



Sistem Programlama

DR. ÖĞR. ÜYESİ ABDULLAH SEVİN

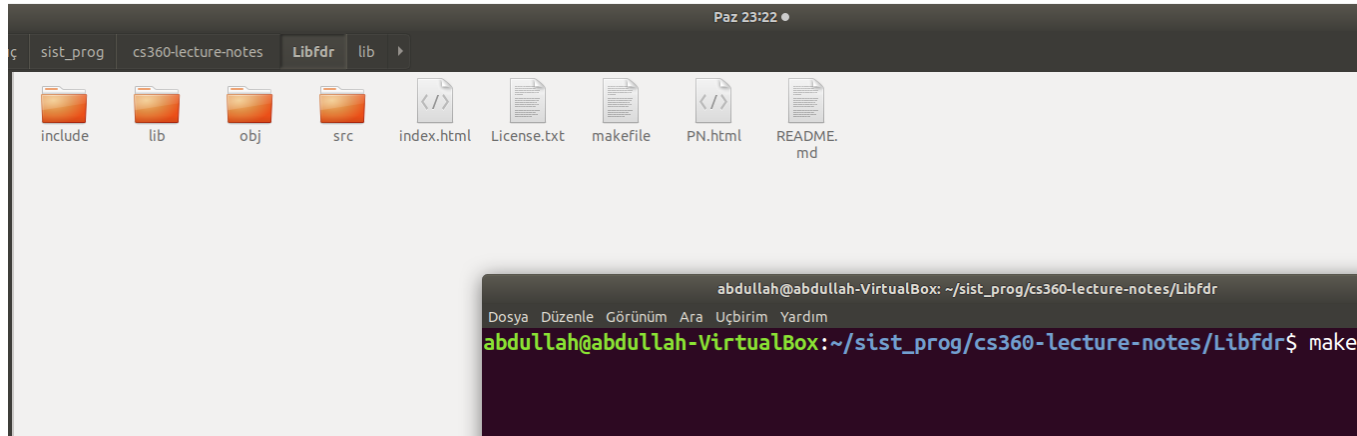
İçerik

a) **Libfdr: Jvals, Fields, Dllists, Red-Black Trees**

- <http://web.eecs.utk.edu/~jplank/plank/classes/cs360/360/notes/Jval/index.html>
- <http://web.eecs.utk.edu/~jplank/plank/classes/cs360/360/notes/Dllists/index.html>
- <http://web.eecs.utk.edu/~jplank/plank/classes/cs360/360/notes/JRB/index.html>

Jvals, Fields, Dllists, Red-Black Trees

- ❑ **Fields** - girdi işlemeyi basitleştiren bir kitaplık
- ❑ **Jvals** - genel bir veri türü
- ❑ **Dllists** - çift yönlü bağlı listeler için bir kitaplık
- ❑ **Red-Black Trees** - kırmızı-siyah ağaçlar için bir kitaplık
- ❑ Kütüphanemizi derlemek için dizinde make yapıyoruz



jval.h

- ❑ Bilmiyorsanız, union bir struct gibidir, ancak aynı anda alanlardan yalnızca birini kullanabilirsiniz.
- ❑ Başka bir deyişle, jv adında bir Jval'iniz varsa, jv.i'yi bir tam sayı olarak ve jv.d'yi bir double olarak ayarlayabilir ve kullanabilirsiniz,
- ❑ ancak ikisini aynı anda kullanamazsınız.

```
typedef union {  
    int i;  
    long l;  
    float f;  
    double d;  
    void *v;  
    char *s;  
    char c;  
    unsigned char uc;  
    short sh;  
    unsigned short ush;  
    unsigned int ui;  
    int iarray[2];  
    float farray[2];  
    char carray[8];  
    unsigned char ucarray[8];  
} Jval;
```

