 -\node [arn_r] {}
23676 - child{ node [arn_n] {}
23677 - child{ node [arn_r] {z}
23678 - child{ node [arn_x] {} }
23679 - child{ node [arn_x] {} }
23680 - }
23681 - child{ node [arn_x] {} }
23682 - }
23683 - child{ node [arn_n] {}
23684 - child { node [arn_x] {} } }
23685 - child { node [arn_x] {} } }
23686 - }
23687 -;
23688 -\end{tikzpicture}
23689 -\colorbox{BurntOrange}{após o \enf{else} nas linhas, 107-117 }
23690 -\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
23691 - level distance = 1cm}]
23692 -\node [arn_n] {}
23693 - child{ node [arn_n] {}
23694 - child{ node [arn_r] {z}
23695 - child{ node [arn_x] {} }
23696 - child{ node [arn_x] {} }
23697 - }
23698 - child{ node [arn_x] {} }
23699 - }
23700 - child{ node [arn_n] {}
23701 - child { node [arn_x] {} } }
23702 - child { node [arn_x] {} } }
23703 - }
23704 -;
23705 -\end{tikzpicture}
23706 -\%
23707 -\%
23708 -\%
23709 -\%
23710 -\%
23711 -\%
23712 +% Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
23713 +% é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
23714 +% tio a direita. O diagrama abaixo representa esta
23715 +% Detalhando, variável \enf{x} é uma variável que itera sobre os nós da árvore ela começa sendo igual
23716 +% ao nó raiz, \enf{y} é uma variável que aponta para pai, pois a lógica dessa rotina é ir rastreando
23717 +% nó e seu pai, como a condição inicial é o nó raiz, a \enf{x} aponta para raiz e a \enf{y} aponta
23718 +% para o nó sentinela, consideremos os casos possíveis, caso um, o nó raiz não existe ou seja é
23719 +% igual a sentinela e estamos incluindo o primeiro nó que se tornará o raiz, neste caso
23720 +% o primeiro \enf{while} não é executado pois a \enf{x} apontará para \enf{rb_s} (sentinela),
23721 +% como \enf{y} rastreia o pai, ela aponta também para sentinela, e fazemos o nó incluído apontar
23722 +% o ponteiro pai para o sentinela através \enf{z->p = y}, caso neste ponto a \enf{y} continua
23723 +% apontando para sentinela significa que temos apenas um nó sendo tratado, logo fazemos este nó
23724 +% raiz na linha \enf{*t = z}, posteriormente fazemos ponteiros pai e filhos apontarem para sentinela
23725 +% e colorimos de vermelho, neste ponto você pode perguntar, mas se ele é o raiz não deveria ser
23726 +% preto, verdade, é que essa linha é feita com vistas aos casos mais comuns que são a inserção
23727 +% de nós não raiz, que é tratado no \enf{while} seguinte, a cor será alterado para preto, no
23728 +% final da função pelo o ponteiro \enf{t} na linha \enf{(*t)->c = BLACK}, ou seja ao final
23729 +% da função ela sempre pinta o nó raiz de preto.
23730 +%
23731 +% Caso dois, neste o nó inserido não é o primeiro, o primeiro \enf{while} itera pelo nó raiz,
23732 +% através dos ponteiros para o(s) filho(s) a direita ou a esquerda de acordo com a comparação
23733 +% das chaves, dos nós visitados e do nó sendo incluído, note que neste ponto, é feito
23734 +% a crítica se as chaves são iguais, como a especificação proíbe chaves iguais, ele retorna
23735 +% código de erro após o \enf{else} ou finaliza quando a \enf{x} apontar para o sentinela
23736 +% que neste caso representa um nó folha, a \enf{y} apontará para o pai deste nó, estno
23737 +% fora é o ponto que será incluído o novo nó, na linha \enf{z->p = y}, enquanto o primeiro
23738 +% \enf{if} setou as variáveis \enf{x} e \enf{y}, o segundo \enf{if}, inclui o novo nó
23739 +% \enf{z} no lado esquerdo ou direito de \enf{y}, aponta os filhos desde nó para sentinela
23740 +% e colore com vermelho, que é a ação mais comum.
23741 +%
23742 +% Todas as inclusões entram numa etapa, o segndo \enf{while}, que chamei de \enf{ajuste}
23743 +% na bibliografia é referiad como \enf{fixup}, esta parte do código raz o ajuste de coloração
23744 +% e rotaciona nós para manter as propriedades da árvore red-black obedecidas.

```

```

23745 +% 0 primeiro \enf{if} dentro deste segundo \enf{while}.
23746 +%
23747 +% Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
23748 +% é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
23749 +% tio a direita. O diagrama abaixo representa esta
23750 +%
23751 +%
23752 +%
23753 +% daqui pra frente eh antigo
23754 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
23755 index 79e54c4..703f958 100644
23756 --- a/src/conjunto_ordenado.c
23757 +++ b/src/conjunto_ordenado.c
23758 @@ -24,6 +24,7 @@ conjord *conjord_cria(char *nome)
23759
23760 /**
23761  * cria conjunto ordenado e retorna o ponteiro para ele
23762  * @param a conjunto ordenando
23763  */
23764 void conjord_destroi(conjord *c)
23765 {
23766 diff --git a/src/red_black.h b/src/red_black.h
23767 index 11f2828..a4143c7 100644
23768 --- a/src/red_black.h
23769 +++ b/src/red_black.h
23770 @@ -40,7 +40,8 @@ typedef enum tipo_screen {TMONO, TCOLOR} tipo_screen;
23771  * @param r right, ponteiro para filho a direita
23772  * @param p parent, ponteiro para noh pai
23773  */
23774 -typedef struct rb_n {
23775 +typedef struct rb_n
23776 +{
23777     int k;
23778     unsigned char c:1;
23779     void *d;
23780 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
23781 index 8d49570..257f24c 100644
23782 --- a/src/testa_item_1.c
23783 +++ b/src/testa_item_1.c
23784 @@ -11,7 +11,8 @@ static void imprime_meu_tipo(void *);
23785  * @param nome string em c
23786  * @param telefone string em c
23787  */
23788 -typedef struct {
23789 +typedef struct
23790 +{
23791     char *nome;
23792     char *telefone;
23793 } meu_tipo;
23794 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
23795 index af21392..fd73b4 100644
23796 --- a/src/testa_item_2.c
23797 +++ b/src/testa_item_2.c
23798 @@ -10,7 +10,8 @@
23799 static void imprime_meu_tipo(void *);
23800
23801 -typedef struct {
23802 +typedef struct
23803 +{
23804     char *nome;
23805     int milhas;
23806 } meu_tipo2;
23807 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
23808 index c6d0004..a648ccf 100644
23809 --- a/src/testa_item_3.c
23810 +++ b/src/testa_item_3.c
23811 @@ -12,7 +12,8 @@
23812 static void imprime_meu_tipo(void *);
23813
23814 -typedef struct {
23815 +typedef struct
23816 +{
23817     char letra;
23818 } meu_tipo3;
23819
23820
23821
23822
23823 commit 1a7df6515af6c8ba566c83666842c4e30ff5c11c4
23824 Author: Joilnen <joilnen@gmail.com>
23825 Date: Tue Jan 18 01:59:53 2022 -0300
23826
23827     atualizando o nome do autor, e documentando
23828
23829 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
23830 index 7299cab..57c96a7 100644
23831 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
23832 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
23833 index fad64d1..89f5221 100644
23834 --- a/doc/tex/relatorio.tex
23835 +++ b/doc/tex/relatorio.tex
23836 @@ -2,11 +2,6 @@
23837
23838 \input{packages}
23839 \usepackage{fancyvrb}
23840 -% \usepackage{fancyhdr}
23841 -% \pagestyle{fancy}
23842 -
23843 -% \fancyhead{}
23844 -% \fancyfoot{}
23845 \usepackage{listingsutf8}
23846 \usepackage{xcolor}
23847 \usepackage{booktabs}
23848 @@ -90,7 +85,7 @@
23849 #1}
23850
23851 % \definecolor{codegreen}{rgb}{0,0.6,0}
23852 -\definecolor{codegreen}{rgb}{.6,.3,0}
23853 +\definecolor{codebrown}{rgb}{.6,.3,0}
23854 \definecolor{codegray}{rgb}{0.5,0.5,0.5}
23855 \definecolor{codepurple}{rgb}{0.58,0,0.82}
23856 \definecolor{backcolour}{rgb}{0.95,0.95,0.92}
23857 @@ -100,7 +95,7 @@
23858
23859 \lstdefinestyle{mystyle}{
23860     backgroundcolor=\color{backcolour},

```



```

23861 - commentstyle=\color{codegreen},
23862 + commentstyle=\color{codebrown},
23863 keywordstyle=\color{magenta},
23864 numberstyle=\tiny\color{codegray},
23865 stringstyle=\color{codepurple},
23866 @@ -205,6 +200,27 @@ da atividade conjunto ordenado \\
23867 [makefile, file]
23868 ]
23869 \end{forest}
23870 +\begin{forest}
23871 + pic dir tree,
23872 + pic root,
23873 + for tree={% folder icons by default; override using file for file icons
23874 + directory,
23875 + },
23876 + [src
23877 + [checklist.h, file]
23878 + [checklist.c, file]
23879 + [conjunto_ordenado.h, file]
23880 + [conjunto_ordenado.c, file]
23881 + [red_black.h, file]
23882 + [red_black.c, file]
23883 + [jurses.h, file]
23884 + [main.c, file]
23885 + [testa_item_1.c, file]
23886 + [testa_item_2.c, file]
23887 + [testa_item_3.c, file]
23888 + [testa_rb.c, file]
23889 + ]
23890 +\end{forest}
23891 }
23892 \end{center}
23893 \end{tcolorbox}
23894 @@ -570,28 +586,28 @@ lembre-se que estamos dentro de um \enf{while} em loop enquanto a cor do pai de
23895 }
23896 ;
23897 \end{tikzpicture}
23898 - \colorbox{BurntOrange}{\scriptsize após o \enf{if} nas linhas, 99-106 }
23899 + \colorbox{BurntOrange}{\scriptsize após o \enf{if} nas linhas, 110-113}
23900 \begin{tikzpicture}[>=stealth',level/.style={sibling distance = 2.5cm/#1,
23901 level distance = .5cm}]
23902 \node [arn_n] {}
23903 child{ node [arn_r] {}
23904 - child{ node [arn_n] {}
23905 - child { node [arn_x] {} } }
23906 - child { node [arn_x] {} } }
23907 }
23908 child{ node [arn_r] {z}
23909 child{ node [arn_n] {}
23910 child{ node [arn_r] {}
23911 + child { node [arn_x] {} } }
23912 + child { node [arn_x] {} } }
23913 }
23914 + child{ node [arn_n] {}
23915 + child{ node [arn_r] {}
23916 + child { node [arn_x] {} } }
23917 + child { node [arn_x] {} } }
23918 }
23919 child { node [arn_x] {} }
23920 }
23921 - child{ node [arn_n] {}
23922 + }
23923 + child{ node [arn_n] {}
23924 + child { node [arn_x] {} } }
23925 + child { node [arn_x] {} } }
23926 }
23927 - }
23928 }
23929 child{ node [arn_n] {y}
23930 child { node [arn_x] {} } }
23931 @@ -610,56 +626,83 @@ lembre-se que estamos dentro de um \enf{while} em loop enquanto a cor do pai de
23932 \hline
23933 110 & checa se \enf{z} é nó a direita \\
23934 \hline
23935 - 111 & colore avô de \enf{z} com vermelho \\
23936 + 111 & faz \enf{z} apontar para nó pai \\
23937 \hline
23938 - 116 & rotaciona avô à direita \\
23939 + 112 & rotaciona nó a esquerda troca de lugar com o nó raiz dessa subárvore \\
23940 \hline
23941 \end{tabular}
23942 \end{mytable}
23943 -% Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
23944 -% é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
23945 -% tio a direita. O diagrama abaixo representa esta
23946 -% Detalhando, variável \enf{x} é uma variável que itera sobre os nós da árvore ela começa sendo igual
23947 -% ao nó raiz, \enf{y} é uma variável que aponta para pai, pois a lógica dessa rotina é ir rastrando
23948 -% nó e seu pai, como a condição inicial é o nó raiz, a \enf{x} aponta para raiz e a \enf{y} aponta
23949 -% para o nó sentinela, consideremos os casos possíveis, caso um, o nó raiz não existe ou seja é
23950 -% igual a sentinela e estamos incluindo o primeiro nó que se tornará o raiz, neste caso
23951 -% o primeiro \enf{while} não é executado pois a \enf{x} apontará para \enf{rb.s} (sentinela),
23952 -% como \enf{y} rastreia o pai, ela aponta também para sentinela, e fazemos o nó incluído apontar
23953 -% o ponteiro pai para o sentinela através \enf{z->p = y}, caso neste ponto a \enf{y} continua
23954 -% apontando para sentinela significa que temos apenas um nó sendo tratado, logo fazemos este nó
23955 -% raiz na linha \enf{t = z}, posteriormente fazemos ponteiros pai e filhos apontarem para sentinela
23956 -% e colorimos de vermelho, neste ponto você pode perguntar, mas se ele é o raiz não deveria ser
23957 -% preto, verdade, é que essa linha é feita com vistas aos casos mais comuns que são a inserção
23958 -% de nós não raiz, que é tratado no \enf{while} seguinte, a cor será alterado para preto, no
23959 -% final da função pelo o ponteiro \enf{t} na linha \enf{(t)->c = BLACK}, ou seja ao final
23960 -% da função ela sempre pinta o nó raiz de preto.
23961 -%
23962 -% Caso dois, neste o nó inserido não é o primeiro, o primeiro \enf{while} itera pelo nó raiz,
23963 -% através dos ponteiros para o(s) filho(s) a direita ou a esquerda de acordo com a comparação
23964 -% das chaves, dos nós visitados e do nó sendo incluído, note que neste ponto, é feito
23965 -% a crítica se as chaves são iguais, como a especificação proíbe chaves iguais, ele retorna
23966 -% código de erro após o \enf{else} ou finaliza quando a \enf{x} apontar para o sentinela
23967 -% que neste caso representa um nó folha, a \enf{y} apontará para o pai deste nó, então
23968 -% fora é o ponto que será incluído o novo nó, na linha \enf{z->p = y}, enquanto o primeiro
23969 -% \enf{if} setou as variáveis \enf{x} e \enf{y}, o segundo \enf{if}, inclui o novo nó
23970 -% \enf{z} no lado esquerdo ou direito de \enf{y}, aponta os filhos desde nó para sentinela
23971 -% e colore com vermelho, que é a ação mais comum.
23972 -%
23973 -% Todas as inclusões entram numa etapa, o segundo \enf{while}, que chamei de \enf{ajuste}
23974 -% na bibliografia é referiad como \enf{fixup}, esta parte do código faz o ajuste de coloração
23975 -% e rotaciona nós para manter as propriedades da árvore red-black obedecidas.
23976 -% O primeiro \enf{if} dentro deste segundo \enf{while}.

```



```

23977 -%
23978 -% Aqui temos a parte mais importante da inclusão, se o nó pai do nó incluído \enf{z}
23979 -% é nó a esquerda, ou seja filho a esquerda do avô, fazemos \enf{y} apontar para o nó
23980 -% tio a direita. O diagrama abaixo representa esta
23981 -%
23982 -%%
23983 -%%
23984 -%% daqui pra frente eh antigo
23985 -%%
23986 -%%
23987 +\ \ \
23988 +\ \ \
23989 +Então chega-se ao \enf{caso 3}, \enf{z} em linha com o nó pai, ambos vermelhos e nó tio preto:
23990 +\begin{tcolorbox}[boxrule = 0.5pt]
23991 +\begin{minipage}{.95\textwidth}
23992 +\centering
23993 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
23994 +\level distance = .5cm}]
23995 +\node [arn_n] {}
23996 +\child{ node [arn_r] {}
23997 +\child{ node [arn_r] {z}
23998 +\child{ node [arn_n] {}
23999 +\child { node [arn_x] {} }
24000 +\child { node [arn_x] {} }
24001 +}
24002 +\child{ node [arn_n] {}
24003 +\child{ node [arn_r] {}
24004 +\child { node [arn_x] {} }
24005 +\child { node [arn_x] {} }
24006 +}
24007 +\child { node [arn_x] {} }
24008 +}
24009 +}
24010 +\child{ node [arn_n] {}
24011 +\child { node [arn_x] {} }
24012 +\child { node [arn_x] {} }
24013 +}
24014 +}
24015 +\child{ node [arn_n] {y}
24016 +\child { node [arn_x] {} }
24017 +\child{ node [arn_r] {}
24018 +\child { node [arn_x] {} }
24019 +\child { node [arn_x] {} }
24020 +}
24021 +}
24022 +}
24023 +\end{tikzpicture}
24024 +\colorbox{BurntOrange}{\scriptsize após as linhas 114-116}
24025 +\begin{tikzpicture}[>,>stealth',level/.style={sibling distance = 2.5cm/#1,
24026 +\level distance = .5cm}]
24027 +\node [arn_n] {}
24028 +\child{ node [arn_r] {z}
24029 +\child{ node [arn_n] {}
24030 +\child { node [arn_x] {} }
24031 +\child { node [arn_x] {} }
24032 +}
24033 +\child{ node [arn_n] {}
24034 +\child{ node [arn_r] {}
24035 +\child { node [arn_x] {} }
24036 +\child { node [arn_x] {} }
24037 +}
24038 +\child { node [arn_x] {} }
24039 +}
24040 +}
24041 +\child{ node [arn_r] {}
24042 +\child{ node [arn_n] {}
24043 +\child { node [arn_x] {} }
24044 +\child { node [arn_x] {} }
24045 +}
24046 +\child{ node [arn_n] {y}
24047 +\child { node [arn_x] {} }
24048 +\child{ node [arn_r] {}
24049 +\child { node [arn_x] {} }
24050 +\child { node [arn_x] {} }
24051 +}
24052 +}
24053 +}
24054 +}
24055 +\end{tikzpicture}
24056 +\end{minipage}
24057 +\end{tcolorbox}
24058 +\enf{conjunto_ordenado.h} é o arquivo cabeçalho que deverá ser incluído no código
24059 +fonte do cliente, \enf{conjunto_ordenado.c} onde estão implementados as funções,
24060 +\enf{red_black.h} e \enf{red_black.c}, cabeçalho e implementações da estrutura
24061 +@@ -718,75 +761,6 @@ referenciada no ato da criação do nó raiz no código cliente, logo no exemplo
24062 +temos uma árvore nomeada \enf{conjunto_a}, enquanto nós filhos nessa árvore é adicionado
24063 +com \enf{insere_elemento}
24064 +\ \ \
24065 -
24066 -\noindent Estrutura principal
24067 -\begin{lstlisting}[caption={Nó Red Black}]
24068 -/**
24069 - * composicao do noh da RB
24070 - * @rb_n noh red black
24071 - * @k chave (k de key)
24072 - * @c cor (c de color ou cor, 0-black, 1-red)
24073 - * @d dado (de de data ou dados)
24074 - * @l @r @p l-left filho a esquerda, r-right filho a direita e p-parent, noh pai
24075 - */
24076 -typedef struct rb_n {
24077 - int k;
24078 - char c;1;
24079 - void *d;
24080 - struct rb_n *l, *r, *p;
24081 -} rb_n;
24082 -\end{lstlisting}
24083 -Significado dos componentes da estrutura \enf{rb_n}
24084 -\begin{itemize}
24085 - \item \enf{k} - chave
24086 - \item \enf{c} - cor (0 - preto e 1 - vermelho, apenas o primeiro bit utilizado)
24087 - \item \enf{d} - dado generico
24088 - \item \enf{l, r, p} - ponteiro para filho a esquerda, filho a direita e nó pai
24089 -\end{itemize}
24090 -Funções que manipulam os nós na Árvore
24091 -\begin{lstlisting}[language=c, caption={Funções da biblioteca}]
24092 -/**

```

