

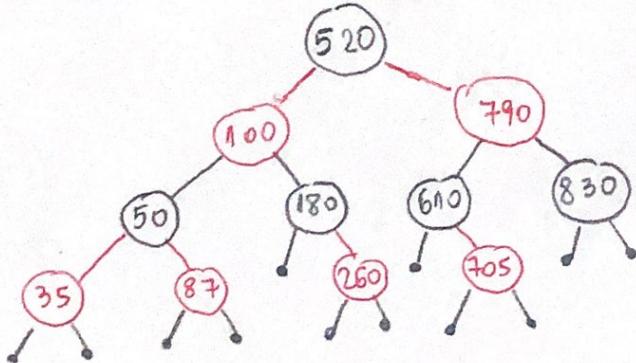
P2 INF1010 - 2021.2

DANIEL STULBERG HUF - 1920468 - TURMA 3WA

10/10

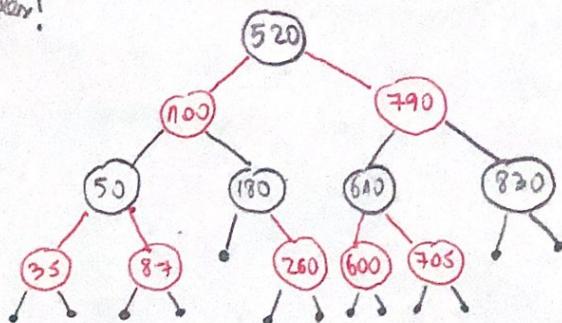
QUESTÃO 1

→ Anore rubro - negra inicial:



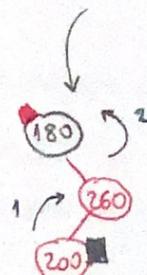
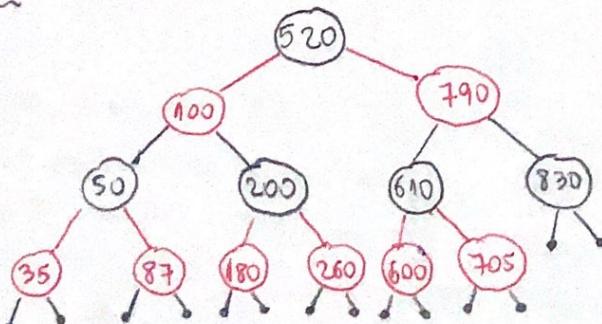
(a) Inscrição sobre 600

→ Operações imóveis!



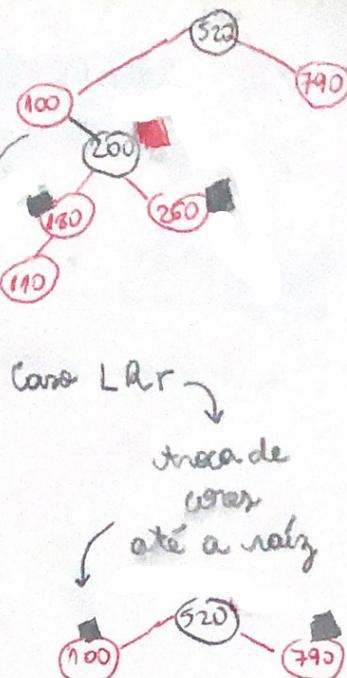
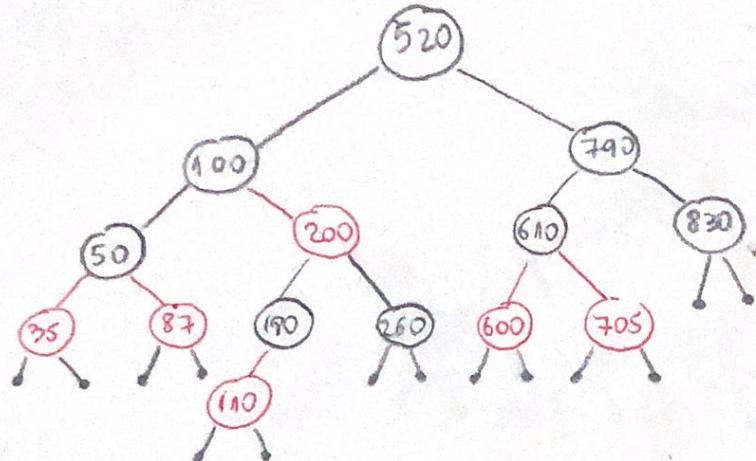
(b) Innergo chose 200

b) Inversão cheve 200
→ Caso P-L b → inversão dupla direita-esquerda, seguida de troca de cores