Jval.c

```
/* Libraries for fields, doubly-linked lists and red-black trees.
   Copyright (C) 2018 James S. Plank */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include "jval.h"

Jval JNULL;

Jval new_jval_i(int i) {
    Jval j;
    j.i = i;
    return j;
}

Jval new_jval_l(long l) {
    Jval j;
    j.l = l;
    return j;
}
```


src/simple_jval.c

```
/* This program shows the very simple use of a jval. */

#include <stdio.h>
#include <stdlib.h>
#include "jval.h"

int main()
{
    Jval jv;

    /* Set and use as an integer. */

    jv.i = 10;
    printf("Using the jval as an integer: %d\n", jv.i);

    /* Set and use as a double. */

    jv.d = 5.6;
    printf("Using the jval as a double: %.21f\n", jv.d);

    /* Set and use as a char *. */

    jv.s = "Fred";
    printf("Using the jval as a char *: %s\n", jv.s);

    /* Since I've set it as a (char *), I shouldn't use it as a double. */

    printf("If I now print the jval as a double, I get garbage: %lg\n", jv.d);
    return 0;
}
```

UNIX> **bin/simple_jval**

Using the jval as an integer: 10

Using the jval as a double: 5.60

Using the jval as a char *: Fred

If I now print the jval as a double, I get garbage: 2.13063e-314

UNIX>

specifier character

<i>specifier</i>	Output	Example
d or i	Signed decimal integer	392
u	Unsigned decimal integer	7235
o	Unsigned octal	610
x	Unsigned hexadecimal integer	7fa
X	Unsigned hexadecimal integer (uppercase)	7FA
f	Decimal floating point, lowercase	392.65
F	Decimal floating point, uppercase	392.65
e	Scientific notation (mantissa/exponent), lowercase	3.9265e+2
E	Scientific notation (mantissa/exponent), uppercase	3.9265E+2
g	Use the shortest representation: %e or %f	392.65
G	Use the shortest representation: %E or %F	392.65
a	Hexadecimal floating point, lowercase	-0xc.90fep-2
A	Hexadecimal floating point, uppercase	-0XC.90FEP-2
c	Character	a
s	String of characters	sample
p	Pointer address	b8000000
n	Nothing printed. The corresponding argument must be a pointer to a signed int. The number of characters written so far is stored in the pointed location.	
%	A % followed by another % character will write a single % to the stream.	%

Kurucu metotlar

□ Elbette, basitçe deklare ederek ve kullanarak bir Jval oluşturabilir ve kullanabilirsiniz. Örneğin:

```
Jval j;  
j.i = 4;
```

□ Jval.h, "Kurucu metotlar" için bir sürü prototip tanımlar:

- extern Jval new_jval_i(int);
- extern Jval new_jval_f(float);
- extern Jval new_jval_d(double);
- extern Jval new_jval_v(void *);
- extern Jval new_jval_s(char *);

```
Jval j; j =  
new_jval_i(4);
```


src/jval_ex2.c:

```
/* A program that demonstrates the "constructor" functions, and how they help
   when you are using a procedure that takes a jval as a parameter. */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "fields.h"
#include "jval.h"

/* Print a jval as an integer, double or string. */

void print_jval(Jval jv, char how)
{
    switch (how) {
        case 'i': printf("%d\n", jv.i); break;
        case 'd': printf("%lg\n", jv.d); break;
        case 's': printf("%s\n", jv.s); break;
        default: fprintf(stderr, "Bad 'how' variable in print_jval: %c\n", how);
                    exit(1);
    }
}

/* Call print_jval. Note that I never have to declare a jval, because I'm using
   the constructor functions. */
int main()
{
    print_jval(new_jval_i(10), 'i');
    print_jval(new_jval_d(5.6), 'd');
    print_jval(new_jval_s("Fred"), 's');
    return 0;
}
```

```
UNIX> bin/jval_ex2
10
5.6
Fred
UNIX>
```

Amaç dışı kullanımlar

- ❑ Jval tipinin amacı, dlistler ve red-black ağaçları gibi genel amaçlı veri yapılarını olabildiğince esnek ve verimli hale getirmektir.
- ❑ Jval'leri başka bir nedenle kodunuzda kullanmamalısınız.
- ❑ Spesifik olarak, sadece çalıştığı için kodunuzda int yerine Jval kullanın demezsiniz.
- ❑ Bu, kodunuzu okunamaz hale getirir ve okunamazlık büyük ölçüde kötüdür.
- ❑ İşte standart girdideki tüm tamsayıların ortalamasını almak için kötü kod örneği