```

24093 - * cria o primeiro noh, o raiz da arvore rb
24094 - * serah o primeiro elemento criado na arvore
24095 - * e o primeiro do conjunto ordenado
24096 - */
24097 -#define CRIA_CONJUNTO_ORDENADO(a, key, dado) \
24098 - rb_n a; \
24099 - a.k = key; \
24100 - a.c = 0; \
24101 - a.d = &dado; \
24102 - a.p = a.l = a.r = NULL;
24103 -int imprime_conjunto(rb_n *a);
24104 -unsigned int cardinalidade(rb_n *a);
24105 -int ta_vazio(rb_n *a);
24106 -int limpa_conjunto(rb_n *a);
24107 -
24108 -int insere_elemento(rb_n *a, int k, void *e);
24109 -int exclui_elemento(rb_n *a, int k);
24110 -int verifica_pertencimento(rb_n *a, int k);
24111 -int imprime_arvore(rb_n *a, short h);
24112 -
24113 -rb_n* intersecta_conjuntos(rb_n *a, rb_n *b);
24114 -rb_n* une_conjuntos(rb_n *a, rb_n *b);
24115 -rb_n* subtrai_conjuntos(rb_n *a, rb_n *b);
24116 -
24117 -int destroi_conjunto(rb_n *a);
24118 -
24119 -int troca_cor(rb_n *a);
24120 -int rotaciona_esquerda(rb_n *r, rb_n *w);
24121 -int rotaciona_direita(rb_n *r, rb_n *w);
24122 -
24123 -void (*imprime_elemento)(rb_n *);
24124 -int seta_imprime_elemento(void (*i)(rb_n *n));
24125 -
24126 -
24127 -\end{lstlisting}
24128 -
24129 -\lstinputlisting [linerange={10-15}, firstnumber=10]{../src/testa_item_1.c}
24130 -Variavel rb\_n com nome y \ref{14}
24131 -% \lstinputlisting {code/del.c}
24132 -% \lstinputlisting {code/utlis.c}
24133 -
24134 \section{}
24135 \nocite{*}
24136 \bibliography{mybib.bib}{*}
24137 diff --git a/src/checklist.c b/src/checklist.c
24138 index c61a1b..bf32a6b 100644
24139 --- a/src/checklist.c
24140 +++ b/src/checklist.c
24141 @@ -1,3 +1,6 @@
24142 /**
24143  * aluno: joilnen leite
24144  */
24145 #include <stdio.h>
24146 #include <string.h>
24147 #include "conjunto_ordenado.h"
24148 diff --git a/src/checklist.h b/src/checklist.h
24149 index aaf3a24..eff8fe0 100644
24150 --- a/src/checklist.h
24151 +++ b/src/checklist.h
24152 @@ -1,3 +1,6 @@
24153 /**
24154  * aluno: joilnen leite
24155  */
24156 #ifndef CHECKLIST_H_
24157 #define CHECKLIST_H_
24158
24159 diff --git a/src/conjunto_ordenado.c b/src/conjunto_ordenado.c
24160 index 703f958..df17496 100644
24161 --- a/src/conjunto_ordenado.c
24162 +++ b/src/conjunto_ordenado.c
24163 @@ -1,3 +1,6 @@
24164 /**
24165  * aluno: joilnen leite
24166  */
24167 #include <stdio.h>
24168 #include <string.h>
24169 #include <math.h>
24170 diff --git a/src/conjunto_ordenado.h b/src/conjunto_ordenado.h
24171 index ca7663e..814d345 100644
24172 --- a/src/conjunto_ordenado.h
24173 +++ b/src/conjunto_ordenado.h
24174 @@ -1,9 +1,8 @@
24175 #ifndef CONJUNTO_ORDENADO_H
24176 #define CONJUNTO_ORDENADO_H
24177 -
24178 /*
24179  * aluno: joilnen leite
24180  */
24181 #ifndef CONJUNTO_ORDENADO_H
24182 #define CONJUNTO_ORDENADO_H
24183
24184 #include "jurses.h"
24185 #include "red_black.h"
24186 diff --git a/src/red_black.h b/src/red_black.h
24187 index a4143c7..2f44c53 100644
24188 --- a/src/red_black.h
24189 +++ b/src/red_black.h
24190 @@ -1,9 +1,9 @@
24191 #ifndef REDBLACK_H
24192 #define REDBLACK_H
24193 -
24194 /*
24195  * aluno: joilnen leite
24196  */
24197 #ifndef REDBLACK_H
24198 #define REDBLACK_H
24199 +
24200 #include <stdlib.h>
24201
24202 /**
24203 diff --git a/src/testa.h b/src/testa.h
24204 index dcff849..8c63171 100644
24205 --- a/src/testa.h
24206 +++ b/src/testa.h
24207 @@ -1,9 +1,8 @@
24208 #ifndef TESTA_H_

```

```

24209 -#define TESTA_H_
24210 -
24211 /*
24212  * aluno: joilnen leite
24213  */
24214 #ifndef TESTA_H_
24215 #define TESTA_H_
24216
24217 #include "jcourses.h"
24218 #include "testa_item_1.h"
24219 diff --git a/src/testa_item_1.c b/src/testa_item_1.c
24220 index 257f24c..90b33c2 100644
24221 --- a/src/testa_item_1.c
24222 +++ b/src/testa_item_1.c
24223 @@ -1,3 +1,7 @@
24224 +/*
24225 + * aluno: joilnen leite
24226 + */
24227 +
24228 #include <stdio.h>
24229 #include <string.h>
24230 #include <time.h>
24231 diff --git a/src/testa_item_1.h b/src/testa_item_1.h
24232 index d22ff07..96af165 100644
24233 --- a/src/testa_item_1.h
24234 +++ b/src/testa_item_1.h
24235 @@ -1,3 +1,7 @@
24236 +/*
24237 + * aluno: joilnen leite
24238 + */
24239 +
24240 #ifndef TESTA_ITEM_1_H_
24241 #define TESTA_ITEM_1_H_
24242
24243 diff --git a/src/testa_item_2.c b/src/testa_item_2.c
24244 index fdf73b4..a100d92 100644
24245 --- a/src/testa_item_2.c
24246 +++ b/src/testa_item_2.c
24247 @@ -1,3 +1,7 @@
24248 +/*
24249 + * aluno: joilnen leite
24250 + */
24251 +
24252 #include <stdio.h>
24253 #include <string.h>
24254 #include <time.h>
24255 diff --git a/src/testa_item_2.h b/src/testa_item_2.h
24256 index 3544916..99999fbc 100644
24257 --- a/src/testa_item_2.h
24258 +++ b/src/testa_item_2.h
24259 @@ -1,10 +1,10 @@
24260 -#ifndef TESTA_ITEM_2_H_
24261 -#define TESTA_ITEM_2_H_
24262 -
24263 /*
24264  * aluno: joilnen leite
24265  */
24266
24267 #ifndef TESTA_ITEM_2_H_
24268 #define TESTA_ITEM_2_H_
24269 +
24270 void tela_item_2(const char *m);
24271 void testa_item_2(const char *m);
24272
24273 diff --git a/src/testa_item_3.c b/src/testa_item_3.c
24274 index a648ccf..cb696a9 100644
24275 --- a/src/testa_item_3.c
24276 +++ b/src/testa_item_3.c
24277 @@ -1,15 +1,13 @@
24278 +/*
24279 + * aluno: joilnen leite
24280 + */
24281 +
24282 #include <stdio.h>
24283 #include <string.h>
24284 #include <time.h>
24285 #include "conjunto_ordenado.h"
24286 #include "testa.h"
24287 -/*
24288 - * aluno: joilnen leite
24289 - */
24290 -
24291 -/*
24292 - * aluno: joilnen leite
24293 - */
24294 static void imprime_meu_tipo(void *);
24295
24296 typedef struct
24297 diff --git a/src/testa_item_3.h b/src/testa_item_3.h
24298 index 30f724e..513ce0c 100644
24299 --- a/src/testa_item_3.h
24300 +++ b/src/testa_item_3.h
24301 @@ -1,10 +1,10 @@
24302 -#ifndef TESTA_ITEM_3_H_
24303 -#define TESTA_ITEM_3_H_
24304 -
24305 /*
24306  * aluno: joilnen leite
24307  */
24308
24309 #ifndef TESTA_ITEM_3_H_
24310 #define TESTA_ITEM_3_H_
24311 +
24312 void tela_item_3(const char *m);
24313 void testa_item_3(const char *m);
24314
24315 diff --git a/src/testa_rb.h b/src/testa_rb.h
24316 index 8224693..7553bea 100644
24317 --- a/src/testa_rb.h
24318 +++ b/src/testa_rb.h
24319 @@ -1,10 +1,10 @@
24320 -#ifndef TESTA_ARVORE_RB_
24321 -#define TESTA_ARVORE_RB_
24322 -
24323 /*
24324  * aluno: joilnen leite

```

```

24325 */
24326
24327 #ifndef TESTA_ARVORE_RB_
24328 #define TESTA_ARVORE_RB_
24329 +
24330 /**
24331  * declaracao da funcao de teste da
24332  * arvore vermelha e preta
24333
24334 commit 012746blab981a56b821cb48f01d1e1e7b63ab0
24335 Author: Joilinen <joilinen@gmail.com>
24336 Date: Tue Jan 18 11:43:03 2022 -0300
24337
24338 melhorando a documentacao
24339
24340 diff --git a/doc/tex/relatorio.pdf b/doc/tex/relatorio.pdf
24341 index 57c96a7..401927c 100644
24342 Binary files a/doc/tex/relatorio.pdf and b/doc/tex/relatorio.pdf differ
24343 diff --git a/doc/tex/relatorio.tex b/doc/tex/relatorio.tex
24344 index 88f5221..1134f59 100644
24345 --- a/doc/tex/relatorio.tex
24346 +++ b/doc/tex/relatorio.tex
24347 @@ -432,6 +432,7 @@ que são representativos de uma rotação a esquerda e não árvores completas.\
24348 \end{tcolorbox}
24349 \ \ \
24350 \ \ \
24351 +\section{Inserção na Árvore Red-Black}
24352 \noindent Podemos agora entender melhor o funcionamento da função \enfrbinsert}.
24353 \lstinputlisting [firstnumber=43, linerange={43-145}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
24354
24355 @@ -703,63 +704,54 @@ Então chega-se ao \encaso 3}, \enfz} em linha com o nó pai, ambos vermelhos
24356 \end{tikzpicture}
24357 \end{minipage}
24358 \end{tcolorbox}
24359
24360 \enfcconjunto_ordenado.h} é o arquivo cabeçalho que deverá ser incluído no código
24361 -fonte do cliente, \enfcconjunto_ordenado.c} onde estão implementados as funções,
24362 \enfrb_black.h} e \enfrb_black.c}, cabeçalho e implementações da estrutura
24363 -de dados, árvore red-black e \enfcjurses.h} um arquivo com macros auxiliares
24364 -de impressao e formatação de saída na tela.
24365 -
24366 -sendo necessário apenas para seu uso a inclusão via, \ \
24367 -O TAD \enfcconjord} é a abstração mais alta, representando conjunto, a qual o código
24368 -cliente tem acesso e suas respectivas funções.
24369 -\enfc{\include "conjunto_ordenado.h"} onde se encontra declarado, sendo definido em \ \
24370 -\enfc("conjunto_ordenado.c"), o TAD \footnote{Tipo Abstrato de Dados}
24371 -\enfcconjord} que oculta do código cliente e manipula internamente a estrutura de dados
24372 -red black \enfc"struct rb_n"} aqui neste trabalho, declarada em \enfrb_black.h} e definida em
24373 -\enfrb_black.c}
24374 -
24375 -Além da convencional inclusão e compilação, o usuário da biblioteca, pessoa
24376 -que escreve o programa cliente, aqui neste trabalho exemplificado pelo os arquivos \ \
24377 -\enfcmain.c} \ \
24378 -\enfc{test\_tree.c} \ \
24379 -deverá escolher definir seu tipo dado, através de uma struct ou utilizar um tipo de dado nativo da
24380 -linguagem.
24381 -\renewcommand{\lstlistingname}{Fonte}
24382 -\begin{lstlisting}[caption={Exemplo do uso da biblioteca conjunto_ordenado}]
24383 -\include "conjunto_ordenado.h"
24384
24385 -int main()
24386 -{
24387 -    typedef struct {
24388 -        char *nome;
24389 -        char *telefone;
24390 -    } meu_tipo;
24391 -    meu_tipo mt0, mt1;
24392 -    mt0.nome = (char *) malloc(sizeof("nome para teste") + 1);
24393 -    strcpy(mt0.nome, "nome para teste", sizeof("nome para teste") + 1);
24394 -    mt0.telefone = (char *) malloc(sizeof("999999999") + 1);
24395 -    strcpy(mt0.telefone, "999999999", sizeof("999999999") + 1);
24396 -    seta_imprime_elemento(imprime_meu_tipo);
24397 -
24398 -    CRIA_CONJUNTO_ORDENDADO(conjunto_a, 10, mt0);
24399 -    mt1.nome = (char *) malloc(sizeof("nome para teste") + 1);
24400 -    strcpy(mt1.nome, "nome para teste", sizeof("nome para teste") + 1);
24401 -    mt1.telefone = (char *) malloc(sizeof("999999999") + 1);
24402 -    strcpy(mt1.telefone, "999999999", sizeof("999999999") + 1);
24403 -
24404 -    insere_elemento(&conjunto_a, 9, &mt1);
24405 -}
24406 -\end{lstlisting}
24407
24408 -\section{Lista de Estruturas e Funções Implementadas}
24409 -Como foi especificado a estrutura de dados utilizada para guardar os dados neste projeto
24410 +\section{Deleção na Árvore Red-Black}
24411 +\lstinputlisting [firstnumber=147, linerange={147-191}, caption = {Fragmento do red_black.c}]{../src/red_black.c}
24412 +
24413 +A deleção faz uso de outras funções que abordamos agora, a primeira delas é a \enfrb_transplant},
24414 +linha 152, ela basicamente põe o nó \enf{v} no lugar do nó \enf{u}, na linha 154, é verificado
24415 +se o nó \enf{u} é raiz, checando se esta aponta ponteiro pai para nó sentinela, caso sim,
24416 +faz nó raiz apontado por \enf{t} apontar para \enf{u}.
24417 +Os outras decisões apenas colocam \enf{v} do lado esquerdo ou direito de acordo com a posição de
24418 +\enf{u}.
24419 +O \enf{if} final checa se nó raiz é válido e por fim atualiza pai
24420 +de \enf{v}, o novo nó, com pai de \enf{u}.
24421 +
24422 +
24423 +A função a seguir a analisármos é a \enf{tree_minimum}, linha 171, este nome segue a
24424 +bibliografia, essa
24425 +função é um loop que itera de um nó qualquer até sua folha mais a esquerda, ela é importante
24426 +para se encontra o sucessor de um nó.
24427 +
24428 +E em seguida \enf{sucessor}, que usa o resultado da \enf{tree_minimum}, essa função é interessante,
24429 +ela trata quando o nó em questão não tem subárvore a direita, logo o sucessor não é possível
24430 +fazendo, \enf{tree_minimum} de nó a direita, então o \enf{while}, linha 188, itera subindo
24431 +aos nós pais e atualizando \enf{x} com nó direita de \enf{y} e \enf{y} com o pai de \enf{x},
24432 +\enf{x}, aqui funciona como nó temporário, e \enf{y} conterá o resultado esperado
24433 +o último desta iteração, como diz a função a intenção aqui é retornar o nó sucessor do
24434 +nó pesquisado, que em alguns contexto é igual ao nó mais esquerda do nó a direita, por isso
24435 +a relação desta função com \enf{tree_minimum}.
24436 +
24437 +Após estas funções podemos analisar melhor a função de deleção, junto com a inserção, os
24438 +dois pontos nevrálgicos no algoritmo, árvore vermelha e preta.
24439 +
24440 +A deleção neste trabalho, foi primeiramente implementada, sem nós sentinelas, caso se

```