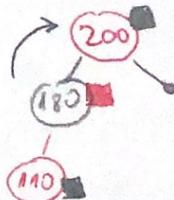
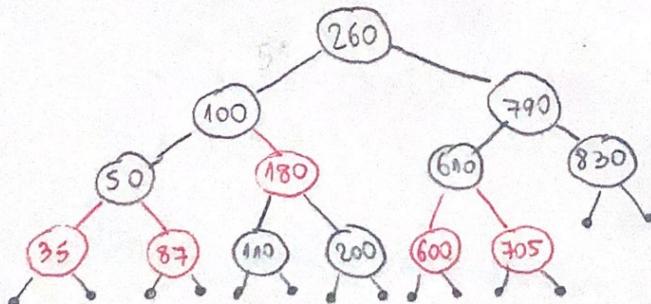
(c) Inserção chave 110

→ Caso LLr → troca de cores



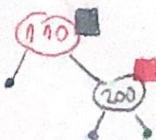
(d) Remoção chave 520

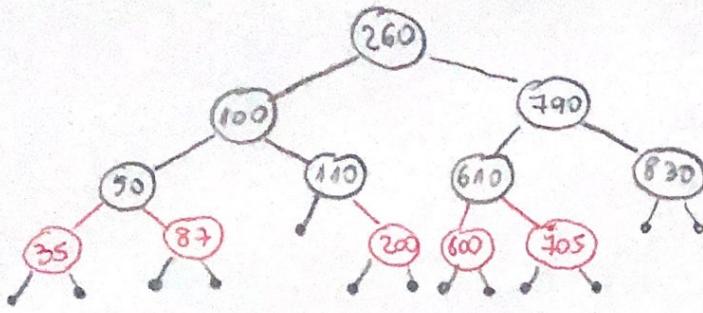
→ Substituir pelo maior nó da sub-árvore à esquerda (260),
depois remover 520. Temos um caso em que o antigo nó 260
agora tem um irmão preto à esquerda e sobrinho vermelho à
direita para corrigir isso, seguida por troca de cores.



(e) Remoção chave 180

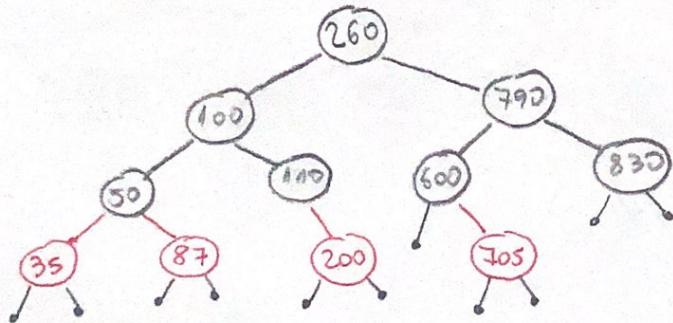
→ Substituir pelo maior nó da sub-árvore à esquerda (110),
depois remover 180. Temos um caso em que o antigo nó 180 agora
tem um irmão preto e dois sobrinhos pretos. Troca de cores em
cima para corrigir isso.





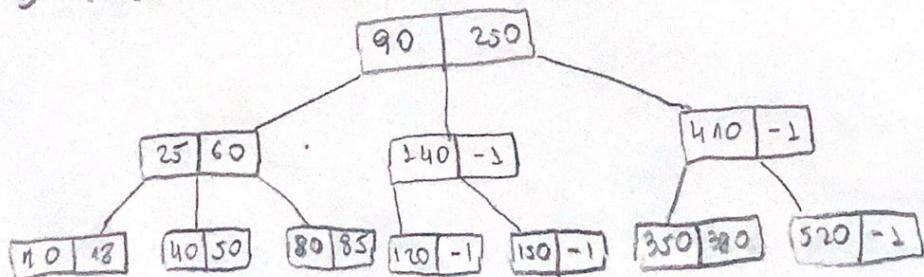
(f) Remoção chave 610

→ Substituir pelo maior nó da sub-árvore à esquerda (600), depois removê-lo. Tratar cor da chave 600.