src/badavg.c

```
/* A program that abuses jvals. It averages integers on standard input. */

#include <stdio.h>
#include <stdlib.h>
#include "jval.h"

int main()
{
    Jval total;      /* You'll note, no types other than Jvals. */
    Jval j;
    Jval n;

    n.i = 0;         /* n and j are always treated as ints. They should be ints. */
    total.i = 0;      /* Total starts as an int */

    while (scanf("%d", &(j.i)) == 1) {
        total.i += j.i;
        n.i++;
    }

    /* And then total is used as a double. You should simply use a double.... */

    total.d = ((double) total.i) / ((double) n.i);
    printf("Average = %lf\n", total.d);
    return 0;
}
```

src/goodavg.c

```
/* The proper version of this program just uses doubles for total and n. It's not
   like there are ints that can't be stored in a double, and you're using floating
   point division, so you need the doubles. */

#include <stdio.h>
#include <stdlib.h>

int main()
{
    double total;
    double n;
    int j;

    n = 0;
    total = 0;

    while (scanf("%d", &j) == 1) {
        total += j;
        n++;
    }

    if (n == 0) exit(1);
    printf("Average = %lf\n", total/n);

    return 0;
}
```

Erişim Fonk. (Accessor functions)

❑ Bir erişim fonksiyonu, bir alana erişmek yerine bir fonk. çağırarak bir Jval'den istenen değeri almanızı sağlar.

❑ `extern int jval_i(Jval);`

❑ `extern long jval_l(Jval);`

❑ `extern float jval_f(Jval);`

❑ `extern double jval_d(Jval);`

❑ `extern void *jval_v(Jval);`

❑ `extern char *jval_s(Jval);`

❑ `extern char jval_c(Jval);`

`jval_i(j)`'yi çağırmak, `j.i`'yi kullanmakla aynıdır.

Doubly Linked Lists

- ❑ Çift yönlü bağlı listeler ("dllists") C'de son derece kullanışlıdır, çünkü ne kadar veriye sahip olduğunuzu bilmediğinizde veri depolamanıza izin verirler.
- ❑ Çift yönlü bağlı listede, her düğümün iki işaretçisi vardır - biri listedeki bir sonraki düğüme ve bir önceki düğüme işaret ederler. Bu sayede;
 - ❑ Listeleri ileri ve geri hareket ettirebilirsiniz.
 - ❑ Bir listede herhangi bir yere kolayca ekleyebilirsiniz.
 - ❑ Düğümleri çok kolay bir şekilde silebilirsiniz.

```
typedef struct dllist {  
    struct dllist *flink;  
    struct dllist *blink;  
    Jval val;  
} *Dllist;
```

dllist.c

```
/* Libraries for fields, doubly-linked lists and red-black trees.
   Copyright (C) 2018 James S. Plank */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include "dllist.h"

/*-----*
 * PROCEDURES FOR MANIPULATING DOUBLY LINKED LISTS
 * Each list contains a sentinal node, so that
 * the first item in list l is l->flink.  If l is
 * empty, then l->flink = l->blink = l.
 *-----*/

Dllist new_dllist()
{
    Dllist d;

    d = (Dllist) malloc (sizeof(struct dllist));
    d->flink = d;
    d->blink = d;
    return d;
}
```


dllist.h

```
/* Libraries for fields, doubly-linked lists and red-black trees.
   Copyright (C) 2018 James S. Plank */

#ifndef _DLLIST_H_
#define _DLLIST_H_

#include "jval.h"

typedef struct dllist {
    struct dllist *flink;
    struct dllist *blink;
    Jval val;
} *Dllist;

extern Dllist new_dllist();
extern void free_dllist(Dllist);

extern void dll_append(Dllist, Jval);
extern void dll_prepend(Dllist, Jval);
extern void dll_insert_b(Dllist, Jval);
extern void dll_insert_a(Dllist, Jval);

extern void dll_delete_node(Dllist);
extern int dll_empty(Dllist);
```

temel işlemler-dllist

Dllist new_dllist(): yeni bir Çift yönlü bağlı liste ayırır ve döndürür.

free_dllist(Dllist l): Listede ayrılan tüm bellekte free() ögesini çağırarak listeyi yok eder. Listenin boş olması gerekmez.

dll_prepend(Dllist l, Jval val): Listenin başına yeni bir düğüm ekler. Bu düğümün değeri val'dir. Dll_prepend()'in dönüş değeri yok.

dll_append(Dllist l, Jval val): Listenin sonuna yeni bir düğüm ekler. Bu düğümün değeri val'dir. Dll_append()'in dönüş değeri yok.

dll_insert_b(Dllist n, Jval val): Belirtilen düğümden hemen önce listeye yeni bir düğüm ekler. Bu düğümün değeri val'dir.

dll_insert_a(Dllist n, Jval val): Belirtilen düğümden hemen sonra listeye yeni bir düğüm ekler. Bu düğümün değeri val'dir.

dll_delete_node(Dllist): Belirtilen düğümü listeden siler. Bunu nöbetçi olarak arama.

int dll_empty(Dllist l): l'nin boş olup olmadığını döndürür.