```

24441 +interesse por esse processo de desenvolvimento, no apêndice está o histórico de cada linha
24442 +de código escrita, mas nesse tipo de árvore o nó sentinela tem um papel importante e facilita
24443 +o funcionamento, boa parte da lógica aqui é a iteração pelo nó pai, tando partindo das folhas
24444 +como com relação ao nó raiz, e sem nó sentinela essa ação
24445 +é mais complexa, especificamente na delegação.
24446 +\lstinputlisting [firstnumber=193, linerange={193-312}, caption = {Fragmento do red_black.c}]{../../src/red_black.c}
24447 +
24448 +
24449 +\section{Lista de Estruturas e Funções Implementadas} Como foi especificado a estrutura de dados utilizada para
guardar os dados neste projeto
24450 é a árvore binária de busca Red Black, no código esta árvore e seus nós são representados
24451 pela estrutura \enfrb{n}\footnote{Red Black Node ou nó Red Black}, a semântica escolhida
24452 é a seguinte, representamos apenas os nós e seus relacionamentos, a árvore completa fica
24453 referenciada no ato da criação do nó raiz no código cliente, logo no exemplo acima
24454 temos uma árvore nomeada \enf{conjunto_a}, enquanto nós filhos nessa árvore é adicionado
24455 com \enf{insere_elemento}
24456 +
24457 \ \ \
24458 \section{}
24459 \nocite{*}

```

Listing 1: histórico do desenvolvimento

Conjunto Ordenado

Gerado por Doxygen 1.9.2

1 Índice das estruturas de dados	1
1.1 Estruturas de dados	1
2 Índice dos ficheiros	3
2.1 Lista de ficheiros	3
3 Documentação da classe	5
3.1 Referência à estrutura conjord	5
3.1.1 Descrição detalhada	5
3.1.2 Documentação dos campos e atributos	5
3.1.2.1 cardinalidade	6
3.1.2.2 idx	6
3.1.2.3 nome	6
3.1.2.4 pr	6
3.2 Referência à estrutura meu_tipo	6
3.2.1 Descrição detalhada	6
3.2.2 Documentação dos campos e atributos	7
3.2.2.1 nome	7
3.2.2.2 telefone	7
3.3 Referência à estrutura meu_tipo2	7
3.3.1 Descrição detalhada	7
3.3.2 Documentação dos campos e atributos	7
3.3.2.1 milhas	8
3.3.2.2 nome	8
3.4 Referência à estrutura meu_tipo3	8
3.4.1 Descrição detalhada	8
3.4.2 Documentação dos campos e atributos	8
3.4.2.1 letra	8
3.5 Referência à estrutura rb_n	8
3.5.1 Descrição detalhada	9
3.5.2 Documentação dos campos e atributos	9
3.5.2.1 c	9
3.5.2.2 d	9
3.5.2.3 k	10
3.5.2.4 l	10
3.5.2.5 p	10
3.5.2.6 r	10
4 Documentação do ficheiro	11
4.1 Referência ao ficheiro src/checklist.c	11
4.1.1 Documentação das funções	11
4.1.1.1 tela_checklist()	11
4.2 checklist.c	12

4.3 Referência ao ficheiro src/checklist.h	12
4.3.1 Documentação das macros	13
4.3.1.1 ITEM1_1	13
4.3.1.2 ITEM1_2	13
4.3.1.3 ITEM1_3	13
4.3.1.4 ITEM1_4	13
4.3.1.5 ITEM2_1	13
4.3.1.6 ITEM2_2	14
4.3.1.7 ITEM2_3	14
4.3.1.8 ITEM3_1	14
4.3.1.9 ITEM3_2	14
4.3.1.10 ITEM3_3	14
4.3.1.11 ITEM4	14
4.3.2 Documentação dos valores da enumeração	14
4.3.2.1 anonymous enum	14
4.3.3 Documentação das funções	15
4.3.3.1 tela_checklist()	15
4.4 checklist.h	15
4.5 Referência ao ficheiro src/conjunto_ordenado.c	16
4.5.1 Documentação das funções	17
4.5.1.1 conjord_cardinalidade()	17
4.5.1.2 conjord_cria()	17
4.5.1.3 conjord_deleta_elemento()	17
4.5.1.4 conjord_destroi()	17
4.5.1.5 conjord_eh_vazio()	18
4.5.1.6 conjord_esvazia()	18
4.5.1.7 conjord_existe_elemento()	18
4.5.1.8 conjord_imprime_arvore_rb()	18
4.5.1.9 conjord_imprime_conjunto()	19
4.5.1.10 conjord_imprime_lista()	19
4.5.1.11 conjord_imprime_venn()	19
4.5.1.12 conjord_insere_elemento()	19
4.5.1.13 conjord_intersecciona()	20
4.5.1.14 conjord_sete_imprime_elemento()	20
4.5.1.15 conjord_subtrai()	20
4.5.1.16 conjord_une()	20
4.5.2 Documentação das variáveis	20
4.5.2.1 conjord_imprime_elemento	21
4.6 conjunto_ordenado.c	21
4.7 Referência ao ficheiro src/conjunto_ordenado.h	23
4.7.1 Documentação das macros	24
4.7.1.1 CKLSITEM	25

4.7.1.2 CM	25
4.7.1.3 CMR	25
4.7.1.4 INFO	25
4.7.1.5 INFO2	25
4.7.1.6 INFO_CARD	26
4.7.1.7 INFO_FILE	26
4.7.1.8 INFO_ST	26
4.7.1.9 LTEL	26
4.7.1.10 TIPO_DE_TERMINAL	26
4.7.2 Documentação dos tipos	26
4.7.2.1 conjord	26
4.7.3 Documentação das funções	27
4.7.3.1 conjord_cardinalidade()	27
4.7.3.2 conjord_cria()	27
4.7.3.3 conjord_deleta_elemento()	27
4.7.3.4 conjord_destroi()	28
4.7.3.5 conjord_eh_vazio()	28
4.7.3.6 conjord_esvazia()	28
4.7.3.7 conjord_exclui_elemento()	28
4.7.3.8 conjord_existe_elemento()	28
4.7.3.9 conjord_imprime_arvore_rb()	28
4.7.3.10 conjord_imprime_conjunto()	29
4.7.3.11 conjord_imprime_lista()	29
4.7.3.12 conjord_imprime_rb()	29
4.7.3.13 conjord_imprime_venn()	29
4.7.3.14 conjord_insere_elemento()	29
4.7.3.15 conjord_intersecciona()	30
4.7.3.16 conjord_sete_imprime_elemento()	30
4.7.3.17 conjord_subtrai()	30
4.7.3.18 conjord_une()	31
4.7.4 Documentação das variáveis	31
4.7.4.1 conjord_imprime_elemento	31
4.7.4.2 conjord_retorna_elemento	31
4.8 conjunto_ordenado.h	31
4.9 Referência ao ficheiro src/jcurses.h	32
4.9.1 Documentação das macros	33
4.9.1.1 S_AZUL	33
4.9.1.2 S_CARD	33
4.9.1.3 S_CHKLST	33
4.9.1.4 S_CHKLST_DONE	33
4.9.1.5 S_CM	33
4.9.1.6 S_CMR	34

4.9.1.7 S_FILEN	34
4.9.1.8 S_INFO	34
4.9.1.9 S_INFO2	34
4.9.1.10 S_INFO_ST	34
4.9.1.11 S_INFO_ST_1	34
4.9.1.12 S_INV	35
4.9.1.13 S_LIST	35
4.9.1.14 S_LTELA	35
4.9.1.15 S_NORM	35
4.9.1.16 S_UNDERL	35
4.9.1.17 S_VENN	35
4.9.1.18 S_VERM	36
4.10 jcurses.h	36
4.11 Referência ao ficheiro src/main.c	36
4.11.1 Documentação das funções	36
4.11.1.1 main()	36
4.12 main.c	37
4.13 Referência ao ficheiro src/red_black.c	37
4.13.1 Documentação das funções	38
4.13.1.1 rb_cr()	38
4.13.1.2 rb_delete()	38
4.13.1.3 rb_destroi()	38
4.13.1.4 rb_insert()	39
4.13.1.5 rb_pr()	39
4.13.1.6 rb_pr2()	40
4.13.1.7 rb_rot_dir()	40
4.13.1.8 rb_rot_esq()	40
4.13.1.9 rb_search()	41
4.13.2 Documentação das variáveis	41
4.13.2.1 rb_impr_elem	41
4.13.2.2 rb_s	41
4.14 red_black.c	42
4.15 Referência ao ficheiro src/red_black.h	46
4.15.1 Documentação dos tipos	47
4.15.1.1 cor_noh	47
4.15.1.2 rb_n	47
4.15.1.3 tipo_screen	48
4.15.2 Documentação dos valores da enumeração	48
4.15.2.1 cor_noh	48
4.15.2.2 tipo_screen	48
4.15.3 Documentação das funções	48
4.15.3.1 rb_cr()	49

4.15.3.2 rb_delete()	49
4.15.3.3 rb_destroi()	49
4.15.3.4 rb_exite_elem()	51
4.15.3.5 rb_insert()	51
4.15.3.6 rb_nelem()	52
4.15.3.7 rb_pr()	52
4.15.3.8 rb_pr2()	52
4.15.3.9 rb_rot_dir()	53
4.15.3.10 rb_rot_esq()	53
4.15.3.11 rb_search()	54
4.15.3.12 rb_troca_cor()	54
4.15.4 Documentação das variáveis	54
4.15.4.1 rb_impr_elem	54
4.15.4.2 rb_s	55
4.16 red_black.h	55
4.17 Referência ao ficheiro src/testa.h	56
4.18 testa.h	56
4.19 Referência ao ficheiro src/testa_item_1.c	56
4.19.1 Documentação das funções	56
4.19.1.1 tela_item_1()	57
4.19.1.2 testa_item_1()	57
4.20 testa_item_1.c	57
4.21 Referência ao ficheiro src/testa_item_1.h	58
4.21.1 Documentação das funções	58
4.21.1.1 tela_item_1()	58
4.21.1.2 testa_item_1()	59
4.22 testa_item_1.h	59
4.23 Referência ao ficheiro src/testa_item_2.c	59
4.23.1 Documentação das funções	59
4.23.1.1 tela_item_2()	59
4.23.1.2 testa_item_2()	60
4.24 testa_item_2.c	60
4.25 Referência ao ficheiro src/testa_item_2.h	61
4.25.1 Documentação das funções	61
4.25.1.1 tela_item_2()	61
4.25.1.2 testa_item_2()	61
4.26 testa_item_2.h	62
4.27 Referência ao ficheiro src/testa_item_3.c	62
4.27.1 Documentação das funções	62
4.27.1.1 tela_item_3()	62
4.27.1.2 testa_item_3()	63
4.28 testa_item_3.c	63

4.29 Referência ao ficheiro src/testa_item_3.h	64
4.29.1 Documentação das funções	64
4.29.1.1 tela_item_3()	64
4.29.1.2 testa_item_3()	65
4.30 testa_item_3.h	65
4.31 Referência ao ficheiro src/testa_rb.c	65
4.31.1 Documentação das funções	65
4.31.1.1 tela_testa_rb()	65
4.31.1.2 testa_arvore_rb()	66
4.32 testa_rb.c	66
4.33 Referência ao ficheiro src/testa_rb.h	67
4.33.1 Documentação das funções	67
4.33.1.1 tela_testa_rb()	67
4.33.1.2 testa_arvore_rb()	68
4.34 testa_rb.h	68
Índice	69

Capítulo 1

Índice das estruturas de dados

1.1 Estruturas de dados

Lista das estruturas de dados com uma breve descrição:

conjord	5
meu_tipo	6
meu_tipo2	7
meu_tipo3	8
rb_n	8

Capítulo 2

Índice dos ficheiros

2.1 Lista de ficheiros

Lista de todos os ficheiros com uma breve descrição:

src/ checklist.c	11
src/ checklist.h	12
src/ conjunto_ordenado.c	16
src/ conjunto_ordenado.h	23
src/ jcurses.h	32
src/ main.c	36
src/ red_black.c	37
src/ red_black.h	46
src/ testa.h	56
src/ testa_item_1.c	56
src/ testa_item_1.h	58
src/ testa_item_2.c	59
src/ testa_item_2.h	61
src/ testa_item_3.c	62
src/ testa_item_3.h	64
src/ testa_rb.c	65
src/ testa_rb.h	67

Capítulo 3

Documentação da classe

3.1 Referência à estrutura conjord

```
#include <conjunto_ordenado.h>
```

Campos de Dados

- `rb_n * pr`
- unsigned int `idx`
- char * `nome`
- int `cardinalidade`

3.1.1 Descrição detalhada

estrutura do tipo abstrado de dados esta estrutura oculta o uso da red black suas funcoes e funcionamento interno do cliente expondo apenas funcoes e operacoes com semantica relativa ao uso do cliente

Parâmetros

<i>pr</i>	ponteiro para para raiz da arvore embarcada
<i>idx</i>	indice para um array de ponteiros para as raizes, eh utilizado internamente para se ter acessor rapido a todas as raizes criadas (controle do numero de conjuntos)
<i>nome</i>	opcional para o conjunto
<i>cardinalidade</i>	do conjunto

Definido na linha 25 do ficheiro `conjunto_ordenado.h`.

3.1.2 Documentação dos campos e atributos

3.1.2.1 cardinalidade

`int cardinalidade`

Definido na linha [30](#) do ficheiro [conjunto_ordenado.h](#).

3.1.2.2 idx

`unsigned int idx`

Definido na linha [28](#) do ficheiro [conjunto_ordenado.h](#).

3.1.2.3 nome

`char* nome`

Definido na linha [29](#) do ficheiro [conjunto_ordenado.h](#).

3.1.2.4 pr

`rb_n* pr`

Definido na linha [27](#) do ficheiro [conjunto_ordenado.h](#).

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/conjunto_ordenado.h](#)

3.2 Referência à estrutura `meu_tipo`

Campos de Dados

- `char *` [nome](#)
- `char *` [telefone](#)

3.2.1 Descrição detalhada

tipo definido pelo usuario

Parâmetros

<i>nome</i>	string em c
<i>telefone</i>	string em c

Definido na linha [14](#) do ficheiro [testa_item_1.c](#).

3.2.2 Documentação dos campos e atributos

3.2.2.1 nome

```
char* nome
```

Definido na linha [15](#) do ficheiro [testa_item_1.c](#).

3.2.2.2 telefone

```
char* telefone
```

Definido na linha [16](#) do ficheiro [testa_item_1.c](#).

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/testa_item_1.c](#)

3.3 Referência à estrutura meu_tipo2

Campos de Dados

- char * [nome](#)
- int [milhas](#)

3.3.1 Descrição detalhada

Definido na linha [13](#) do ficheiro [testa_item_2.c](#).

3.3.2 Documentação dos campos e atributos

3.3.2.1 milhas

```
int milhas
```

Definido na linha 15 do ficheiro [testa_item_2.c](#).

3.3.2.2 nome

```
char* nome
```

Definido na linha 14 do ficheiro [testa_item_2.c](#).

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/testa_item_2.c](#)

3.4 Referência à estrutura meu_tipo3

Campos de Dados

- char [letra](#)

3.4.1 Descrição detalhada

Definido na linha 15 do ficheiro [testa_item_3.c](#).

3.4.2 Documentação dos campos e atributos

3.4.2.1 letra

```
char letra
```

Definido na linha 16 do ficheiro [testa_item_3.c](#).

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/testa_item_3.c](#)

3.5 Referência à estrutura rb_n

```
#include <red_black.h>
```

Campos de Dados

- `int k`
- `unsigned char c:1`
- `void * d`
- `struct rb_n * l`
- `struct rb_n * r`
- `struct rb_n * p`

3.5.1 Descrição detalhada

composicao do noh da RB

Parâmetros

<code>rb_n</code>	noh red black
<code>k</code>	chave (k de key)
<code>c</code>	cor (c de color ou cor, 0-black, 1-red)
<code>d</code>	dado (d de data ou dados)
<code>l</code>	left, ponteiro para filho a esquerda
<code>r</code>	right, ponteiro para filho a direita
<code>p</code>	parent, ponteiro para noh pai

Definido na linha 43 do ficheiro `red_black.h`.

3.5.2 Documentação dos campos e atributos

3.5.2.1 c

`unsigned char c`

Definido na linha 45 do ficheiro `red_black.h`.

3.5.2.2 d

`void* d`

Definido na linha 46 do ficheiro `red_black.h`.

3.5.2.3 k

```
int k
```

Definido na linha [44](#) do ficheiro [red_black.h](#).

3.5.2.4 l

```
struct rb_n* l
```

Definido na linha [47](#) do ficheiro [red_black.h](#).

3.5.2.5 p

```
struct rb_n * p
```

Definido na linha [47](#) do ficheiro [red_black.h](#).

3.5.2.6 r

```
struct rb_n * r
```

Definido na linha [47](#) do ficheiro [red_black.h](#).

A documentação para esta estrutura foi gerada a partir do seguinte ficheiro:

- [src/red_black.h](#)
-

Capítulo 4

Documentação do ficheiro

4.1 Referência ao ficheiro src/checklist.c

```
#include <stdio.h>
#include <string.h>
#include "conjunto_ordenado.h"
#include "checklist.h"
```

Funções

- void [tela_checklist](#) (int i)

4.1.1 Documentação das funções

4.1.1.1 [tela_checklist\(\)](#)

```
void tela_checklist (
    int i )
```

mostra itens da avaliacao

Parâmetros

<i>i</i>	especifica que itens estao selecionados
----------	---

Definido na linha [9](#) do ficheiro [checklist.c](#).

4.2 checklist.c

[Ir para a documentação deste ficheiro.](#)

```
00001 #include <stdio.h>
00002 #include <string.h>
00003 #include "conjunto_ordenado.h"
00004 #include "checklist.h"
00005
00009 void tela_checklist(int i)
00010 {
00011     int l = 5, c = 24, bo = 1UL;
00012     static unsigned int cklst_estado = 0;
00013     cklst_estado |= i ;
00014
00015     CKLSITEM(l++, c + 11, 0, S_UNDERL"avaliacao"S_NORM);
00016     CKLSITEM(l++, c, 0, " ");
00017     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_1); bo <= 1;
00018     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_2); bo <= 1;
00019     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_3); bo <= 1;
00020     CKLSITEM(l++, c, cklst_estado & bo, ITEM1_4); bo <= 1;
00021     CKLSITEM(l++, c, 0, " ");
00022     CKLSITEM(l++, c, cklst_estado & bo, ITEM2_1); bo <= 1;
00023     CKLSITEM(l++, c, cklst_estado & bo, ITEM2_2); bo <= 1;
00024     CKLSITEM(l++, c, cklst_estado & bo, ITEM2_3); bo <= 1;
00025     CKLSITEM(l++, c, 0, " ");
00026     CKLSITEM(l++, c, cklst_estado & bo, ITEM3_1); bo <= 1;
00027     CKLSITEM(l++, c, cklst_estado & bo, ITEM3_2); bo <= 1;
00028     CKLSITEM(l++, c, cklst_estado & bo, ITEM3_3); bo <= 1;
00029     CKLSITEM(l++, c, 0, " ");
00030     CKLSITEM(l, c, cklst_estado & bo, ITEM4);
00031 }
00032
00033
```

4.3 Referência ao ficheiro src/checklist.h

Macros

- #define [ITEM1_1](#) "1.1 listar ordenado por chave"
- #define [ITEM1_2](#) "1.2 cardinalidade do conjunto"
- #define [ITEM1_3](#) "1.3 verificar se conj. vazio"
- #define [ITEM1_4](#) "1.4 tornar o conjunto vazio"
- #define [ITEM2_1](#) "2.1 inserir elemento no conjunto"
- #define [ITEM2_2](#) "2.2 excluir elemento do conjunto"
- #define [ITEM2_3](#) "2.3 verif. se elem. esta no conj."
- #define [ITEM3_1](#) "3.1 itersecao entre conjuntos"
- #define [ITEM3_2](#) "3.2 uniao entre conjuntos"
- #define [ITEM3_3](#) "3.3 subtracao de conjuntos"
- #define [ITEM4](#) "4 imprimir arvore rubro negra"

Enumerações

- enum {
[l1_1](#) , [l1_2](#) , [l1_3](#) , [l1_4](#) ,
[l2_1](#) , [l2_2](#) , [l2_3](#) , [l3_1](#) ,
[l3_2](#) , [l3_3](#) , [l4](#) }

Funções

- void [tela_checklist](#) (int i)

4.3.1 Documentação das macros

4.3.1.1 ITEM1_1

```
#define ITEM1_1 "1.1 listar ordenado por chave"
```

itens da avalicao mostrados na tela

Definido na linha 14 do ficheiro [checklist.h](#).

4.3.1.2 ITEM1_2

```
#define ITEM1_2 "1.2 cardinalidade do conjunto"
```

Definido na linha 15 do ficheiro [checklist.h](#).

4.3.1.3 ITEM1_3

```
#define ITEM1_3 "1.3 verificar se conj. vazio"
```

Definido na linha 16 do ficheiro [checklist.h](#).

4.3.1.4 ITEM1_4

```
#define ITEM1_4 "1.4 tornar o conjunto vazio"
```

Definido na linha 17 do ficheiro [checklist.h](#).

4.3.1.5 ITEM2_1

```
#define ITEM2_1 "2.1 inserir elemento no conjunto"
```

Definido na linha 19 do ficheiro [checklist.h](#).

4.3.1.6 ITEM2_2

```
#define ITEM2_2 "2.2 excluir elemento do conjunto"
```

Definido na linha 20 do ficheiro [checklist.h](#).

4.3.1.7 ITEM2_3