QUESTÃO 2

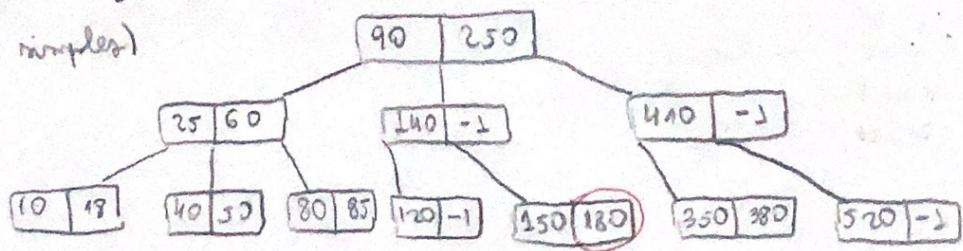
Árvore B inicial:



(a) Inserção chave 180

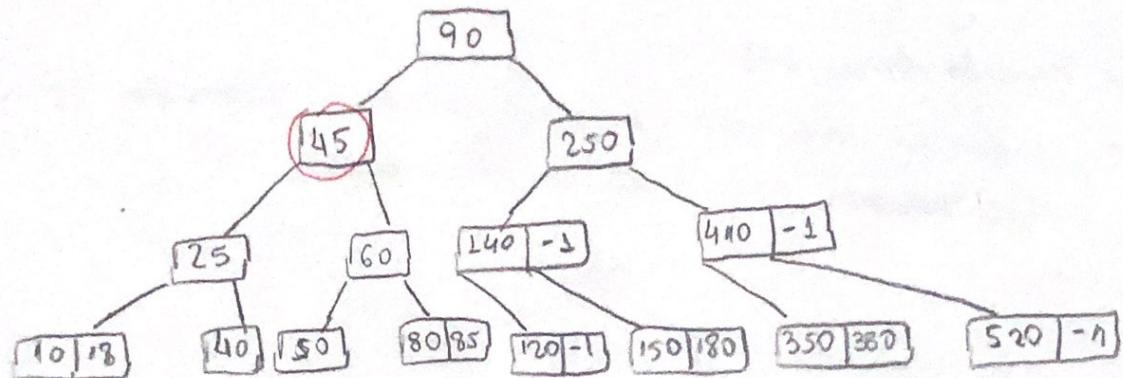
(inserção simples)

↓
folha
não tem
2 chaves



(b) Inserção chave 45

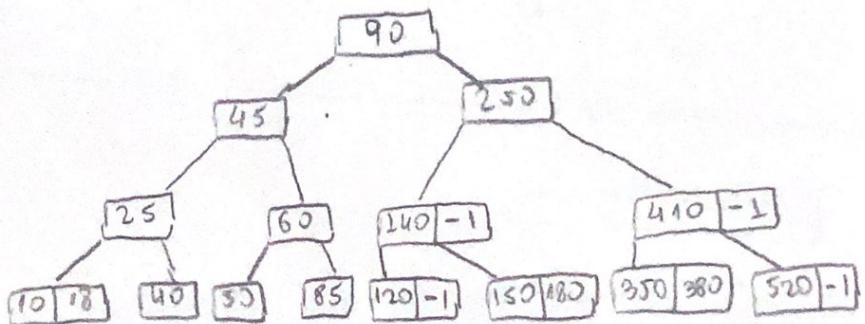
(move 45 para o pai,
depois move 45 para o
pai da raiz, depois
(raiz nova raiz com 90)



(c) Remoção chave 80

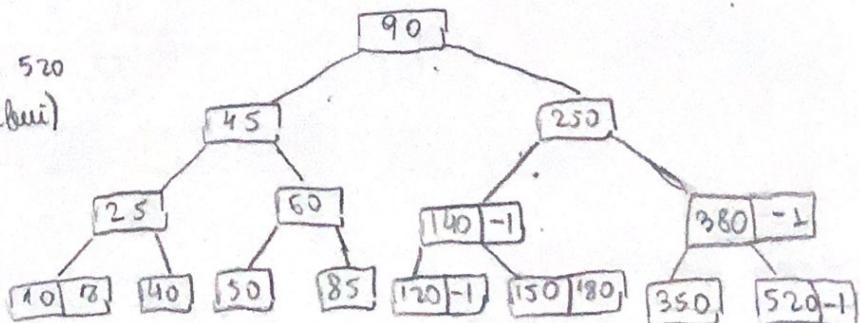
(remoção simples)

↓
folha removida
2 chaves



(d) Remoção chave 410

(troca 410 com sucessor 520
remove 410 e redistribui)



QUESTÃO 3

$$\text{Função de Hash} \rightarrow h(x) = ((x \% 11) + k) \% 11, \quad k = 0, 1, 2, \dots$$

(a) Inver_2 done ?

$$h(9) = (9+0) \% 11 = 9 \rightarrow \text{menhuma colisão}$$

Tabela Hash:

② foreign chose 10

$$h(10) = (10+0)\% \cdot 11 = 10 \rightarrow \text{menor na coluna}$$

Takela Hash:

③ inversão clare 17

$$h(17) = (6+0)\% \cdot 17 = 6 \rightarrow \text{menor taxa de inflação}$$

Tatela
Hank

(4) ~~Torrey~~ chose 3

$$h(3) = (3+0) \% 11 = 3 \rightarrow \text{menherra colisão}$$

Taleela Hashim

⑤ Inverão Jornal 2

$$h(2) = (2+0)\%11 = 2 \rightarrow \text{nenhuma colisão}$$

Takela Hash:

⑥ Inverigo done 13 ✓ Tentative 0

$$h(53) = (2+0)\% \cdot 11 = 2 \rightarrow \text{conflicto primario}$$

Tendência 1 → $(2+1)\% \cdot 11 = 3 \rightarrow$ cara não está negada

Tentativa 2 → (2+2) % AA = 4 → menor taxa de calirae

Tabela Hash:

③ Izmerješ stave L5

$$h(25) = (4+0)\% \cdot 11 = 4 \rightarrow \text{conflicto secundário}$$

→ close 13 reais para próxima casa disponível (5)

- Stellar Mass:

⑧ Inspeção cheve 25 Tentativa 0

$$h(25) = (3+0) \% 11 = 3 \rightarrow \text{conflicto primário}$$

Tentativa 1 → $(3+1)\% \text{ de } 11 = 4 \rightarrow$ cara não está ruim
 → não está ruim

Tentativa 1 $\rightarrow (3+1)\% \cdot 11 = 4 \rightarrow$ casa não está vazia
Tentativa 2 $\rightarrow (3+2)\% \cdot 11 = 5 \rightarrow$ casa não está vazia

$$\text{Tortalina 2} \rightarrow (3+2) \% \cdot 11 = 5 \rightarrow$$

$$\text{Tentativa 3} \rightarrow (3+3)\%_{AA} = 6 \rightarrow \begin{array}{l} \text{casa não seria no j} \\ \text{nenhuma colisão.} \end{array}$$

Tentativa 3 $\rightarrow (3+2)\% = 11$
Tentativa 4 $\rightarrow (3+4)\% = 11 = 7 \rightarrow$ nenhuma colisão.

Talula Hash:

⑨ Immense shore 24

$$b_{(14)} = (320)\% \cdot 11 = 3 \rightarrow \text{confiabilidade primária}$$

$h(14) = (3+0)\% \cdot 11 = 3 \rightarrow$ Tentativa 1 → 4 (cheio) | Tentativa 2 → 5 (cheio) | Tentativa 3 → 6 (cheio)
 Tentativa 4 → 7 (cheio) | Tentativa 5 → 8 (menos volumoso)

Tentakinea I → 1 | Tentakinea S → 8 | Membranae

Tentative 4 → + (chlo) |

Takela Nash:

(b)

ii. Remoção chave 3

$$\text{Hash}(3) = 3$$

↳ O bit de exclusão no índice 3 irá mudar para 1, de modo que o campo estaria aberto para novos inserções.

O campo que aponta para o próximo deseja permanecer, para não perder a posição das outras chaves da mesma "família".

Tabela Hash:

	0	1	2	3	4	5	6	7	8	9	10
Chave	-1	-1	2	-1	15	13	17	25	14	9	10
Puxa	-1	-1	5	8	-1	-1	-1	-1	7	-1	-1
Exclusão	0	0	0	1	0	0	0	0	0	0	0

iii. Inserção chave 4

$$h(4) = (4+0) \% 11 \rightarrow 4 \rightarrow \text{conflicto primário}$$

Tentativa 1 \rightarrow 5 (chave) | Tentativa 2 \rightarrow 6 (chave) | Tentativa 3 \rightarrow 7 (chave)

Tentativa 4 \rightarrow 8 (chave) | Tentativa 5 \rightarrow 9 (chave) | Tentativa 6 \rightarrow 10 (chave)

Tentativa 7 \rightarrow 0 (nenhuma colisão)

Tabela Hash:

	0	1	2	3	4	5	6	7	8	9	10
Chave	4	-1	2	-1	15	13	17	25	14	9	10
Puxa	-1	-1	5	8	0	-1	-1	-1	7	-1	-1
Exclusão	0	0	0	1	0	0	0	0	0	0	0

QUESTÃO 4

(a) OBS: O código abaixo acaba destruindo a heap original (+ com o heaproot flag), portanto se for implementado é recomendado fazer uma cópia da heap original para não perdê-la.