Diğer işlemler

Dlist dll_nil(Dlist l): Liste için **sıfır (sentinel)** düğümüne bir işaretçi döndürür.

Dlist dll_first(Dlist l): Listedeki ilk düğüme bir işaretçi döndürür.

Dlist dll_last(Dlist l): Listedeki son düğüme bir işaretçi döndürür.

Dlist dll_next(Dlist n): Listede n'den sonraki sonraki düğüme bir işaretçi döndürür.

Dlist dll_prev(Dlist n): Listede n'den önceki önceki düğüme bir işaretçi döndürür.

Jval dll_val(Dlist n): n düğümünün val alanını döndürür.

int dll_delete_node(Dlist n): n düğümünü siler ve serbest bırakır.

2 makro

□ Son olarak, dll listelerinde ileri ve geri geçiş yapmak için iki makro vardır.

```
#define dll_traverse(ptr, list) \  
    for (ptr = (list)->flink; ptr != (list); ptr = ptr->flink)  
#define dll_rtraverse(ptr, list) \  
    for (ptr = (list)->blink; ptr != (list); ptr = ptr->blink)
```

Dlist yapısı

□ dairesel, çift yönlü bağlı bir liste gibidir.

Some ascii art: Here's an empty list `l`, that would be returned from `new_dllist()`:

```

1 -----> |-----|
           |-->| flink -----\
           |-->| blink -----\
           ||   | val = ?      ||
           ||-----||
           \-----/
          \-----/

```

And here's that list after calling `dll_append(l, new_jval_i(3))`; (or `dll_prepend(l, new_jval_i(3))` for that matter).

```

1 -----> |-----| |-----|
           /--> |flink -----> |flink -----\
           | /-> |blink -----> |blink -----\
           ||   |val = ?      |    |val.i = 3   | |
           ||   |-----|       |    |-----|
           ||   |               |    |
           ||   \----->-----/
           \|----->-----/

```

Dlist yapısı

Actually, it makes the drawing cleaner to have the back links go backwards:

```

1 ----->|-----|-----|
      /-->| flink ----->| flink -----\
      | /----- blink |<----- blink |<-\ | | |
      | | val = ?       |           | val.i = 3 | |
      | | -----      |           | -----   |
      | |               |           |             |
      | \-----/      |           |             |
      |                 |           |             |
      \-----/        |           |             |

```

Here's that list after calling `dll_append(l, new_jval_i(5));`:

```

1 ----->|-----| |-----| |-----|
   /-->| flink ----->| flink ----->| flink -----\
   | /----- blink |<----- blink |<----- blink |<-\
   | | val = ?      |          | val.i = 3    |          | val.i = 5 |
   | |-----|      |          |-----|      |-----|
   | \-----/
   |
   \-----/

```

src/dllreverse.c

```
/* Printing the lines of standard input in reverse order, by appending
   to a dllist, and then printing the dllist backward. */

#include <stdio.h>
#include <string.h>
#include "fields.h"
#include "dllist.h"

int main()
{
    IS is;
    Dllist l;
    Dllist tmp;

    is = new_inputstruct(NULL);
    l = new_dllist();

    while (get_line(is) >= 0) {
        dll_append(l, new_jval_s(strdup(is->text1)));
    }

    /* Notice how convenient this macro is! */

    dll_rtraverse(tmp, l) printf("%s", tmp->val.s);
    return 0;
}
```