```
#define ITEM2_3 "2.3 verif. se elem. esta no conj."
```

Definido na linha 21 do ficheiro [checklist.h](#).

4.3.1.8 ITEM3_1

```
#define ITEM3_1 "3.1 itersecao entre conjuntos"
```

Definido na linha 23 do ficheiro [checklist.h](#).

4.3.1.9 ITEM3_2

```
#define ITEM3_2 "3.2 uniao entre conjuntos"
```

Definido na linha 24 do ficheiro [checklist.h](#).

4.3.1.10 ITEM3_3

```
#define ITEM3_3 "3.3 subtracao de conjuntos"
```

Definido na linha 25 do ficheiro [checklist.h](#).

4.3.1.11 ITEM4

```
#define ITEM4 "4 imprimir arvore rubro negra"
```

Definido na linha 27 do ficheiro [checklist.h](#).

4.3.2 Documentação dos valores da enumeração

4.3.2.1 anonymous enum

```
anonymous enum
```

usado na operacao de bits na mostra de itens da avalicao

Valores de enumerações

l1↔ _1	
l1↔ _2	
l1↔ _3	
l1↔ _4	
l2↔ _1	
l2↔ _2	
l2↔ _3	
l3↔ _1	
l3↔ _2	
l3↔ _3	
l4	

Definido na linha 30 do ficheiro [checklist.h](#).

4.3.3 Documentação das funções

4.3.3.1 tela_checklist()

```
void tela_checklist (
    int i )
```

mostra itens da avaliacao

Parâmetros

<i>i</i>	especifica que itens estao selecionados
----------	---

Definido na linha 9 do ficheiro [checklist.c](#).

4.4 checklist.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef CHECKLIST_H_
00002 #define CHECKLIST_H_
```

```

00003
00004 /*
00005  * aluno: joilnen leite
00006  */
00007
00011 void tela_checklist(int i);
00012
00014 #define ITEM1_1 "1.1 listar ordenado por chave"
00015 #define ITEM1_2 "1.2 cardinalidade do conjunto"
00016 #define ITEM1_3 "1.3 verificar se conj. vazio"
00017 #define ITEM1_4 "1.4 tornar o conjunto vazio"
00018
00019 #define ITEM2_1 "2.1 inserir elemento no conjunto"
00020 #define ITEM2_2 "2.2 excluir elemento do conjunto"
00021 #define ITEM2_3 "2.3 verif. se elem. esta no conj."
00022
00023 #define ITEM3_1 "3.1 itersecao entre conjuntos"
00024 #define ITEM3_2 "3.2 uniao entre conjuntos"
00025 #define ITEM3_3 "3.3 subtracao de conjuntos"
00026
00027 #define ITEM4 "4 imprimir arvore rubro negra"
00028
00030 enum {I1_1, I1_2, I1_3, I1_4, I2_1, I2_2, I2_3, \
00031       I3_1, I3_2, I3_3, I4};
00032
00033 #endif
00034
00035

```

4.5 Referência ao ficheiro src/conjunto_ordenado.c

```

#include <stdio.h>
#include <string.h>
#include <math.h>
#include "conjunto_ordenado.h"

```

Funções

- `conjord * conjord_cria` (char *nome)
- `void conjord_destroi` (conjord *c)
- `int conjord_insererelemento` (conjord *a, int chave, void *elemento)
- `void conjord_deleta_elemento` (conjord *a, int chave)
- `int conjord_cardinalidade` (conjord *c)
- `int conjord_eh_vazio` (const conjord *a)
- `int conjord_esvazia` (conjord *c)
- `void conjord_imprime_arvore_rb` (conjord *c, int s)
- `void conjord_imprime_lista` (int *l, int b, int e)
- `int conjord_seta_imprime_elemento` (void(*i)(void *n))
- `void conjord_imprime_conjunto` (conjord *c)
- `void conjord_imprime_venn` (conjord *c)
- `conjord * conjord_intersecciona` (conjord *a, conjord *b)
- `conjord * conjord_une` (conjord *a, conjord *b)
- `conjord * conjord_subtrai` (conjord *a, conjord *b)
- `int conjord_existe_elemento` (conjord *a, int chave)

Variáveis

- `void(* conjord_imprime_elemento)(void *n)`

4.5.1 Documentação das funções

4.5.1.1 conjord_cardinalidade()

```
int conjord_cardinalidade (
    conjord * c )
```

Definido na linha 80 do ficheiro `conjunto_ordenado.c`.

4.5.1.2 conjord_cria()

```
conjord * conjord_cria (
    char * nome )
```

cria conjunto ordenado e retorna o ponteiro para ele

Parâmetros

<i>nome</i>	string que nomeia conjunto, caso sem nome, NULL
-------------	---

Definido na linha 14 do ficheiro `conjunto_ordenado.c`.

4.5.1.3 conjord_deleta_elemento()

```
void conjord_deleta_elemento (
    conjord * a,
    int chave )
```

insere elemento no conjunto ordenado

Definido na linha 67 do ficheiro `conjunto_ordenado.c`.

4.5.1.4 conjord_destroi()

```
void conjord_destroi (
    conjord * c )
```

cria conjunto ordenado e retorna o ponteiro para ele

Definido na linha 28 do ficheiro `conjunto_ordenado.c`.

4.5.1.5 conjord_eh_vazio()

```
int conjord_eh_vazio (
    const conjord * a )
```

testa se o conjunto estah vazio

Definido na linha 91 do ficheiro [conjunto_ordenado.c](#).

4.5.1.6 conjord_esvazia()

```
int conjord_esvazia (
    conjord * c )
```

deleta todos os elementos do conjunto e zera a cardinalidade

Definido na linha 110 do ficheiro [conjunto_ordenado.c](#).

4.5.1.7 conjord_existe_elemento()

```
int conjord_existe_elemento (
    conjord * a,
    int chave )
```

testa se existe elemento com a chave especificado no conjunto ordenado

Definido na linha 241 do ficheiro [conjunto_ordenado.c](#).

4.5.1.8 conjord_imprime_arvore_rb()

```
void conjord_imprime_arvore_rb (
    conjord * c,
    int s )
```

item 3 da especificação "Imprime a árvore rubro negra nesta operação, cada nó da árvore será representado pela sua chave de identificação seguida da sua cor"

Parâmetros

<i>c</i>	ponteiro para o conjunto ordenado
<i>s</i>	espaco entre os nós na apresentação da árvore

Definido na linha 121 do ficheiro [conjunto_ordenado.c](#).

4.5.1.9 conjord_imprime_conjunto()

```
void conjord_imprime_conjunto (
    conjord * c )
```

Definido na linha 151 do ficheiro [conjunto_ordenado.c](#).

4.5.1.10 conjord_imprime_lista()

```
void conjord_imprime_lista (
    int * l,
    int b,
    int e )
```

Definido na linha 126 do ficheiro [conjunto_ordenado.c](#).

4.5.1.11 conjord_imprime_venn()

```
void conjord_imprime_venn (
    conjord * c )
```

Definido na linha 168 do ficheiro [conjunto_ordenado.c](#).

4.5.1.12 conjord_inserere_elemento()

```
int conjord_inserere_elemento (
    conjord * a,
    int chave,
    void * elemento )
```

insere elemento no conjunto ordenado

Parâmetros

<i>a</i>	conjunto ordenando
<i>chave</i>	
<i>ponteiro</i>	para tipo do cliente

Definido na linha 45 do ficheiro [conjunto_ordenado.c](#).

4.5.1.13 conjord_intersecciona()

```
conjord * conjord_intersecciona (
    conjord * a,
    conjord * b )
```

intersecciona dois conjuntos formando um terceiro como resultado desta operacao retornando seu ponteiro a seguir $A \cap B = (A \cup B) - [(A - B) \cup (B - A)]$

Definido na linha 185 do ficheiro [conjunto_ordenado.c](#).

4.5.1.14 conjord_sete_imprime_elemento()

```
int conjord_sete_imprime_elemento (
    void(*) (void *) i )
```

sete a rotina de impressao do dado criado pelo cliente

Definido na linha 135 do ficheiro [conjunto_ordenado.c](#).

4.5.1.15 conjord_subtrai()

```
conjord * conjord_subtrai (
    conjord * a,
    conjord * b )
```

subtrai dois conjuntos formando um terceiro como resultado desta operacao retornando seu ponteiro a seguir

Definido na linha 233 do ficheiro [conjunto_ordenado.c](#).

4.5.1.16 conjord_une()

```
conjord * conjord_une (
    conjord * a,
    conjord * b )
```

une dois conjuntos formando um terceiro como resultado desta operacao retornando seu ponteiro a seguir

Definido na linha 210 do ficheiro [conjunto_ordenado.c](#).

4.5.2 Documentação das variáveis

4.5.2.1 conjord_imprime_elemento

```
void(* conjord_imprime_elemento) (void *n) (
    void * n )
```

Definido na linha 133 do ficheiro `conjunto_ordenado.c`.

4.6 conjunto_ordenado.c

[Ir para a documentação deste ficheiro.](#)

```
00001 #include <stdio.h>
00002 #include <string.h>
00003 #include <math.h>
00004 #include "conjunto_ordenado.h"
00005
00006 static rb_n **raiz;
00007 static unsigned int r_counter = 0;
00008
00014 conjord *conjord_cria(char *nome)
00015 {
00016     conjord *c = (conjord *) malloc (sizeof(conjord));
00017     c->pr = &rb_s;
00018     c->idx = r_counter;
00019     c->nome = nome;
00020     c->cardinalidade = 0;
00021
00022     return c;
00023 }
00024
00028 void conjord_destroi(conjord *c)
00029 {
00030     if (!c)
00031         return;
00032     if (!c->pr)
00033         return;
00034     conjord_esvazia(c);
00035     // free(c->pr);
00036     free(c);
00037 }
00038
00045 int conjord_insere_elemento(conjord *a, int chave, void *elemento)
00046 {
00047     if (a->pr != &rb_s)
00048     {
00049         rb_n *n = (rb_n *) malloc(sizeof(rb_n));
00050         n->k = chave;
00051         n->d = elemento;
00052         if (!rb_insert(&a->pr, n))
00053             a->cardinalidade++;
00054         else
00055             return -1;
00056     }
00057     else
00058     {
00059         raiz = (rb_n **) malloc(sizeof(rb_n));
00060         a->pr = raiz[a->idx] = rb_cr(chave, elemento);
00061         a->cardinalidade++;
00062     }
00063
00064     return 0;
00065 }
00066
00067 void conjord_deleta_elemento(conjord *a, int chave)
00068 {
00069     rb_n *t = rb_search(a->pr, chave);
00070
00071     if (t)
00072     {
00073         rb_delete(&a->pr, t);
00074         a->cardinalidade--;
00075         if (a->cardinalidade < 0)
00076             a->cardinalidade = 0;
00077     }
00078 }
00079
00080 int conjord_cardinalidade(conjord *c)
00081 {
00082     if (c)
```

```

00083         return c->cardinalidade;
00084     else
00085         return -1;
00086 }
00087
00091 int conjord_eh_vazio(const conjord *a)
00092 {
00093     return a->pr == &rb_s;
00094 }
00095
00096 static void conjord_esvazia_helper(rb_n *r)
00097 {
00098     if (r == &rb_s)
00099         return;
00100     rb_n *temp = r->p;
00101     conjord_esvazia_helper(r->l);
00102     free(r->d);
00103     r->d = NULL;
00104     conjord_esvazia_helper(r->r);
00105 }
00110 int conjord_esvazia(conjord *c)
00111 {
00112     if (c && c->pr)
00113     {
00114         conjord_esvazia_helper(c->pr);
00115         c->cardinalidade = 0;
00116         free(c->pr);
00117         c->pr = &rb_s;
00118     }
00119 }
00120
00121 void conjord_imprime_arvore_rb(conjord *c, int s)
00122 {
00123     rb_pr(c->pr, s, TCOLOR);
00124 }
00125
00126 void conjord_imprime_lista(int *l, int b, int e)
00127 {
00128     int i;
00129     for (i = b; i < e; i++)
00130         printf(S_LIST, i + 1, l[i]);
00131 }
00132
00133 void (*conjord_imprime_elemento)(void *n);
00134
00135 int conjord_sete_imprime_elemento(void (*i)(void *n))
00136 {
00137     conjord_imprime_elemento = i;
00138 }
00139
00140 static void conjord_imprime_conjunto_helper(rb_n *n)
00141 {
00142     if (n == &rb_s)
00143         return;
00144     conjord_imprime_conjunto_helper(n->l);
00145     printf("chave: %d\t", n->k);
00146     conjord_imprime_elemento(n->d);
00147     conjord_imprime_conjunto_helper(n->r);
00148 }
00149
00150
00151 void conjord_imprime_conjunto(conjord *c)
00152 {
00153     if (c && c->pr)
00154         conjord_imprime_conjunto_helper(c->pr);
00155 }
00156
00157 static void conjord_imprime_venn_helper(rb_n *n)
00158 {
00159     if (n == &rb_s)
00160         return;
00161     conjord_imprime_venn_helper(n->l);
00162     conjord_imprime_elemento(n->d);
00163     printf(", ");
00164     conjord_imprime_venn_helper(n->r);
00165 }
00166
00167
00168 void conjord_imprime_venn(conjord *c)
00169 {
00170     if (c && c->pr)
00171     {
00172         printf("%s = {", c->nome);
00173         conjord_imprime_venn_helper(c->pr);
00174         if (c->cardinalidade)
00175             printf("\033[D\033[D");
00176         printf("}");

```

```

00177     }
00178 }
00179
00185 conjord *conjord_intersecciona(conjord *a, conjord *b)
00186 {
00187     conjord *cj_a = conjord_cria("A");
00188     conjord *aub = conjord_une(conjord_une(cj_a, a), b);
00189     conjord *amb = conjord_subtrai(conjord_une(cj_a, a), b);
00190     conjord *bma = conjord_subtrai(conjord_une(cj_a, b), a);
00191     conjord *ambubma = conjord_une(amb, bma);
00192     return conjord_subtrai(aub, ambubma);
00193 }
00194
00195 static void conjord_une_helper(conjord *c, rb_n *n)
00196 {
00197     if (n == &rb_s)
00198         return;
00199
00200     conjord_insere_elemento(c, n->k, n->d);
00201     conjord_une_helper(c, n->l);
00202     conjord_une_helper(c, n->r);
00203 }
00204
00210 conjord *conjord_une(conjord *a, conjord *b)
00211 {
00212     conjord *cj_c = conjord_cria("C");
00213     conjord_une_helper(cj_c, a->pr);
00214     conjord_une_helper(cj_c, b->pr);
00215     return cj_c;
00216 }
00217
00218 static void conjord_subtrai_helper(conjord *c, rb_n *n)
00219 {
00220     if (n == &rb_s)
00221         return;
00222
00223     conjord_deleta_elemento(c, n->k);
00224     conjord_subtrai_helper(c, n->l);
00225     conjord_subtrai_helper(c, n->r);
00226 }
00227
00233 conjord *conjord_subtrai(conjord *a, conjord *b)
00234 {
00235     conjord *cj_c = conjord_cria("C");
00236     conjord_une_helper(cj_c, a->pr);
00237     conjord_subtrai_helper(cj_c, b->pr);
00238     return cj_c;
00239 }
00240
00241 int conjord_existe_elemento(conjord *a, int chave)
00242 {
00243     return rb_search(a->pr, chave) ? 1: 0;
00244 }
00245

```

4.7 Referência ao ficheiro src/conjunto_ordenado.h

```

#include "jurses.h"
#include "red_black.h"

```

Estruturas de Dados

- struct [conjord](#)

Macros

- #define [LTELA](#) printf([S_LTELA](#))
- #define [TIPO_DE_TERMINAL](#) TCOLOR
- #define [INFO](#)(x) printf("%s%s%s", S_INFO, x, [S_NORM](#))

- #define `INFO2(x)` `printf("%s%s%s", S_INFO2, x, S_NORM)`
- #define `INFO_ST(x)` `printf("%s%s%s%s", S_INV, S_INFO_ST, x, S_NORM)`
- #define `CM` `printf("%s", m); fflush(stdout); getc(stdin)`
- #define `CMR(x)` `printf("%s", S_CMR"q + * tecla enter *"); fflush(stdout); x = getc(stdin)`
- #define `CKLSITEM(l, c, b, x)`
- #define `INFO_FILE` `printf("%s%s%sf:%s%s", S_INFO_ST, __FILE__, S_INFO_ST_1, __FUNCTION__↵, S_NORM); fflush(stdout);`
- #define `INFO_CARD(x)` `printf("%scardinalidade: %d%s", S_CARD, x, S_NORM)`

Definições de tipos

- typedef struct `conjord` `conjord`

Funções

- `conjord * conjord_cria` (char *nome)
- int `conjord_insere_elemento` (`conjord *a`, int chave, void *elemento)
- void `conjord_deleta_elemento` (`conjord *a`, int chave)
- int `conjord_cardinalidade` ()
- int `conjord_eh_vazio` (const `conjord *a`)
- int `conjord_esvazia` ()
- void `conjord_destroi` (`conjord *a`)
- void `conjord_exclui_elemento` (`conjord *a`, int chave)
- int `conjord_existe_elemento` (`conjord *a`, int chave)
- `conjord * conjord_intersecciona` (`conjord *a`, `conjord *b`)
- `conjord * conjord_une` (`conjord *a`, `conjord *b`)
- `conjord * conjord_subtrai` (`conjord *a`, `conjord *b`)
- void `conjord_imprime_rb` (rb_n *t, int s, int e)
- void `conjord_imprime_arvore_rb` (`conjord *c`, int s)
- void `conjord_imprime_lista` (int *l, int s, int e)
- int `conjord_seta_imprime_elemento` (void(*) (void *n))
- void `conjord_imprime_conjunto` (`conjord *c`)
- void `conjord_imprime_venn` (`conjord *c`)

Variáveis

- void(* `conjord_imprime_elemento`)(void *n)
- void(* `conjord_retorna_elemento`)(void *n)

4.7.1 Documentação das macros

4.7.1.1 CKLSITEM

```
#define CKLSITEM(  
    l,  
    c,  
    b,  
    x )
```

Valor:

```
if (b) printf(S_CHKLST_DONE, l, c, x); \  
else printf(S_CHKLST, l, c, x)
```

Definido na linha 151 do ficheiro [conjunto_ordenado.h](#).

4.7.1.2 CM

```
#define CM printf("%s", m); fflush(stdout); getc(stdin)
```

Definido na linha 149 do ficheiro [conjunto_ordenado.h](#).

4.7.1.3 CMR

```
#define CMR(  
    x ) printf("%s", S_CMR"q + * tecla enter *"); fflush(stdout); x = getc(stdin)
```

Definido na linha 150 do ficheiro [conjunto_ordenado.h](#).

4.7.1.4 INFO

```
#define INFO(  
    x ) printf("%s%s%s", S_INFO, x, S_NORM)
```

Definido na linha 146 do ficheiro [conjunto_ordenado.h](#).

4.7.1.5 INFO2

```
#define INFO2(  
    x ) printf("%s%s%s", S_INFO2, x, S_NORM)
```

Definido na linha 147 do ficheiro [conjunto_ordenado.h](#).