int nMenoresQue (Heap* heap, int x)

```
{  
    /* checa se existem elementos na heap */  
    if (heap->pos <= 0)  
        return 0;  
    /* retira a raiz da heap */  
    int topo = minheap_remove(heap);  
    /* compara raiz ao valor x */  
    if (topo >= x)  
        return 0;  
    /* chamada recursiva com a heap corrigida após a retirada da raiz */  
    return (1 + nMenoresQue (heap, x));  
}
```

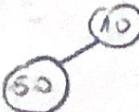
funcão já existente
no TAD heap, que
tira a raiz e corrige
a heap que restou

(b) Sequência: 10 - 50 - 20 - 5 - 60 - 80 - 15

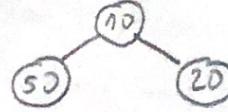
① Inserção 10

10

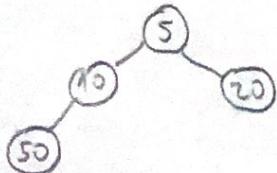
② Inserção 50



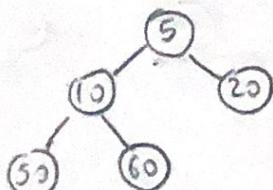
③ Inserção 20



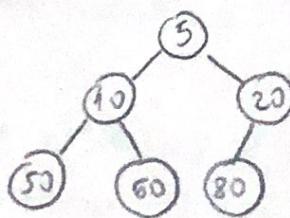
④ Inserção 5
(raiz até a raiz)



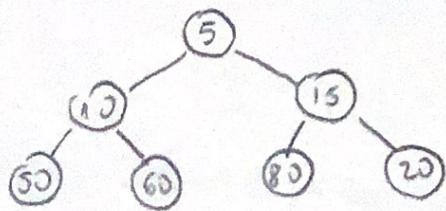
⑤ Inserção 60



⑥ Inserção 80



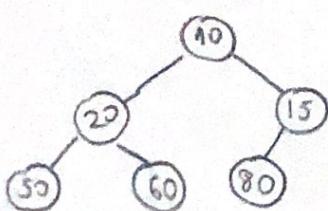
⑦ Inserção 15
(troca com 20)



(c) Comejamos trabalhando com uma min heap, \Rightarrow heap será
descrescente.

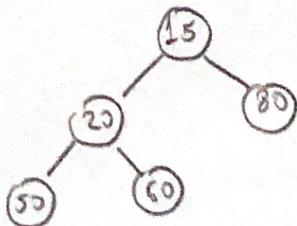
① Sequência atualizada: 5 - 10 - 15 - 50 - 60 - 80 - 20 |

Retirando a raiz e reorganizando: 10 - 20 - 15 - 50 - 60 - 80 | 5



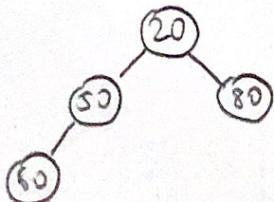
(5 troca com 20 e é inserido,
20 troca com 10).

- ② Retirando a raiz, colocando no inicio do novo array e
reorganizando heap: $15 - 20 - 80 - 50 - 60 \mid 10 - 5$



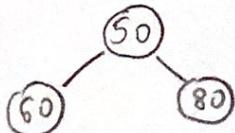
(10 troca com 80 e é removido,
80 troca com 15).

- ③ Retirando a raiz, colocando no inicio do novo array e
reorganizando heap: $20 - 50 - 80 - 60 \mid 15 - 10 - 5$



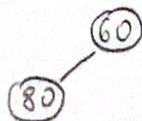
(15 troca com 60 e é removido,
60 troca com 20 e depois com 50).

- ④ Retirando a raiz, colocando no inicio do novo array e
reorganizando heap: $50 - 60 - 80 \mid 20 - 15 - 10 - 5$



(20 troca com 60 e é removido,
60 troca com 50).

- ⑤ Retirando a raiz, colocando no inicio do novo array e
reorganizando heap: $60 - 80 \mid 50 - 20 - 15 - 10 - 5$



(50 troca com 80 e é removido,
80 troca com 60).

⑥ retirando naiz, colocando no início da nova arraiz e
reparando heapt: 80 | 60-50-20-15-10-5

(80)

(60 tira com 80 e é removido)

⑦ retirando naiz, colocando no início da nova arraiz e
finalizando heapt: | 80 - 60-50-20-15-10-5

Sequência decrescente: 80-60-50-20-15-10-5