src/dllreverse.c

```
abduallah@abduallah-VirtualBox: ~/sist_prog/cs360-lecture-notes/Dllists/bin
Dosya Düzenle Görünüm Ara Uçbirim Yardım
abduallah@abduallah-VirtualBox:~/sist_prog/cs360-lecture-notes/Dllists/bin$ ./dllr
everse
1
2
3
4
5
5
4
3
2
1
abduallah@abduallah-VirtualBox:~/sist_prog/cs360-lecture-notes/Dllists/bin$
```

```
abduallah@abduallah-VirtualBox: ~/sist_prog/cs360-lecture-notes/Dllists/bin
Dosya Düzenle Görünüm Ara Uçbirim Yardım
abduallah@abduallah-VirtualBox:~/sist_prog/cs360-lecture-notes/Dllists/bin$ ./dllr
everse < ../README.md

kept up to date.
you'll have to [read them from my UTK web site](http://web.eecs.utk.edu/~plank/p
lank/classes/cs360/360/notes/Dllists/index.html). I store them here on the repo
, so the site at UTK is
My lecture notes are in HTML rather than Markdown, so to read them,

James S. Plank

# Dllists: Doubly-Linked Lists in C
abduallah@abduallah-VirtualBox:~/sist_prog/cs360-lecture-notes/Dllists/bin$
```

src/dlltail.c.

□ standart girdinin son n satırını yazdırmak. Bunu bir Dllist'e standart girişi okuyarak ve Dllist'in her zaman en fazla n düğüme sahip olduğundan emin olarak yaparız. Sonra çıktısını alıyoruz.

```
/* This program implements tail using a dllist. It reads lines from stdin until it contains the proper number of lines. Then, each time it reads a line to the dllist and then deletes the first line of the dllist. Finally, it prints the dllist. */

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "fields.h"
#include "dllist.h"

int main(int argc, char **argv)
{
    IS is;
    int n;
    Dllist l;
    Dllist tmp;

    /* Error check and read the number of lines to maintain. */

    if (argc != 2) {
        fprintf(stderr, "usage: dlltail n\n");
        exit(1);
    }
    n = atoi(argv[1]);
    if (n < 0) {
        fprintf(stderr, "usage: dlltail n -- n must be >= 0\n");
        exit(1);
    }
}
```

```
/* Create the inputstruct and dllist */

is = new_inputstruct(NULL);
l = new_dllist();

/* Read each line and append to the dllist. */

while (get_line(is) >= 0) {
    dll_append(l, new_jval_s(strdup(is->text1)));

    if (is->line > n) {
        /* If the dllist is full, delete the first node. */
        free(l->flink->val.s);
        dll_delete_node(l->flink);
    }
}

/* At the end, print the dllist -- again note the c */

dll_traverse(tmp, l) printf("%s", tmp->val.s);
return 0;
}
```

src/dlltail.c.

- ❑ Bu program hakkında birkaç not. İlk olarak, standart girdiden okuduğunuz string'i kopyalamak için `strdup()`'u çağırmalısınız.
- ❑ Aksi takdirde, `get_line()` dizenin üzerine yazacak ve tüm satırlarınız aynı olacaktır.
- ❑ İkinci olarak, `dll_delete_node()` ögesini çağırdığınızda, düğümü listeden kaldırır ve düğümü serbest bırakır.
- ❑ Ancak, val alanı `malloc()` ile tahsis edilmiş verileri gösteriyorsa, bu durumda bu veriler otomatik olarak serbest bırakılmaz.
- ❑ Bunun nedeni, `dllist` kitaplığının verilerin ne olduğu hakkında hiçbir fikrinin olmamasıdır.
- ❑ Belki serbest bırakılmasını istiyorsunuz, belki de istemiyorsunuz,

COS: Columns of strings and doubles

- ❑ Bu program kelimeleri okur ve her kelimeyi iki listeden birine koyar.
- ❑ Kelime double ise, onu bir listeye koyar. Değilse, diğer listeye koyar.
- ❑ Ardından, iki listeyi biçimlendirilmiş olarak iki sütun halinde yazdırır.

```
UNIX> cat txt/input1.txt
1 Fred 2 3 Binky Dontonio
UNIX> bin/cos < txt/input1.txt
Fred                                1.0000
Binky                              2.0000
Dontonio                           3.0000
UNIX> cat txt/input2.txt
1 2 3 Fred
UNIX> cos < txt/input2.txt
Fred                                1.0000
                                   2.0000
                                   3.0000

UNIX> cat txt/input3.txt
1 2 Fred Binky Dontonio
UNIX> bin/cos < txt/input3.txt
Fred                                1.0000
Binky                              2.0000
Dontonio
UNIX> bin/cos < txt/input3.txt | cat -e
Fred                                1.0000$
Binky                              2.0000$
Dontonio                            $
UNIX>
```