4.7.1.6 INFO_CARD

```
#define INFO_CARD(  
    x ) printf("%scardinalidade:  %d%s", S_CARD, x, S_NORM)
```

Definido na linha 155 do ficheiro [conjunto_ordenado.h](#).

4.7.1.7 INFO_FILE

```
#define INFO_FILE printf("%s%s%sf:%s%s", S_INFO_ST, __FILE__, S_INFO_ST_1, __FUNCTION__↵  
, S_NORM); fflush(stdout);
```

Definido na linha 154 do ficheiro [conjunto_ordenado.h](#).

4.7.1.8 INFO_ST

```
#define INFO_ST(  
    x ) printf("%s%s%s", S_INV, S_INFO_ST, x, S_NORM)
```

Definido na linha 148 do ficheiro [conjunto_ordenado.h](#).

4.7.1.9 LTELA

```
#define LTELA printf(S_LTELA)
```

foi utilizado macro em vez de funcao pra evitar o footprint da chamada pois essa funcao limpa a tela e eh chamada varias a vezes eh preferivel q seja inline mas como n tem funcao inline usei a macro foi primeiro implementado com um loop mas pensando melhor dessa forma eh mais rapido

Definido na linha 144 do ficheiro [conjunto_ordenado.h](#).

4.7.1.10 TIPO_DE_TERMINAL

```
#define TIPO_DE_TERMINAL TCOLOR
```

Definido na linha 145 do ficheiro [conjunto_ordenado.h](#).

4.7.2 Documentação dos tipos

4.7.2.1 conjord

```
typedef struct conjord conjord
```

estrutura do tipo abstrado de dados esta estrutura oculta o uso da red black suas funcoes e funcionamento interno do cliente expondo apenas funcoes e operacoes com semantica relativa ao uso do cliente

Parâmetros

<i>pr</i>	ponteiro para para raiz da arvore embarcada
<i>idx</i>	índice para um array de ponteiros para as raízes, eh utilizado internamente para se ter acessor rapido a todas as raízes criadas (controle do numero de conjuntos)
<i>nome</i>	opcional para o conjunto
<i>cardinalidade</i>	do conjunto

4.7.3 Documentação das funções

4.7.3.1 `conjord_cardinalidade()`

```
int conjord_cardinalidade ( )
```

retorna a cardinalidade do conjunto

4.7.3.2 `conjord_cria()`

```
conjord * conjord_cria (
    char * nome )
```

cria conjunto ordenado e retorna o ponteiro para ele

Parâmetros

<i>nome</i>	string que nomeia conjunto, caso sem nome, NULL
-------------	---

Definido na linha [14](#) do ficheiro [conjunto_ordenado.c](#).

4.7.3.3 `conjord_deleta_elemento()`

```
void conjord_deleta_elemento (
    conjord * a,
    int chave )
```

insere elemento no conjunto ordenado

Definido na linha [67](#) do ficheiro [conjunto_ordenado.c](#).

4.7.3.4 conjord_destroi()

```
void conjord_destroi (
    conjord * c )
```

destroi e desaloca uma instancia de conjunto ordenado

cria conjunto ordenado e retorna o ponteiro para ele

Definido na linha 28 do ficheiro [conjunto_ordenado.c](#).

4.7.3.5 conjord_eh_vazio()

```
int conjord_eh_vazio (
    const conjord * a )
```

testa se o conjunto estah vazio

Definido na linha 91 do ficheiro [conjunto_ordenado.c](#).

4.7.3.6 conjord_esvazia()

```
int conjord_esvazia ( )
```

torna o conjunto vazio

4.7.3.7 conjord_exclui_elemento()

```
void conjord_exclui_elemento (
    conjord * a,
    int chave )
```

exclui elemento do conjunto ordenado através da chave é o identificador de acordo com especificacao

4.7.3.8 conjord_existe_elemento()

```
int conjord_existe_elemento (
    conjord * a,
    int chave )
```

testa se existe elemento com a chave epecificado no conjunto ordenado

Definido na linha 241 do ficheiro [conjunto_ordenado.c](#).

4.7.3.9 conjord_imprime_arvore_rb()

```
void conjord_imprime_arvore_rb (
    conjord * c,
    int s )
```

item 3 da especificao "Imprime a arvore rubro negra nesta operacao, cada noh da arvore serah representado pela sua chave de identificacao seguida da sua cor"

Parâmetros

c	ponteiro para o conjunto ordenado
s	espaco entre os nohs na apresentacao da arvore

Definido na linha 121 do ficheiro [conjunto_ordenado.c](#).

4.7.3.10 conjord_imprime_conjunto()

```
void conjord_imprime_conjunto (
    conjord * c )
```

Definido na linha 151 do ficheiro [conjunto_ordenado.c](#).

4.7.3.11 conjord_imprime_lista()

```
void conjord_imprime_lista (
    int * l,
    int s,
    int e )
```

Definido na linha 126 do ficheiro [conjunto_ordenado.c](#).

4.7.3.12 conjord_imprime_rb()

```
void conjord_imprime_rb (
    rb_n * t,
    int s,
    int e )
```

item 3 da especificação "Imprime a árvore rubro negra nesta operação, cada nó da árvore será representado pela sua chave de identificação seguida da sua cor"

4.7.3.13 conjord_imprime_venn()

```
void conjord_imprime_venn (
    conjord * c )
```

Definido na linha 168 do ficheiro [conjunto_ordenado.c](#).

4.7.3.14 conjord_insererelemento()

```
int conjord_insererelemento (
    conjord * a,
    int chave,
    void * elemento )
```

insere elemento no conjunto ordenado

insere elemento no conjunto ordenado

Parâmetros

<i>a</i>	conjunto ordenando
<i>chave</i>	
<i>ponteiro</i>	para tipo do cliente

Definido na linha [45](#) do ficheiro [conjunto_ordenado.c](#).

4.7.3.15 conjord_intersecciona()

```
conjord * conjord_intersecciona (
    conjord * a,
    conjord * b )
```

intersecciona dois conjuntos formando um terceiro como resultado desta operacao retornando seu ponteiro a seguir

intersecciona dois conjuntos formando um terceiro como resultado desta operacao retornando seu ponteiro a seguir $A \text{ intesec } B = (A \text{ uniao } B) - [(A - B) \text{ uniao } (B - A)]$

Definido na linha [185](#) do ficheiro [conjunto_ordenado.c](#).

4.7.3.16 conjord_seta_imprime_elemento()

```
int conjord_seta_imprime_elemento (
    void(*) (void *n) i )
```

seta a rotina de impressao do dado criado pelo cliente

Definido na linha [135](#) do ficheiro [conjunto_ordenado.c](#).

4.7.3.17 conjord_subtrai()

```
conjord * conjord_subtrai (
    conjord * a,
    conjord * b )
```

subtrai dois conjuntos formando um terceiro como resultado desta operacao retornando seu ponteiro a seguir

Definido na linha [233](#) do ficheiro [conjunto_ordenado.c](#).

4.7.3.18 conjord_une()

```
conjord * conjord_une (
    conjord * a,
    conjord * b )
```

une dois conjuntos formando um terceiro como resultado desta operacao retornando seu ponteiro a seguir

Definido na linha 210 do ficheiro [conjunto_ordenado.c](#).

4.7.4 Documentação das variáveis

4.7.4.1 conjord_imprime_elemento

```
void(* conjord_imprime_elemento) (void *n) (
    void * n ) [extern]
```

Definido na linha 133 do ficheiro [conjunto_ordenado.c](#).

4.7.4.2 conjord_retorna_elemento

```
void *(* conjord_retorna_elemento) (void *n) (
    void * n ) [extern]
```

4.8 conjunto_ordenado.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef CONJUNTO_ORDENADO_H
00002 #define CONJUNTO_ORDENADO_H
00003
00004 /*
00005  * aluno: joilnen leite
00006  */
00007
00008 #include "j curses.h"
00009 #include "red_black.h"
00010
00025 typedef struct conjord
00026 {
00027     rb_n *pr;
00028     unsigned int idx;
00029     char *nome;
00030     int cardinalidade;
00031 } conjord;
00032
00038 conjord *conjord_cria(char *nome);
00039
00043 int conjord_insere_elemento(conjord *a, int chave, void *elemento);
00044
00048 void conjord_deleta_elemento(conjord *a, int chave);
00049
00053 int conjord_cardinalidade();
00054
00058 int conjord_eh_vazio(const conjord *a);
00059
```

```

00063 int conjord_esvazia();
00064
00068 void conjord_destroi(conjord *a);
00069
00074 void conjord_exclui_elemento(conjord *a, int chave);
00075
00080 int conjord_existe_elemento(conjord *a, int chave);
00081
00087 conjord *conjord_intersecciona(conjord *a, conjord *b);
00088
00094 conjord *conjord_une(conjord *a, conjord *b);
00095
00101 conjord *conjord_subtrai(conjord *a, conjord *b);
00102
00109 void conjord_imprime_rb(rb_n *t, int s, int e);
00110
00119 void conjord_imprime_arvore_rb(conjord *c, int s);
00120
00121 void conjord_imprime_lista(int *l, int s, int e);
00122
00123 extern void (*conjord_imprime_elemento)(void *n);
00124 extern void* (*conjord_retorna_elemento)(void *n);
00128 int conjord_sete_imprime_elemento(void (*i)(void *n));
00129
00130 void conjord_imprime_conjunto(conjord *c);
00131
00132 void conjord_imprime_venn(conjord *c);
00133
00134
00144 #define LTELA printf(S_LTELA)
00145 #define TIPO_DE_TERMINAL TCOLOR
00146 #define INFO(x) printf("%s%s", S_INFO, x, S_NORM)
00147 #define INFO2(x) printf("%s%s", S_INFO2, x, S_NORM)
00148 #define INFO_ST(x) printf("%s%s", S_INV, S_INFO_ST, x, S_NORM)
00149 #define CM printf("%s", m); fflush(stdout); getc(stdin)
00150 #define CMR(x) printf("%s", S_CMR"q + * tecle enter *"); fflush(stdout); x = getc(stdin)
00151 #define CKLSITEM(l,c,b,x) \
00152     if (b) printf(S_CHKLSIT_DONE, l, c, x); \
00153     else printf(S_CHKLSIT, l, c, x)
00154 #define INFO_FILE printf("%s%sf:%s", S_INFO_ST, __FILE__, S_INFO_ST_1, __FUNCTION__, S_NORM);
00155     fflush(stdout);
00156 #define INFO_CARD(x) printf("%scardinalidade: %d", S_CARD, x, S_NORM)
00157 #endif
00158
00159

```

4.9 Referência ao ficheiro src/jcurses.h

Macros

- #define **S_LTELA** "\033[H\033[J"
- #define **S_CM** "\033[24;62H"
- #define **S_CMR** "\033[24;58H"
- #define **S_UNDERL** "\033[1m"
- #define **S_AZUL** "\033[44m\033[37m"
- #define **S_VERM** "\033[41m\033[37m"
- #define **S_INV** "\033[7m"
- #define **S_NORM** "\033[0m"
- #define **S_INFO** "\033[24;0H\033[7m"
- #define **S_INFO_ST** "\033[1;65H"
- #define **S_INFO_ST_1** "\033[2;65H"
- #define **S_CARD** "\033[0;60H"
- #define **S_CHKLSIT** "\033[%d;%dH%s\033[0m"
- #define **S_CHKLSIT_DONE** "\033[46m\033[30m\033[%d;%dH%s\033[0m"
- #define **S_INFO2** "\033[46m\033[30m\033[24;0H"
- #define **S_LIST** "\033[%d;78H%d"
- #define **S_FILE** "\033[0;55H"
- #define **S_VENN** **S_CHKLSIT**

4.9.1 Documentação das macros

4.9.1.1 S_AZUL

```
#define S_AZUL "\033[44m\033[37m"
```

Definido na linha 21 do ficheiro [jcurses.h](#).

4.9.1.2 S_CARD

```
#define S_CARD "\033[0;60H"
```

Definido na linha 28 do ficheiro [jcurses.h](#).

4.9.1.3 S_CHKLST

```
#define S_CHKLST "\033[%d;%dH\033[0m"
```

Definido na linha 29 do ficheiro [jcurses.h](#).

4.9.1.4 S_CHKLST_DONE

```
#define S_CHKLST_DONE "\033[46m\033[30m\033[%d;%dH\033[0m"
```

Definido na linha 30 do ficheiro [jcurses.h](#).

4.9.1.5 S_CM

```
#define S_CM "\033[24;62H"
```

Definido na linha 18 do ficheiro [jcurses.h](#).

4.9.1.6 S_CMR

```
#define S_CMR "\033[24;58H"
```

Definido na linha 19 do ficheiro [jcurses.h](#).

4.9.1.7 S_FILEN

```
#define S_FILEN "\033[0;55H"
```

Definido na linha 33 do ficheiro [jcurses.h](#).

4.9.1.8 S_INFO

```
#define S_INFO "\033[24;0H\033[7m"
```

Definido na linha 25 do ficheiro [jcurses.h](#).

4.9.1.9 S_INFO2

```
#define S_INFO2 "\033[46m\033[30m\033[24;0H"
```

Definido na linha 31 do ficheiro [jcurses.h](#).

4.9.1.10 S_INFO_ST

```
#define S_INFO_ST "\033[1;65H"
```

Definido na linha 26 do ficheiro [jcurses.h](#).

4.9.1.11 S_INFO_ST_1

```
#define S_INFO_ST_1 "\033[2;65H"
```

Definido na linha 27 do ficheiro [jcurses.h](#).

4.9.1.12 S_INV

```
#define S_INV "\033[7m"
```

Definido na linha [23](#) do ficheiro `jcurses.h`.

4.9.1.13 S_LIST

```
#define S_LIST "\033[%d;78H%d"
```

Definido na linha [32](#) do ficheiro `jcurses.h`.

4.9.1.14 S_LTELA

```
#define S_LTELA "\033[H\033[J"
```

marcos que criei baseado no que pesquisei sobre formatacao e codificacao de terminal vt100 que eh a base dos emuladores de terminal do linux curses eh como chamam o tipo de software q faz isso curses e ncurses sao exemplos, aqui eh soh uma tentativa primaria que atende as necessidades apenas dessa avaliacao

Definido na linha [17](#) do ficheiro `jcurses.h`.

4.9.1.15 S_NORM

```
#define S_NORM "\033[0m"
```

Definido na linha [24](#) do ficheiro `jcurses.h`.

4.9.1.16 S_UNDERL

```
#define S_UNDERL "\033[1m"
```

Definido na linha [20](#) do ficheiro `jcurses.h`.

4.9.1.17 S_VENN

```
#define S_VENN S_CHKLST
```

Definido na linha [34](#) do ficheiro `jcurses.h`.

4.9.1.18 S_VERM

```
#define S_VERM "\033[41m\033[37m"
```

Definido na linha [22](#) do ficheiro [jcurses.h](#).

4.10 jcurses.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef JOILNEN_CURSES_H
00002 #define JOILNEN_CURSES_H
00003
00004 /*
00005  * aluno: joilnen
00006  */
00007
00017 #define S_LTELA "\033[H\033[J"
00018 #define S_CM "\033[24;62H"
00019 #define S_CMR "\033[24;58H"
00020 #define S_UNDERL "\033[1m"
00021 #define S_AZUL "\033[44m\033[37m"
00022 #define S_VERM "\033[41m\033[37m"
00023 #define S_INV "\033[7m"
00024 #define S_NORM "\033[0m"
00025 #define S_INFO "\033[24;0H\033[7m"
00026 #define S_INFO_ST "\033[1;65H"
00027 #define S_INFO_ST_1 "\033[2;65H"
00028 #define S_CARD "\033[0;60H"
00029 #define S_CHKLST "\033[%d;%dH%s\033[0m"
00030 #define S_CHKLST_DONE "\033[46m\033[30m\033[%d;%dH%s\033[0m"
00031 #define S_INFO2 "\033[46m\033[30m\033[24;0H"
00032 #define S_LIST "\033[%d;78H%d"
00033 #define S_FILEN "\033[0;55H"
00034 #define S_VENN S_CHKLST
00035
00036 #endif
```

4.11 Referência ao ficheiro src/main.c

```
#include <stdio.h>
#include "testa.h"
```

Funções

- int [main](#) ()

4.11.1 Documentação das funções

4.11.1.1 main()

```
int main ( )
```

mensagem de controle

Definido na linha [7](#) do ficheiro [main.c](#).

4.12 main.c

[Ir para a documentação deste ficheiro.](#)

```

00001 /*
00002  * aluno: joilnen leite
00003  */
00004 #include <stdio.h>
00005 #include "testa.h"
00006
00007 int main()
00008 {
00009     char op, nitems = 4;
00010
00014     const char mc[] = S_CM" * tecla enter * ";
00015
00016     for (op = 0; op < nitems; ++op)
00017     {
00018         switch (op)
00019         {
00020             case 0:
00021                 tela_item_1(mc);
00022                 testa_item_1(mc);
00023                 break;
00024             case 1:
00025                 tela_item_2(mc);
00026                 testa_item_2(mc);
00027                 break;
00028             case 2:
00029                 tela_item_3(mc);
00030                 testa_item_3(mc);
00031                 break;
00032             case 3:
00033                 tela_testa_rb(mc);
00034                 testa_arvore_rb(mc);
00035         }
00036     }
00037
00038     return 0;
00039 }
00040
00041

```

4.13 Referência ao ficheiro src/red_black.c

```

#include "jcsurses.h"
#include "red_black.h"
#include <stdio.h>

```

Funções

- `rb_n * rb_cr (int key, void *dado)`
- `int rb_insert (rb_n **t, rb_n *z)`
- `int rb_delete (rb_n **t, rb_n *z)`
- `int rb_destroi (rb_n **a)`
- `int rb_rot_esq (rb_n **t, rb_n *x)`
- `int rb_rot_dir (rb_n **t, rb_n *x)`
- `void rb_pr (rb_n *t, int s, int e)`
- `void rb_pr2 (rb_n *t, int s, int e)`
- `rb_n * rb_search (rb_n *t, int k)`

Variáveis

- `rb_n rb_s`
- `void(* rb_impr_elem)(void *)`

4.13.1 Documentação das funções

4.13.1.1 `rb_cr()`

```
rb_n * rb_cr (
    int key,
    void * dado )
```

criacao do noh raiz

Parâmetros

<i>k</i>	chave escolhida
<i>dado</i>	especificado pelo o codigo cliente

raiz sempre colorida com preto

Definido na linha 30 do ficheiro `red_black.c`.

4.13.1.2 `rb_delete()`

```
int rb_delete (
    rb_n ** t,
    rb_n * z )
```

deleta noh da arvore red black

Parâmetros

<i>t</i>	ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao e manter a alteracao depois do retorno da funcao
<i>z</i>	ponteiro para o noh a ser deletado, para deletar por chave o TAD deve passar a saida da funcao pesquisar, num arranjo funcional na chamada

Definido na linha 200 do ficheiro `red_black.c`.

4.13.1.3 `rb_destroi()`

```
int rb_destroi (
    rb_n ** a )
```

deleta, e desaloca recursos de memoria para a arvore apontada por

Parâmetros

<i>a</i>	
----------	--

Definido na linha 314 do ficheiro red_black.c.