Uygulama ayrıntıları

```
Dllist new_dllist()
{
    Dllist d;

    d = (Dllist) malloc (sizeof(struct dllist));
    d->flink = d;
    d->blink = d;
    return d;
}
```

```
dll_empty(Dllist l)
{
    return (l->flink == l);
}
```

```
free_dllist(Dllist l)
{
    while (!dll_empty(l)) {
        dll_delete_node(dll_first(l));
    }
    free(l);
}
```

```
free_dllist(Dllist l)
{
    while (!dll_empty(l)) {
        dll_delete_node(dll_first(l));
    }
    free(l);
}
```

```
dll_insert_b(Dllist node, Jval v)    /* Inserts before a given node */
{
    Dllist new;

    new = (Dllist) malloc (sizeof(struct dllist));
    new->val = v;

    new->flink = node;
    new->blink = node->blink;
    new->flink->blink = new;
    new->blink->flink = new;
}
```

Uygulama ayrıntıları

```
dll_insert_a(Dllist n, Jval val)      /* Inserts after a given node */
{
    dll_insert_b(n->flink, val);
}

dll_append(Dllist l, Jval val)        /* Inserts at the end of the list */
{
    dll_insert_b(l, val);
}

dll_prepend(Dllist l, Jval val)       /* Inserts at the beginning of the list */
{
    dll_insert_b(l->flink, val);
}

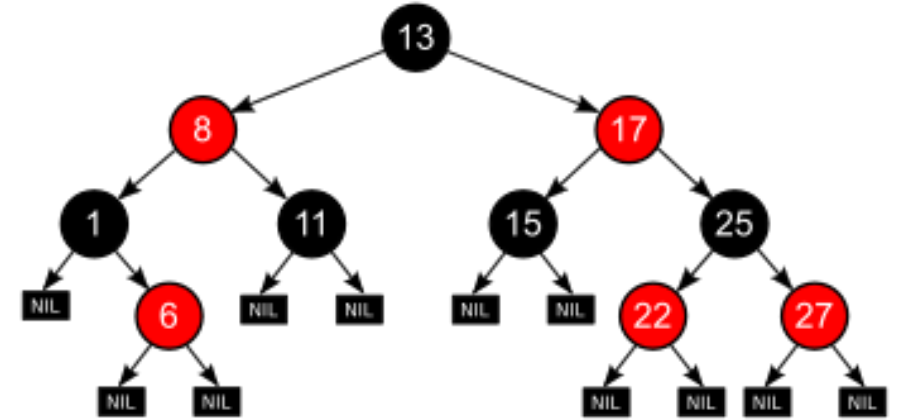
free_dllist dll_delete_node(Dllist node) /* Deletes an arbitrary item */
{
    while (!d {
        node->flink->blink = node->blink;
        dll_del node->blink->flink = node->flink;
    }
    free(l); free(node);
}
```

Red-Black Trees

- ❑ Rb-ağaçları, dengeli ikili ağaçlara dayalı veri yapılarıdır.
- ❑ Nasıl çalıştıklarını bilmenize gerek yok -- sadece işe yaradıklarını ve tüm işlemlerin $O(\log(n))$ zamanında olduğunu, burada n 'nin ağaçtaki eleman sayısı olduğunu bilmek yeterli.

Red-Black Trees

1. Her düğümün, her biri kırmızı veya siyah renkli iki çocuğu vardır.
2. Her ağaç yaprağı düğümü siyah renklidir.
3. Her kırmızı düğümün her iki çocuğu da siyah renklidir.
4. Kökten bir ağaç yaprağına giden her yol, aynı sayıda ("siyah yükseklik") siyah düğüm içerir



Red-Black Trees

- ❑ **JRB jrb_insert_str(JRB ağacı, char *key, Jval val):** anahtar olarak standart bir karakter dizisi kullanarak ağaca yeni bir düğüm ekleyin.
- ❑ **JRB jrb_insert_int(JRB ağacı, int anahtarı, Jval val):** anahtar olarak bir tamsayı kullanarak ağaca ekleyin.
- ❑ **JRB jrb_insert_dbl(JRB ağacı, çift anahtar, Jval val):** double anahtar olarak kullanarak ağaca ekleyin.
- ❑ **JRB jrb_insert_gen(JRB ağacı, Jval anahtarı, Jval val, int (*func)(Jval, Jval))** Şimdi, anahtarınız bir jval. İki Jval'i bağımsız değişken olarak alan ve şunu döndüren bir karşılaştırma işlevi func() sağlarsınız:
 - ❑ ilk anahtar ikinciden küçükse negatif bir tamsayı.
 - ❑ ilk anahtar ikinciden büyükse pozitif bir tamsayı.
 - ❑ anahtarlar eşitse sıfır.