4.13.1.4 rb_insert()

```
int rb_insert (
    rb_n ** t,
    rb_n * z )
```

inclusao de um novo noh na arvore rb

Parâmetros

<i>t</i>	eh a noh raiz
<i>z</i>	eh o noh a ser incluido

y receberah noh pai imediato do noh adicionado x eh um ponteiro temporario para iteracao feita no loop while a seguir para chegar ao noh folha

itera para chegar no noh folha da posicao a ser inserido o novo noh

seta noh pai para o noh adicionado

se esse noh adicionado eh unico y permance nulo que neste caso apontaria para o pai que eh nulo neste cenario torna o noh adicionado raiz caso n adiciona na esquerda ou diretia de acordo com o valor da chave, noh recebe RED como padrao

ajuste

Definido na linha 48 do ficheiro red_black.c.

4.13.1.5 rb_pr()

```
void rb_pr (
    rb_n * t,
    int s,
    int e )
```

imprime arvore

Parâmetros

<i>a</i>	raiz
<i>t</i>	tipo de terminal

Definido na linha 363 do ficheiro [red_black.c](#).

4.13.1.6 rb_pr2()

```
void rb_pr2 (
    rb_n * t,
    int s,
    int e )
```

imprime arvore uma segunda versao

Parâmetros

<i>a</i>	raiz
<i>t</i>	tipo de terminal

Definido na linha 393 do ficheiro [red_black.c](#).

4.13.1.7 rb_rot_dir()

```
int rb_rot_dir (
    rb_n ** t,
    rb_n * x )
```

rotaciona subarvore a direita apartir do noh

Parâmetros

<i>a</i>	noh
----------	-----

Definido na linha 344 do ficheiro [red_black.c](#).

4.13.1.8 rb_rot_esq()

```
int rb_rot_esq (
    rb_n ** t,
    rb_n * x )
```

rotaciona a esquerda trocando as referências entre nós

Parâmetros

<i>t</i>	nó raiz (ou da subarvore)
<i>x</i>	nó que serah rotacionado

Definido na linha 325 do ficheiro red_black.c.

4.13.1.9 rb_search()

```
rb_n * rb_search (
    rb_n * t,
    int k )
```

tch

Parâmetros

<i>a</i>	noh raiz da arvore
<i>b</i>	elemento a ser inserido

Definido na linha 426 do ficheiro red_black.c.

4.13.2 Documentação das variáveis

4.13.2.1 rb_impr_elem

```
void(* rb_impr_elem) (void *) (
    void * )
```

ponteiro para funcao de impressao do tipo dado especificado pelo codigo cliente

Parâmetros

<i>void</i>	* ponteiro dado especificado pelo o codigo cliente
-------------	--

Definido na linha 23 do ficheiro red_black.c.

4.13.2.2 rb_s

```
rb_n rb_s
```

definicao do noh sentinela

Parâmetros

<i>rb</i> ↔ <i>_s</i>	red black noh setinela
--------------------------	------------------------

Definido na linha 15 do ficheiro `red_black.c`.

4.14 `red_black.c`

[Ir para a documentação deste ficheiro.](#)

```

00001 /*
00002  * aluno: joilnen leite
00003  * arquivo separado do main, como especificado
00004  * de implementacao das funcoes da redblack
00005  * utilizada como base no TAD conjunto ordenado
00006  */
00007 #include "jurses.h"
00008 #include "red_black.h"
00009 #include <stdio.h>
00010
00015 rb_n rb_s;
00016
00023 void (*rb_impr_elem)(void *);
00024
00030 rb_n* rb_cr(int key, void *dado)
00031 {
00032     rb_n *a = (rb_n *) malloc(sizeof(rb_n));
00033     a->k = key;
00035     a->c = BLACK;
00036     a->d = dado;
00037     a->p = a->l = a->r = &rb_s;
00038     rb_s.c = BLACK;
00039
00040     return a;
00041 }
00042
00048 int rb_insert(rb_n **t, rb_n *z)
00049 {
00055     rb_n *y = &rb_s;
00056     rb_n *x = *t;
00057
00062     while (x != &rb_s)
00063     {
00064         y = x;
00065         if (z->k < x->k)
00066             x = x->l;
00067         else if (z->k > x->k)
00068             x = x->r;
00069         else
00070             return -1;
00071     }
00072
00076     z->p = y;
00077
00085     if (y == &rb_s)
00086         *t = z;
00087     else if (z->k < y->k)
00088         y->l = z;
00089     else
00090         y->r = z;
00091     z->l = z->r = &rb_s;
00092     z->c = RED;
00093
00095     while (z->p->c == RED)
00096     {
00097         if (z->p == z->p->p->l)
00098         {
00099             y = z->p->p->r;
00100             if (y->c == RED)
00101             {
00102                 z->p->c = BLACK;
00103                 y->c = BLACK;
00104                 z->p->p->c = RED;
00105                 z = z->p->p;
00106             }
00107             else
00108             {
00109                 if (z == z->p->r)
00110                 {
00111                     z = z->p;
00112                     rb_rot_esq(t, z);
00113                 }
00114                 z->p->c = BLACK;
00115                 z->p->p->c = RED;
00116                 rb_rot_dir(t, z->p->p);
00117             }

```

```

00118     }
00119     else
00120     {
00121         y = z->p->p->l;
00122         if (y->c == RED)
00123         {
00124             z->p->c = BLACK;
00125             y->c = BLACK;
00126             z->p->p->c = RED;
00127             z = z->p->p;
00128         }
00129         else
00130         {
00131             if (z == z->p->l)
00132             {
00133                 z = z->p;
00134                 rb_rot_dir(t, z);
00135             }
00136             z->p->c = BLACK;
00137             z->p->p->c = RED;
00138             rb_rot_esq(t, z->p->p);
00139         }
00140     }
00141 }
00142 (*t)->c = BLACK;
00143
00144 return 0;
00145 }
00146
00152 static int rb_transplant(rb_n **t, rb_n *u, rb_n *v)
00153 {
00154     if (u->p == &rb_s)
00155         *t = v;
00156     else if (u == u->p->l)
00157         u->p->l = v;
00158     else
00159         u->p->r = v;
00160     if (v)
00161         v->p = u->p;
00162
00163     return 0;
00164 }
00165
00171 static rb_n *tree_minimum(rb_n *x)
00172 {
00173     rb_n *a = x;
00174     while (a->l != &rb_s)
00175         a = a->l;
00176     return a;
00177 }
00178
00183 static rb_n* sucessor(rb_n *x)
00184 {
00185     rb_n *y = NULL;
00186     if (x->r != &rb_s)
00187         return tree_minimum(x->r);
00188     while ((y = x->p) && x == y->r)
00189         x = y;
00190     return y;
00191 }
00192
00200 int rb_delete(rb_n **t, rb_n *z)
00201 {
00202     rb_n *x = &rb_s;
00203     rb_n *y = z;
00204     unsigned char yoc = z->c;
00205
00206     if (z->l == &rb_s)
00207     {
00208         x = z->r;
00209         rb_transplant(t, z, z->r);
00210     }
00211     else if (z->r == &rb_s)
00212     {
00213         x = z->l;
00214         rb_transplant(t, z, z->l);
00215     }
00216     else
00217     {
00218         // y = tree_minimum(z->r);
00219         y = sucessor(z);
00220         yoc = y->c;
00221         x = y->r;
00222         if (y->p == z)
00223             x->p = y;
00224         else
00225         {

```



```

00226         rb_transplant(t, y, y->r);
00227         y->r = z->r;
00228         y->r->p = y;
00229     }
00230     rb_transplant(t, z, y);
00231     y->l = z->l;
00232     y->l->p = y;
00233     y->c = z->c;
00234 }
00235
00236 if (yoc == BLACK)
00237 {
00238     rb_n *w = &rb_s;
00239     while (x != *t && x->c == BLACK)
00240     {
00241         if (x == x->p->l)
00242         {
00243             w = x->p->r;
00244             if (w->c == RED)
00245             {
00246                 w->c = BLACK;
00247                 x->p->c = RED;
00248                 rb_rot_esq(t, x->p);
00249                 w = x->p->r;
00250             }
00251
00252             if (w->l->c == BLACK && w->r->c == BLACK)
00253             {
00254                 w->c = RED;
00255                 x = x->p;
00256             }
00257             else
00258             {
00259                 if (w->r->c == BLACK)
00260                 {
00261                     w->l->c = BLACK;
00262                     w->c = RED;
00263                     rb_rot_dir(t, w);
00264                     w = x->p->r;
00265                 }
00266
00267                 w->c = x->p->c;
00268                 x->p->c = BLACK;
00269                 w->r->c = BLACK;
00270                 rb_rot_esq(t, x->p);
00271                 x = *t;
00272             }
00273         }
00274         else
00275         {
00276             w = x->p->l;
00277             if (w->c == RED)
00278             {
00279                 w->c = BLACK;
00280                 x->p->c = RED;
00281                 rb_rot_dir(t, x->p);
00282                 w = x->p->l;
00283             }
00284
00285             if (w->l->c == BLACK && w->r->c == BLACK)
00286             {
00287                 w->c = RED;
00288                 x = x->p;
00289             }
00290             else
00291             {
00292                 if (w->l->c == BLACK)
00293                 {
00294                     w->r->c = BLACK;
00295                     w->c = RED;
00296                     rb_rot_esq(t, w);
00297                     w = x->p->l;
00298                 }
00299
00300                 w->c = x->p->c;
00301                 x->p->c = BLACK;
00302                 w->l->c = BLACK;
00303                 rb_rot_dir(t, x->p);
00304                 x = *t;
00305             }
00306         }
00307     }
00308     x->c = BLACK;
00309 } /* if (yoc == BLACK) */
00310
00311 return 0;
00312 }

```

```

00313
00314 int rb_destroi(rb_n **a)
00315 {
00316     return 0;
00317 }
00318
00325 int rb_rot_esq(rb_n **t, rb_n *x)
00326 {
00327     rb_n *y = x->r;
00328     x->r = y->l;
00329     if (y->l != &rb_s)
00330         y->l->p = x;
00331     y->p = x->p;
00332     if (x->p == &rb_s)
00333         *t = y;
00334     else if (x == x->p->l)
00335         x->p->l = y;
00336     else
00337         x->p->r = y;
00338     y->l = x;
00339     x->p = y;
00340
00341     return 0;
00342 }
00343
00344 int rb_rot_dir(rb_n **t, rb_n *x)
00345 {
00346     rb_n *y = x->l;
00347     x->l = y->r;
00348     if (y->r != &rb_s)
00349         y->r->p = x;
00350     y->p = x->p;
00351     if (x->p == &rb_s)
00352         *t = y;
00353     else if (x == x->p->r)
00354         x->p->r = y;
00355     else
00356         x->p->l = y;
00357     y->r = x;
00358     x->p = y;
00359
00360     return 0;
00361 }
00362
00363 void rb_pr(rb_n *t, int s, int e)
00364 {
00365     const int dist = 5;
00366     if (t == &rb_s)
00367         return;
00368
00369     s += dist;
00370
00371     rb_pr(t->r, s, e);
00372
00373     printf("%*c", s - dist, ' ');
00374     if (t->c == BLACK)
00375     {
00376         if (e == TCOLOUR)
00377             printf("\033[44m\033[37m%d\033[0m\n", t->k);
00378         else
00379             printf("%d\n", t->k);
00380     }
00381     else
00382     {
00383         if (e == TCOLOUR)
00384             printf("\033[41m\033[37m%d\033[0m\n", t->k);
00385         else
00386             printf("\033[7m%d\033[0m\n", t->k, t->c);
00387     }
00388
00389     rb_pr(t->l, s, e);
00390 }
00391
00392
00393 void rb_pr2(rb_n *t, int s, int e)
00394 {
00395     const int dist = 8;
00396     if (t == &rb_s)
00397         return;
00398
00399     s += dist;
00400
00401     rb_pr2(t->r, s, e);
00402
00403     // for (int i = dist; i < s; i++)
00404     //     printf(" ");
00405     // printf("%*s", s, "");

```

```

00406     printf("%*c", s - dist, ' ');
00407     if (t->c == BLACK)
00408     {
00409         if (e == TCOLOR)
00410             printf("\033[44m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
00411         else
00412             printf("k:%dc:%d\n", t->k, t->c);
00413     }
00414     else
00415     {
00416         if (e == TCOLOR)
00417             printf("\033[41m\033[37mk:%dc:%d\033[0m\n", t->k, t->c);
00418         else
00419             printf("\033[7mk:%dc:%d\033[0m\n", t->k, t->c);
00420     }
00421 }
00422
00423 rb_pr2(t->l, s, e);
00424 }
00425
00426 rb_n *rb_search(rb_n *t, int k)
00427 {
00428     rb_n *y = NULL;
00429     rb_n *x = t;
00430
00431     while (x != &rb_s)
00432     {
00433         y = x;
00434         if (k < x->k)
00435             x = x->l;
00436         else if (k > x->k)
00437             x = x->r;
00438         else
00439             return x;
00440     }
00441
00442     return NULL;
00443 }
00444
00445

```

4.15 Referência ao ficheiro src/red_black.h

```
#include <stdlib.h>
```

Estruturas de Dados

- struct `rb_n`

Definições de tipos

- typedef enum `cor_noh cor_noh`
- typedef enum `tipo_screen tipo_screen`
- typedef struct `rb_n rb_n`

Enumerações

- enum `cor_noh` { `BLACK` = 0 , `RED` }
- enum `tipo_screen` { `TMONO` , `TCOLOR` }

Funções

- `rb_n * rb_cr` (int key, void *dado)
- unsigned int `rb_nelem` (`rb_n *a`)
- int `rb_insert` (`rb_n **t`, `rb_n *z`)
- `rb_n * rb_search` (`rb_n *t`, int k)
- int `rb_delete` (`rb_n **t`, `rb_n *z`)
- int `rb_exite_elem` (`rb_n *a`, int k)
- int `rb_destroi` (`rb_n **a`)
- int `rb_troca_cor` (`rb_n *a`)
- int `rb_rot_esq` (`rb_n **t`, `rb_n *x`)
- int `rb_rot_dir` (`rb_n **t`, `rb_n *x`)
- void `rb_pr` (`rb_n *t`, int s, int e)
- void `rb_pr2` (`rb_n *t`, int s, int e)

Variáveis

- `rb_n rb_s`
- void(* `rb_impr_elem`)(void *)

4.15.1 Documentação dos tipos

4.15.1.1 cor_noh

```
typedef enum cor_noh cor_noh
```

definicao do que eh considerado RED e BLACK neste codigo para maior clareza, resolvi usar enum no lugar de defines por ser mais seguro #define BLACK 0 #define RED 1

4.15.1.2 rb_n

```
typedef struct rb_n rb_n
```

composicao do noh da RB

Parâmetros

<code>rb_n</code>	noh red black
<code>k</code>	chave (k de key)
<code>c</code>	cor (c de color ou cor, 0-black, 1-red)
<code>d</code>	dado (d de data ou dados)
<code>l</code>	left, ponteiro para filho a esquerda
<code>r</code>	right, ponteiro para filho a direita
<code>p</code>	parent, ponteiro para noh pai

4.15.1.3 tipo_screen

```
typedef enum tipo_screen tipo_screen
```

tipo de terminal se suporta formatacao e cor ou se eh monocromatico simples

4.15.2 Documentação dos valores da enumeração

4.15.2.1 cor_noh

```
enum cor_noh
```

definicao do que eh considerado RED e BLACK neste codigo para maior clareza, resolvi usar enum no lugar de defines por ser mais seguro #define BLACK 0 #define RED 1

Valores de enumerações

BLACK	
RED	

Definido na linha 16 do ficheiro [red_black.h](#).

4.15.2.2 tipo_screen

```
enum tipo_screen
```

tipo de terminal se suporta formatacao e cor ou se eh monocromatico simples

Valores de enumerações

TMONO	
TCOLOR	

Definido na linha 22 do ficheiro [red_black.h](#).

4.15.3 Documentação das funções

4.15.3.1 rb_cr()

```
rb_n * rb_cr (
    int key,
    void * dado )
```

cria o primeiro noh, a raiz da arvore rb,

criacao do noh raiz

Parâmetros

<i>k</i>	chave escolhida
<i>dado</i>	especificado pelo o codigo cliente

raiz sempre colorida com preto

Definido na linha 30 do ficheiro red_black.c.

4.15.3.2 rb_delete()

```
int rb_delete (
    rb_n ** t,
    rb_n * z )
```

tc

Parâmetros

<i>a</i>	noh raiz da arvore
<i>b</i>	elemento a ser deletado

deleta noh da arvore red black

Parâmetros

<i>t</i>	ponteiro duplo para raiz, assim pq ele pode ser alterado dentro da funcao e manter a alteracao depois do retorno da fucao
<i>z</i>	ponteiro para o noh a ser deletado, para deletar por chave o TAD deve passar a saida da funcao pesquisar, num arranjo funcional na chamada

Definido na linha 200 do ficheiro red_black.c.

4.15.3.3 rb_destroi()

```
int rb_destroi (
    rb_n ** a )
```

deleta, e desaloca recursos de memoria para a arvore apontada por

Parâmetros

<i>a</i>	
----------	--

Definido na linha 314 do ficheiro red_black.c.

4.15.3.4 rb_exite_elem()

```
int rb_exite_elem (
    rb_n * a,
    int k )
```

Parâmetros

<i>a</i>	noh raiz da arvore
<i>k</i>	chave do elemento a ser checado retorna 0 se o lelemento nao existe na arvore ou diferente de zero caso exista

4.15.3.5 rb_insert()

```
int rb_insert (
    rb_n ** t,
    rb_n * z )
```

tc

Parâmetros

<i>a</i>	noh raiz da arvore
<i>b</i>	elemento a ser inserido

inclusao de um novo noh na arvore rb

Parâmetros

<i>t</i>	eh a noh raiz
<i>z</i>	eh o noh a ser incluido

y receberah noh pai imediato do noh adicionado x eh um ponteiro temporario para iteracao feita no loop while a seguir para chegar ao noh folha

itera para chegar no noh folha da posicao a ser inserido o novo noh

seta noh pai para o noh adicionado

se esse noh adicionado eh unico y permance nulo que neste caso apontaria para o pai que eh nulo neste cenario torna o noh adicionado raiz caso n adiciona na esquerda ou diretia de acordo com o valor da chave, noh recebe RED como padrao

ajuste

Definido na linha 48 do ficheiro [red_black.c](#).

4.15.3.6 rb_nelem()

```
unsigned int rb_nelem (
    rb_n * a )
```

Parâmetros

<i>a</i>	noh raiz da arvore retorna numero de elementos da arvore
----------	--

4.15.3.7 rb_pr()

```
void rb_pr (
    rb_n * t,
    int s,
    int e )
```

imprime arvore

Parâmetros

<i>a</i>	raiz
<i>t</i>	tipo de terminal

Definido na linha 363 do ficheiro [red_black.c](#).

4.15.3.8 rb_pr2()

```
void rb_pr2 (
    rb_n * t,
    int s,
    int e )
```

imprime arvore uma segunda versao

Parâmetros

<i>a</i>	raiz
<i>t</i>	tipo de terminal

Definido na linha 393 do ficheiro red_black.c.

4.15.3.9 rb_rot_dir()

```
int rb_rot_dir (
    rb_n ** t,
    rb_n * x )
```

rotaciona subarvore a direita apartir do noh

Parâmetros

<i>a</i>	noh
----------	-----

Definido na linha 344 do ficheiro red_black.c.

4.15.3.10 rb_rot_esq()

```
int rb_rot_esq (
    rb_n ** t,
    rb_n * x )
```

rotaciona subarvore a esquerda apartir do noh

Parâmetros

<i>a</i>	
----------	--

rotaciona a esquerda trocando as referências entre nós

Parâmetros

<i>t</i>	nó raiz (ou da subarvore)
<i>x</i>	nó que serah rotacionado

Definido na linha 325 do ficheiro red_black.c.

4.15.3.11 rb_search()

```
rb_n * rb_search (
    rb_n * t,
    int k )
```

tch

Parâmetros

<i>a</i>	noh raiz da arvore
<i>b</i>	elemento a ser inserido

Definido na linha [426](#) do ficheiro [red_black.c](#).

4.15.3.12 rb_troca_cor()

```
int rb_troca_cor (
    rb_n * a )
```

troca a cor de um noh apontado por

Parâmetros

<i>a</i>	
----------	--

4.15.4 Documentação das variáveis

4.15.4.1 rb_impr_elem

```
void(* rb_impr_elem) (void *) (
    void * ) [extern]
```

imprime elemento como definido pelo o codigo cliente da biblioteca

ponteiro para funcao de impressao do tipo dado especificado pelo codigo cliente

Parâmetros

<i>void</i>	* ponteiro dado especificado pelo o codigo cliente
-------------	--

Definido na linha [23](#) do ficheiro [red_black.c](#).

4.15.4.2 rb_s

rb_n rb_s [extern]

noh sentinela

definicao do noh sentinela

Parâmetros

rb↔ _s	red black noh setinela
-----------	------------------------

Definido na linha 15 do ficheiro red_black.c.

4.16 red_black.h

[Ir para a documentação deste ficheiro.](#)

```

00001 #ifndef REDBLACK_H
00002 #define REDBLACK_H
00003
00004 /*
00005  * aluno: joilnen leite
00006  */
00007 #include <stdlib.h>
00008
00016 typedef enum cor_noh {BLACK = 0, RED} cor_noh;
00017
00022 typedef enum tipo_screen {TMONO, TCOLOR} tipo_screen;
00023
00024 /*
00025  * rubro negra
00026  * 0 - todo noh eh vermelho ou preto
00027  * 1 - raiz sempre preta
00028  * 2 - noh NULL considerado noh externo preto
00029  * 3 - vermelhos tem filhos pretos
00030  * 4 - noh para suas folhas tem o mesmo numero de nohs preto
00031  */
00032
00043 typedef struct rb_n {
00044     int k;
00045     unsigned char c:1;
00046     void *d;
00047     struct rb_n *l, *r, *p;
00048 } rb_n;
00049
00053 extern rb_n rb_s;
00054
00058 rb_n* rb_cr(int key, void *dado);
00059
00064 unsigned int rb_nelem(rb_n *a);
00065
00071 int rb_insert(rb_n **t, rb_n *z);
00072
00078 rb_n *rb_search(rb_n *t, int k);
00079
00085 int rb_delete(rb_n **t, rb_n *z);
00086
00093 int rb_exite_elem(rb_n *a, int k);
00094
00099 int rb_destroi(rb_n **a);
00100
00104 int rb_troca_cor(rb_n *a);
00105
00109 int rb_rot_esq(rb_n **t, rb_n *x);
00110
00115 int rb_rot_dir(rb_n **t, rb_n *x);
00116
00120 extern void (*rb_impr_elem)(void *);
00121
00127 void rb_pr(rb_n *t, int s, int e);
00128
00134 void rb_pr2(rb_n *t, int s, int e);
00135
00136 #endif

```

4.17 Referência ao ficheiro src/testa.h

```
#include "jcurses.h"
#include "testa_item_1.h"
#include "testa_item_2.h"
#include "testa_item_3.h"
#include "testa_rb.h"
#include "checklist.h"
```

4.18 testa.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef TESTA_H_
00002 #define TESTA_H_
00003
00004 /*
00005  * aluno: joilnen leite
00006  */
00007
00008 #include "jcurses.h"
00009 #include "testa_item_1.h"
00010 #include "testa_item_2.h"
00011 #include "testa_item_3.h"
00012 #include "testa_rb.h"
00013 #include "checklist.h"
00014
00015 #endif
00016
```

4.19 Referência ao ficheiro src/testa_item_1.c

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "conjunto_ordenado.h"
#include "testa.h"
```

Estruturas de Dados

- struct [meu_tipo](#)

Funções

- void [testa_item_1](#) (const char *m)
- void [tela_item_1](#) (const char *m)

4.19.1 Documentação das funções

4.19.1.1 tela_item_1()

```
void tela_item_1 (
    const char * m )
```

Definido na linha 85 do ficheiro [testa_item_1.c](#).

4.19.1.2 testa_item_1()

```
void testa_item_1 (
    const char * m )
```

seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao

Definido na linha 44 do ficheiro [testa_item_1.c](#).

4.20 testa_item_1.c

[Ir para a documentação deste ficheiro.](#)

```
00001 #include <stdio.h>
00002 #include <string.h>
00003 #include <time.h>
00004 #include "conjunto_ordenado.h"
00005 #include "testa.h"
00006
00007 static void imprime_meu_tipo(void *);
00008
00014 typedef struct {
00015     char *nome;
00016     char *telefone;
00017 } meu_tipo;
00018
00019 static void adiciona_lista_de_elementos(conjord *c)
00020 {
00021     int i;
00022     srand(time(NULL));
00023     for (i = 0; i < 20;)
00024     {
00028         meu_tipo *mt0 = (meu_tipo *) malloc (sizeof(meu_tipo));
00029         mt0->nome = (char *) malloc(sizeof("nome para testel") + 1);
00030         strncpy(mt0->nome, "nome para testel", sizeof("nome para testel") + 1);
00031         mt0->telefone = (char *) malloc(sizeof("999999999") + 1);
00032         strncpy(mt0->telefone, "999999999", sizeof("999999999") + 1);
00033         int chave = rand() % 99 + 1;
00034         if (!conjord_insere_elemento(c, chave, mt0))
00035         {
00036             ++i;
00037             printf("chave:%d \t", chave);
00038             printf("nome:%s\t", mt0->nome);
00039             printf("telefone:%s\n", mt0->telefone);
00040         }
00041     }
00042 }
00043
00044 void testa_item_1(const char *m)
00045 {
00049     conjord_sete_imprime_elemento(imprime_meu_tipo);
00050
00051     LTELA;
00052     tela_checklist(0);
00053     CM;
00054
00055     conjord *cj_a = conjord_cria("A");
00056     LTELA;
00057     adiciona_lista_de_elementos(cj_a);
00058     INFO("* item 2.1 inserido 20 elem. chave unica entre 1 a 99");
00059     CM;
00060     LTELA;
```

```

00061     tela_checklist(1 « I2_1);
00062     CM;
00063
00064     LTELA;
00065     conjord_imprime_arvore_rb(cj_a, 0);
00066     INFO("** item 4 mostra da arvore atual");
00067     CM;
00068
00069     LTELA;
00070     tela_checklist(1 « I4);
00071     CM;
00072
00073     LTELA;
00074     conjord_imprime_conjunto(cj_a);
00075     INFO("** item 1.1 ordenado por chave de identificacao");
00076     CM;
00077
00078     LTELA;
00079     tela_checklist(1 « I1_1);
00080     CM;
00081
00082     conjord_destroi(cj_a);
00083 }
00084
00085 void tela_item_1(const char *m)
00086 {
00087     LTELA;
00088     int l = 5, c = 24;
00089     CKLSITEM(l++, c, 0, S_UNDERL"serah efetuado os testes:"S_NORM);
00090     CKLSITEM(l++, c, 0, "insersao de elementos no conjunto");
00091     CKLSITEM(l++, c, 0, "com chaves randomicas, mostra do");
00092     CKLSITEM(l++, c, 0, "estado da arvore red black");
00093     CKLSITEM(l++, c, 0, "e ordenacao dos elementos do conjunto");
00094     CKLSITEM(l++, c, 0, "o tipo de dado do usuario aqui eh uma");
00095     CKLSITEM(l++, c, 0, "estrutura com duas strings, nome e telefone");
00096     CKLSITEM(l++, c, 0, "");
00097     CKLSITEM(l++, c, 0, "                                autor: joilnen leite");
00098     INFO_FILE;
00099     CM;
00100 }
00101
00102 void imprime_meu_tipo(void *n)
00103 {
00104     printf("Nome: %s \t", ((meu_tipo *)n)->nome);
00105     printf("Telefone: %s\n", ((meu_tipo *)n)->telefone);
00106 }
00107
00108

```

4.21 Referência ao ficheiro src/testa_item_1.h

Funções

- void [tela_item_1](#) (const char *m)
- void [testa_item_1](#) (const char *m)

4.21.1 Documentação das funções

4.21.1.1 tela_item_1()

```

void tela_item_1 (
    const char * m )

```

Definido na linha [85](#) do ficheiro [testa_item_1.c](#).

4.21.1.2 testa_item_1()

```
void testa_item_1 (
    const char * m )
```

seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao

Definido na linha 44 do ficheiro [testa_item_1.c](#).

4.22 testa_item_1.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef TESTA_ITEM_1_H_
00002 #define TESTA_ITEM_1_H_
00003
00004 void tela_item_1(const char *m);
00005 void testa_item_1(const char *m);
00006
00007 #endif
00008
```

4.23 Referência ao ficheiro src/testa_item_2.c

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "conjunto_ordenado.h"
#include "testa.h"
```

Estruturas de Dados

- struct [meu_tipo2](#)

Funções

- void [testa_item_2](#) (const char *m)
- void [tela_item_2](#) (const char *m)

4.23.1 Documentação das funções

4.23.1.1 tela_item_2()

```
void tela_item_2 (
    const char * m )
```

Definido na linha 83 do ficheiro [testa_item_2.c](#).

4.23.1.2 testa_item_2()

```
void testa_item_2 (
    const char * m )
```

seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao

Definido na linha 46 do ficheiro [testa_item_2.c](#).

4.24 testa_item_2.c

[Ir para a documentação deste ficheiro.](#)

```
00001 #include <stdio.h>
00002 #include <string.h>
00003 #include <time.h>
00004 #include "conjunto_ordenado.h"
00005 #include "testa.h"
00006
00007 /*
00008  * aluno: joilnen leite
00009  */
00010
00011 static void imprime_meu_tipo(void *);
00012
00013 typedef struct {
00014     char *nome;
00015     int milhas;
00016 } meu_tipo2;
00017
00018 static void adiciona_lista_de_elementos(conjord *c)
00019 {
00020     int i;
00021     srand(time(NULL));
00022     for (i = 0; i < 20;)
00023     {
00031         meu_tipo2 *mt0 = (meu_tipo2 *) malloc (sizeof(meu_tipo2));
00032         mt0->nome = (char *) malloc(sizeof("nome para testel") + 1);
00033         strncpy(mt0->nome, "nome para testel", sizeof("nome para testel") + 1);
00034         mt0->milhas = 19;
00035         int chave = rand() % 99 + 1;
00036         if (!conjord_insere_elemento(c, chave, mt0))
00037         {
00038             ++i;
00039             printf("chave:%d \t", chave);
00040             printf("nome:%s\t", mt0->nome);
00041             printf("milhas:%d\n", mt0->milhas);
00042         }
00043     }
00044 }
00045
00046 void testa_item_2(const char *m)
00047 {
00051     conjord_setar_imprime_elemento(imprime_meu_tipo);
00052     conjord *cj_a = conjord_cria("A");
00053     int imp;
00054
00055     LTELA;
00056     adiciona_lista_de_elementos(cj_a);
00057     INFO("* item 2.2 excluir todos com chave impar");
00058     CM;
00059
00060     for (imp = 1; imp < 100; imp += 2)
00061         conjord_deleta_elemento(cj_a, imp);
00062     LTELA;
00063     conjord_imprime_conjunto(cj_a);
00064     INFO("* item 1.4 excluir todos");
00065     INFO_CARD(conjord_cardinalidade(cj_a));
00066     CM;
00067
00068     LTELA;
00069     conjord_esvazia(cj_a);
00070     INFO_CARD(conjord_cardinalidade(cj_a));
00071     INFO("* item 1.3 conjunto vazio");
00072     CM;
00073
00074     LTELA;
00075     tela_checklist(1 « I1_2);
```

```

00076     tela_checklist(1 « I1_3);
00077     tela_checklist(1 « I1_4);
00078     tela_checklist(1 « I2_2);
00079     tela_checklist(1 « I2_3);
00080     CM;
00081 }
00082
00083 void tela_item_2(const char *m)
00084 {
00085     LTELA;
00086     int l = 5, c = 24, f;
00087     char mens[8][45] =
00088     {
00089         S_UNDERL"serah efetuado os testes:"S_NORM,
00090         "exclusao de elemento do conjunto",
00091         "verif. se elemento pertence ao conjunto",
00092         "cardinalide do conjunto",
00093         "verif. se conjunto eh vazio",
00094         "tornar conjunto vazio",
00095         "o tipo de dado do usuario aqui eh uma",
00096         "estrutura com uma string e um inteiro"
00097     };
00098     for (f = 0; f < 8; f++)
00099         CKLSITEM(l++, c, 0, mens[f]);
00100     INFO_FILE;
00101     CM;
00102 }
00103
00104 void imprime_meu_tipo(void *n)
00105 {
00106     if(n)
00107     {
00108         printf("nome: %s \t", ((meu_tipo2 *)n)->nome);
00109         printf("milhas: %d\n", ((meu_tipo2 *)n)->milhas);
00110     }
00111 }
00112
00113

```

4.25 Referência ao ficheiro src/testa_item_2.h

Funções

- void [tela_item_2](#) (const char *m)
- void [testa_item_2](#) (const char *m)

4.25.1 Documentação das funções

4.25.1.1 tela_item_2()

```

void tela_item_2 (
    const char * m )

```

Definido na linha [83](#) do ficheiro [testa_item_2.c](#).

4.25.1.2 testa_item_2()

```

void testa_item_2 (
    const char * m )

```

seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao

Definido na linha [46](#) do ficheiro [testa_item_2.c](#).

4.26 testa_item_2.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef TESTA_ITEM_2_H_
00002 #define TESTA_ITEM_2_H_
00003
00004 /*
00005  * aluno: joilnen leite
00006  */
00007
00008 void tela_item_2(const char *m);
00009 void testa_item_2(const char *m);
00010
00011 #endif
00012
```

4.27 Referência ao ficheiro src/testa_item_3.c

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "conjunto_ordenado.h"
#include "testa.h"
```

Estruturas de Dados

- struct [meu_tipo3](#)

Funções

- void [tela_item_3](#) (const char *m)
- void [testa_item_3](#) (const char *m)

4.27.1 Documentação das funções

4.27.1.1 tela_item_3()

```
void tela_item_3 (
    const char * m )
```

Definido na linha [35](#) do ficheiro [testa_item_3.c](#).

4.27.1.2 testa_item_3()

```
void testa_item_3 (
    const char * m )
```

seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao

Definido na linha 55 do ficheiro [testa_item_3.c](#).

4.28 testa_item_3.c

[Ir para a documentação deste ficheiro.](#)

```
00001 #include <stdio.h>
00002 #include <string.h>
00003 #include <time.h>
00004 #include "conjunto_ordenado.h"
00005 #include "testa.h"
00006 /*
00007  * aluno: joilnen leite
00008  */
00009
00010 /*
00011  * aluno: joilnen leite
00012  */
00013 static void imprime_meu_tipo(void *);
00014
00015 typedef struct {
00016     char letra;
00017 } meu_tipo3;
00018
00019 static void adiciona_venn(conjord *c, int pos, char *letras)
00020 {
00021     char *i;
00022     for (i = letras; *i;)
00023     {
00027         meu_tipo3 *mt0 = (meu_tipo3 *) malloc (sizeof(meu_tipo3));
00028         mt0->letra = *i;
00029
00030         if (!conjord_insere_elemento(c, *i, mt0))
00031             ++i;
00032     }
00033 }
00034
00035 void tela_item_3(const char *m)
00036 {
00037     LTELA;
00038     int l = 5, c = 24, f;
00039     char mens[6][48] =
00040     {
00041         S_UNDERL"serah efetuado os testes:"S_NORM,
00042         "interseccao de conjuntos",
00043         "uniao de conjuntos",
00044         "subtracao de conjuntos",
00045         "o tipo de dado do usuario aqui eh um",
00046         "caracter representando letras do alfabeto"
00047     };
00048     for (f = 0; f < 6; f++)
00049         CKLSITEM(l++, c, 0, mens[f]);
00050     INFO_FILE(__FILE__);
00051     CM;
00052 }
00053
00054
00055 void testa_item_3(const char *m)
00056 {
00060     conjord_set_a_imprime_elemento(imprime_meu_tipo);
00061     conjord *cj_a = conjord_cria("A");
00062     conjord *cj_b = conjord_cria("B");
00063     int imp, pos0 = 0, pos1 = 1, pos2 = 2;
00064     char elem0[] = "ABCDEF";
00065     char elem1[] = "DEFGHI";
00066
00067     LTELA;
00068     adiciona_venn(cj_a, pos1, elem0);
00069     adiciona_venn(cj_b, pos2, elem1);
00070     printf("\033[8;30H");
00071     conjord_imprime_venn(cj_a);
```

```

00072     printf("\n");
00073     printf("\033[9;30H");
00074     conjord_imprime_venn(cj_b);
00075     printf("\n");
00076     printf("\033[10;30H");
00077     conjord_imprime_venn(conjord_une(cj_a, cj_b));
00078     INFO("* item 3.1 uniao de conjuntos");
00079     INFO_FILE;
00080     CM;
00081
00082     LTELA;
00083     printf("\033[8;30H");
00084     conjord_imprime_venn(cj_a);
00085     printf("\n");
00086     printf("\033[9;30H");
00087     conjord_imprime_venn(cj_b);
00088     printf("\n");
00089     printf("\033[10;30H");
00090     conjord_imprime_venn(conjord_subtrai(cj_a, cj_b));
00091     INFO("* item 3.1 subtracao de conjuntos");
00092     INFO_FILE;
00093     CM;
00094
00095     LTELA;
00096     printf("\033[8;30H");
00097     conjord_imprime_venn(cj_a);
00098     printf("\n");
00099     printf("\033[9;30H");
00100     conjord_imprime_venn(cj_b);
00101     printf("\n");
00102     printf("\033[10;30H");
00103     conjord_imprime_venn(conjord_intersecciona(cj_a, cj_b));
00104     INFO("* item 3.1 interseccao com conjuntos");
00105     INFO_FILE;
00106     CM;
00107
00108     LTELA;
00109     tela_checklist(1 « I3_1);
00110     tela_checklist(1 « I3_2);
00111     tela_checklist(1 « I3_3);
00112     CM;
00113 }
00114
00115 void imprime_meu_tipo(void *n)
00116 {
00117     if (n)
00118         printf("%c", ((meu_tipo3 *)n)->letra);
00119 }
00120
00121

```

4.29 Referência ao ficheiro src/testa_item_3.h

Funções

- void [tela_item_3](#) (const char *m)
- void [testa_item_3](#) (const char *m)

4.29.1 Documentação das funções

4.29.1.1 tela_item_3()

```

void tela_item_3 (
    const char * m )

```

Definido na linha [35](#) do ficheiro [testa_item_3.c](#).

4.29.1.2 testa_item_3()

```
void testa_item_3 (
    const char * m )
```

seta a funcao para impressao utiliza aqui a tecnica de ponteiro para funcao

Definido na linha 55 do ficheiro [testa_item_3.c](#).

4.30 testa_item_3.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef TESTA_ITEM_3_H_
00002 #define TESTA_ITEM_3_H_
00003
00004 /*
00005  * aluno: joilnen leite
00006  */
00007
00008 void tela_item_3(const char *m);
00009 void testa_item_3(const char *m);
00010
00011 #endif
00012
```

4.31 Referência ao ficheiro src/testa_rb.c

```
#include "conjunto_ordenado.h"
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "testa_rb.h"
```

Funções

- void [tela_testa_rb](#) (const char *m)
- void [testa_arvore_rb](#) (const char *m)

4.31.1 Documentação das funções

4.31.1.1 tela_testa_rb()

```
void tela_testa_rb (
    const char * m )
```

inclusao da bilioteca implementada baseada em arvores red black

Definido na linha 14 do ficheiro [testa_rb.c](#).

4.31.1.2 testa_arvore_rb()

```
void testa_arvore_rb (
    const char * m )
```

funcao main de teste separada do codigo da biblioteca como especificado definicao de um tipo exemplo pelo o usuario

Definido na linha 53 do ficheiro `testa_rb.c`.

4.32 testa_rb.c

[Ir para a documentação deste ficheiro.](#)

```
00001 /*
00002  * aluno: joilnen leite
00003  */
00004
00008 #include "conjunto_ordenado.h"
00009 #include <stdio.h>
00010 #include <string.h>
00011 #include <time.h>
00012 #include "testa_rb.h"
00013
00014 void tela_testa_rb(const char *m)
00015 {
00016     int l = 5, c = 24, f;
00017     char mens[5][48] =
00018     {
00019         S_UNDERL"serah efetuado os testes:"S_NORM,
00020         "insercao e remocao na arvore",
00021         "nao foi pedido mas me ajudou a",
00022         "checar a correcao da arvore",
00023         "chaves randomicas de 01 a 99"
00024     };
00025     LTELA;
00026     for (f = 0; f < 5; f++)
00027         CKLSITEM(l++, c, 0, mens[f]);
00028     INFO_FILE(__FILE__);
00029     CM;
00030
00031 }
00032
00033 static int pega_unico(int *l, int s)
00034 {
00035     srand(time(NULL));
00036     int i = rand() % 99 + 1;
00037     int t;
00038     for (t = 0; t < s; ++t)
00039     {
00040         if (i == l[t])
00041         {
00042             t = 0;
00043             i = rand() % 99 + 1;
00044         }
00045     }
00046
00047     return i;
00048 }
00049
00053 void testa_arvore_rb(const char *m)
00054 {
00058     typedef struct {
00059         char *nome;
00060         char *telefone;
00061     } meu_tipo;
00062
00063     int ctest;
00064     conjord *cj_b = conjord_cria("B");
00065     const int size_n = 21;
00066     int test_n[size_n];
00067     char sai = 10;
00068     while (sai == 10)
00069     {
00070         for (ctest = 0; ctest < size_n; ++ctest)
00071             test_n[ctest] = pega_unico(test_n, ctest);
00072 }
```

```

00073     LTELA;
00074     conjord_imprime_lista(test_n, 0, size_n);
00075     INFO("** insere/remove 20 elem. chave unica entre 1 a 99");
00076     CM;
00077     for (ctest = 0; ctest < size_n; ++ctest)
00078     {
00079         conjord_insere_elemento(cj_b, test_n[ctest], NULL);
00080         LTELA;
00081         conjord_imprime_lista(test_n, ctest + 1, size_n);
00082         printf("\033[H");
00083         conjord_imprime_arvore_rb(cj_b, 1);
00084         INFO("** insere/remove 20 elem. chave unica entre 1 a 99");
00085         CM;
00086     }
00087
00088     // conjord_imprime_arvore_rb(cj_b, 1);
00089     for (ctest = 0; ctest < size_n; ++ctest)
00090     {
00091         LTELA;
00092         conjord_deleta_elemento(cj_b, test_n[ctest]);
00093         conjord_imprime_lista(test_n, 0, ctest + 1);
00094         printf("\033[H");
00095         conjord_imprime_arvore_rb(cj_b, 1);
00096         INFO("** insere/remove 20 elem. chave unica entre 1 a 99");
00097         CM;
00098     }
00099     LTELA;
00100     CMR(sai);
00101 }
00102 LTELA;
00103 }
00104
00105

```

4.33 Referência ao ficheiro src/testa_rb.h

Funções

- void [tela_testa_rb](#) (const char *m)
- void [testa_arvore_rb](#) (const char *m)

4.33.1 Documentação das funções

4.33.1.1 tela_testa_rb()

```

void tela_testa_rb (
    const char * m )

```

declaracao da funcao de teste da arvore vermelha e preta

inclusao da bilioteca implementada baseada em arvores red black

Definido na linha 14 do ficheiro [testa_rb.c](#).

4.33.1.2 testa_arvore_rb()

```
void testa_arvore_rb (
    const char * m )
```

funcao main de teste separada do codigo da biblioteca como especificado definicao de um tipo exemplo pelo o usuario

Definido na linha [53](#) do ficheiro [testa_rb.c](#).

4.34 testa_rb.h

[Ir para a documentação deste ficheiro.](#)

```
00001 #ifndef TESTA_ARVORE_RB_
00002 #define TESTA_ARVORE_RB_
00003
00004 /*
00005  * aluno: joilnen leite
00006  */
00007
00012 void tela_testa_rb(const char *m);
00013 void testa_arvore_rb(const char *m);
00014
00015 #endif
00016
00017
```

Índice

BLACK

red_black.h, [48](#)

c

rb_n, [9](#)

cardinalidade

conjord, [5](#)

checklist.c

tela_checklist, [11](#)

checklist.h

l1_1, [15](#)

l1_2, [15](#)

l1_3, [15](#)

l1_4, [15](#)

l2_1, [15](#)

l2_2, [15](#)

l2_3, [15](#)

l3_1, [15](#)

l3_2, [15](#)

l3_3, [15](#)

l4, [15](#)

ITEM1_1, [13](#)

ITEM1_2, [13](#)

ITEM1_3, [13](#)

ITEM1_4, [13](#)

ITEM2_1, [13](#)

ITEM2_2, [13](#)

ITEM2_3, [14](#)

ITEM3_1, [14](#)

ITEM3_2, [14](#)

ITEM3_3, [14](#)

ITEM4, [14](#)

tela_checklist, [15](#)

CKLSITEM

conjunto_ordenado.h, [24](#)

CM

conjunto_ordenado.h, [25](#)

CMR

conjunto_ordenado.h, [25](#)

conjord, [5](#)

cardinalidade, [5](#)

conjunto_ordenado.h, [26](#)

idx, [6](#)

nome, [6](#)

pr, [6](#)

conjord_cardinalidade

conjunto_ordenado.c, [17](#)

conjunto_ordenado.h, [27](#)

conjord_cria

conjunto_ordenado.c, [17](#)

conjunto_ordenado.h, [27](#)

conjord_deleta_elemento

conjunto_ordenado.c, [17](#)

conjunto_ordenado.h, [27](#)

conjord_destroi

conjunto_ordenado.c, [17](#)

conjunto_ordenado.h, [27](#)

conjord_eh_vazio

conjunto_ordenado.c, [17](#)

conjunto_ordenado.h, [28](#)

conjord_esvazia

conjunto_ordenado.c, [18](#)

conjunto_ordenado.h, [28](#)

conjord_exclui_elemento

conjunto_ordenado.h, [28](#)

conjord_existe_elemento

conjunto_ordenado.c, [18](#)

conjunto_ordenado.h, [28](#)

conjord_imprime_arvore_rb

conjunto_ordenado.c, [18](#)

conjunto_ordenado.h, [28](#)

conjord_imprime_conjunto

conjunto_ordenado.c, [18](#)

conjunto_ordenado.h, [29](#)

conjord_imprime_elemento

conjunto_ordenado.c, [20](#)

conjunto_ordenado.h, [31](#)

conjord_imprime_lista

conjunto_ordenado.c, [19](#)

conjunto_ordenado.h, [29](#)

conjord_imprime_rb

conjunto_ordenado.h, [29](#)

conjord_imprime_venn

conjunto_ordenado.c, [19](#)

conjunto_ordenado.h, [29](#)

conjord_insere_elemento

conjunto_ordenado.c, [19](#)

conjunto_ordenado.h, [29](#)

conjord_intersecciona

conjunto_ordenado.c, [19](#)

conjunto_ordenado.h, [30](#)

conjord_retorna_elemento

conjunto_ordenado.h, [31](#)

conjord_seta_imprime_elemento

conjunto_ordenado.c, [20](#)

conjunto_ordenado.h, [30](#)

conjord_subtrai

conjunto_ordenado.c, [20](#)

- conjunto_ordenado.h, 30
- conjord_une
 - conjunto_ordenado.c, 20
 - conjunto_ordenado.h, 30
- conjunto_ordenado.c
 - conjord_cardinalidade, 17
 - conjord_cria, 17
 - conjord_deleta_elemento, 17
 - conjord_destroi, 17
 - conjord_eh_vazio, 17
 - conjord_esvazia, 18
 - conjord_existe_elemento, 18
 - conjord_imprime_arvore_rb, 18
 - conjord_imprime_conjunto, 18
 - conjord_imprime_elemento, 20
 - conjord_imprime_lista, 19
 - conjord_imprime_venn, 19
 - conjord_insere_elemento, 19
 - conjord_intersecciona, 19
 - conjord_seta_imprime_elemento, 20
 - conjord_subtrai, 20
 - conjord_une, 20
- conjunto_ordenado.h
 - CKLSITEM, 24
 - CM, 25
 - CMR, 25
 - conjord, 26
 - conjord_cardinalidade, 27
 - conjord_cria, 27
 - conjord_deleta_elemento, 27
 - conjord_destroi, 27
 - conjord_eh_vazio, 28
 - conjord_esvazia, 28
 - conjord_exclui_elemento, 28
 - conjord_existe_elemento, 28
 - conjord_imprime_arvore_rb, 28
 - conjord_imprime_conjunto, 29
 - conjord_imprime_elemento, 31
 - conjord_imprime_lista, 29
 - conjord_imprime_rb, 29
 - conjord_imprime_venn, 29
 - conjord_insere_elemento, 29
 - conjord_intersecciona, 30
 - conjord_retorna_elemento, 31
 - conjord_seta_imprime_elemento, 30
 - conjord_subtrai, 30
 - conjord_une, 30
 - INFO, 25
 - INFO2, 25
 - INFO_CARD, 25
 - INFO_FILE, 26
 - INFO_ST, 26
 - LTELA, 26
 - TIPO_DE_TERMINAL, 26
- cor_noh
 - red_black.h, 47, 48
- d
 - rb_n, 9
- l1_1
 - checklist.h, 15
- l1_2
 - checklist.h, 15
- l1_3
 - checklist.h, 15
- l1_4
 - checklist.h, 15
- l2_1
 - checklist.h, 15
- l2_2
 - checklist.h, 15
- l2_3
 - checklist.h, 15
- l3_1
 - checklist.h, 15
- l3_2
 - checklist.h, 15
- l3_3
 - checklist.h, 15
- l4
 - checklist.h, 15
- idx
 - conjord, 6
- INFO
 - conjunto_ordenado.h, 25
- INFO2
 - conjunto_ordenado.h, 25
- INFO_CARD
 - conjunto_ordenado.h, 25
- INFO_FILE
 - conjunto_ordenado.h, 26
- INFO_ST
 - conjunto_ordenado.h, 26
- ITEM1_1
 - checklist.h, 13
- ITEM1_2
 - checklist.h, 13
- ITEM1_3
 - checklist.h, 13
- ITEM1_4
 - checklist.h, 13
- ITEM2_1
 - checklist.h, 13
- ITEM2_2
 - checklist.h, 13
- ITEM2_3
 - checklist.h, 14
- ITEM3_1
 - checklist.h, 14
- ITEM3_2
 - checklist.h, 14
- ITEM3_3
 - checklist.h, 14
- ITEM4
 - checklist.h, 14
- jcurses.h
 - S_AZUL, 33

- S_CARD, 33
- S_CHKLST, 33
- S_CHKLST_DONE, 33
- S_CM, 33
- S_CMR, 33
- S_FILEN, 34
- S_INFO, 34
- S_INFO2, 34
- S_INFO_ST, 34
- S_INFO_ST_1, 34
- S_INV, 34
- S_LIST, 35
- S_LTELA, 35
- S_NORM, 35
- S_UNDERL, 35
- S_VENN, 35
- S_VERM, 35
- k
 - rb_n, 9
- l
 - rb_n, 10
- letra
 - meu_tipo3, 8
- LTELA
 - conjunto_ordenado.h, 26
- main
 - main.c, 36
- main.c
 - main, 36
- meu_tipo, 6
 - nome, 7
 - telefone, 7
- meu_tipo2, 7
 - milhas, 7
 - nome, 8
- meu_tipo3, 8
 - letra, 8
- milhas
 - meu_tipo2, 7
- nome
 - conjord, 6
 - meu_tipo, 7
 - meu_tipo2, 8
- p
 - rb_n, 10
- pr
 - conjord, 6
- r
 - rb_n, 10
- rb_cr
 - red_black.c, 38
 - red_black.h, 48
- rb_delete
 - red_black.c, 38
- red_black.h, 49
- rb_destroi
 - red_black.c, 38
 - red_black.h, 49
- rb_exite_elem
 - red_black.h, 51
- rb_impr_elem
 - red_black.c, 41
 - red_black.h, 54
- rb_insert
 - red_black.c, 39
 - red_black.h, 51
- rb_n, 8
 - c, 9
 - d, 9
 - k, 9
 - l, 10
 - p, 10
 - r, 10
 - red_black.h, 47
- rb_nelem
 - red_black.h, 52
- rb_pr
 - red_black.c, 39
 - red_black.h, 52
- rb_pr2
 - red_black.c, 40
 - red_black.h, 52
- rb_rot_dir
 - red_black.c, 40
 - red_black.h, 53
- rb_rot_esq
 - red_black.c, 40
 - red_black.h, 53
- rb_s
 - red_black.c, 41
 - red_black.h, 54
- rb_search
 - red_black.c, 41
 - red_black.h, 53
- rb_troca_cor
 - red_black.h, 54
- RED
 - red_black.h, 48
- red_black.c
 - rb_cr, 38
 - rb_delete, 38
 - rb_destroi, 38
 - rb_impr_elem, 41
 - rb_insert, 39
 - rb_pr, 39
 - rb_pr2, 40
 - rb_rot_dir, 40
 - rb_rot_esq, 40
 - rb_s, 41
 - rb_search, 41
- red_black.h
 - BLACK, 48

- cor_noh, 47, 48
- rb_cr, 48
- rb_delete, 49
- rb_destroi, 49
- rb_exite_elem, 51
- rb_impr_elem, 54
- rb_insert, 51
- rb_n, 47
- rb_nelem, 52
- rb_pr, 52
- rb_pr2, 52
- rb_rot_dir, 53
- rb_rot_esq, 53
- rb_s, 54
- rb_search, 53
- rb_troca_cor, 54
- RED, 48
- TCOLOR, 48
- tipo_screen, 47, 48
- TMONO, 48
- S_AZUL
 - jcurses.h, 33
- S_CARD
 - jcurses.h, 33
- S_CHKLIST
 - jcurses.h, 33
- S_CHKLIST_DONE
 - jcurses.h, 33
- S_CM
 - jcurses.h, 33
- S_CMR
 - jcurses.h, 33
- S_FILEN
 - jcurses.h, 34
- S_INFO
 - jcurses.h, 34
- S_INFO2
 - jcurses.h, 34
- S_INFO_ST
 - jcurses.h, 34
- S_INFO_ST_1
 - jcurses.h, 34
- S_INV
 - jcurses.h, 34
- S_LIST
 - jcurses.h, 35
- S_LTELA
 - jcurses.h, 35
- S_NORM
 - jcurses.h, 35
- S_UNDERL
 - jcurses.h, 35
- S_VENN
 - jcurses.h, 35
- S_VERM
 - jcurses.h, 35
- src/checklist.c, 11, 12
- src/checklist.h, 12, 15
- src/conjunto_ordenado.c, 16, 21
- src/conjunto_ordenado.h, 23, 31
- src/jcurses.h, 32, 36
- src/main.c, 36, 37
- src/red_black.c, 37, 42
- src/red_black.h, 46, 55
- src/testa.h, 56
- src/testa_item_1.c, 56, 57
- src/testa_item_1.h, 58, 59
- src/testa_item_2.c, 59, 60
- src/testa_item_2.h, 61, 62
- src/testa_item_3.c, 62, 63
- src/testa_item_3.h, 64, 65
- src/testa_rb.c, 65, 66
- src/testa_rb.h, 67, 68
- TCOLOR
 - red_black.h, 48
- tela_checklist
 - checklist.c, 11
 - checklist.h, 15
- tela_item_1
 - testa_item_1.c, 56
 - testa_item_1.h, 58
- tela_item_2
 - testa_item_2.c, 59
 - testa_item_2.h, 61
- tela_item_3
 - testa_item_3.c, 62
 - testa_item_3.h, 64
- tela_testa_rb
 - testa_rb.c, 65
 - testa_rb.h, 67
- telefone
 - meu_tipo, 7
- testa_arvore_rb
 - testa_rb.c, 65
 - testa_rb.h, 67
- testa_item_1
 - testa_item_1.c, 57
 - testa_item_1.h, 58
- testa_item_1.c
 - tela_item_1, 56
 - testa_item_1, 57
- testa_item_1.h
 - tela_item_1, 58
 - testa_item_1, 58
- testa_item_2
 - testa_item_2.c, 59
 - testa_item_2.h, 61
- testa_item_2.c
 - tela_item_2, 59
 - testa_item_2, 59
- testa_item_2.h
 - tela_item_2, 61
 - testa_item_2, 61
- testa_item_3
 - testa_item_3.c, 62
 - testa_item_3.h, 64

testa_item_3.c
 tela_item_3, [62](#)
 testa_item_3, [62](#)
testa_item_3.h
 tela_item_3, [64](#)
 testa_item_3, [64](#)
testa_rb.c
 tela_testa_rb, [65](#)
 testa_arvore_rb, [65](#)
testa_rb.h
 tela_testa_rb, [67](#)
 testa_arvore_rb, [67](#)
TIPO_DE_TERMINAL
 conjunto_ordenado.h, [26](#)
tipo_screen
 red_black.h, [47](#), [48](#)
TMONO
 red_black.h, [48](#)