Örnekler

Example programs:

- [src/strsort.c](#): Uses red-black trees to sort standard input lexicographically.
- [src/strrsort1.c](#): Uses red-black trees to sort standard input lexicographically in reverse order. It does this by traversing the tree in reverse order.
- [src/strrsort2.c](#): Uses red-black trees to sort standard input lexicographically in reverse order. It does this by creating a new comparison function **revcomp**, which simply returns **-strcmp()**. Now the tree sorts in reverse order, so it is traversed in the forward direction.
- [src/strusort.c](#): Uses red-black trees to sort standard input lexicographically, and it removes duplicate lines. It does this by checking for a line before inserting it into the tree.
- [src/strisort.c](#): Uses red-black trees to sort standard input lexicographically, ignoring upper and lower case. It does this by creating a separate key, which is the string in upper case. It sorts with that string, and stores the original string in the **val**.
- [src/nsort.c](#): Uses red-black trees to sort like **sort -n --** i.e. it treats each line as an integer, and sorts it that way. If the lines are not integers, or there are duplicate lines, anything goes.
- [src/nsort2.c](#): Uses red-black trees to sort like **sort -n** only now if two lines have the same **atoi()** value, then they are sorted lexicographically. This uses **jrb_insert_gen()**.
- [src/nsort3.c](#): Same as **nsort2**, but instead it uses a two-level red-black tree. If this is confusing to you, please read the next example, which does a very similar thing.

A two-level tree example

```
UNIX> head -n 10 data/input-nn.txt
```

```
    Molly Skyward          60
Taylor  Becloud           47
    Brody  Hysteresis      56
Tristan Covenant          75
Adam    Dyeing             38
Brianna Domain            54
    Jonathan Value         5
        Max Head           48
Adam    Bobbie             68
        Jack Indescribable  99
UNIX>
```

src/ni_sort1.c.

```
UNIX> bin/ni_sort1 < data/input-nn.txt > data/output-1.txt
```

```
UNIX> wc data/input-nn.txt data/output-1.txt
```

```
500 1583 24446 data/input-nn.txt
```

```
500 1583 22000 data/output-1.txt
```

```
1000 3166 46446 total
```

```
UNIX>
```

*wc(Word count)

```
UNIX> head data/output-1.txt
```

Addison Paige Chain	0
Eli Gneiss	0
Ella Craftsperson	0
Lilly Gianna Zen	0
Matthew Stiffen	0
Evan Boorish	1
Isaiah Metabolism	1
Mason Fourier	1
Xavier Agave	1
Daniel Berman	2

```
UNIX> tail data/output-1.txt
```

Layla Option	96
Lucas Fay Jr	96
Madeline Task	96
Sofia Nitrous	96
Gianna Sinh	97
Lucy Quaternary	97
Sophia Contrariety	97
Charlie Lucas Vine	98
Jack Indescribable	99
Lily Span	99

```
UNIX>
```

örnek

❑ **grep** sorgu dosya

❑ sorgu – aradığınız kelime

❑ dosya – sorguyu aradığınız dosya

❑ **Sed** bir dosyadan veya standart girdiden(klavyeden) bilgi okur ve standart çıktıya (ekrana) okuduğu bilgileri kullanıcının belirlediği düzene sokarak yazar

```
UNIX> grep 96 data/output-1.txt
Alexander Bstj          96
Grace Globulin          96
Jonathan Blanket Esq    96
Kaitlyn Thwack          96
Layla Option            96
Lucas Fay Jr            96
Madeline Task           96
Sofia Nitrous           96
UNIX> grep 96 data/input-nn.txt
Grace Globulin          96
Jonathan Blanket Esq    96
Alexander Bstj          96
Madeline Task           96
Layla Option            96
Kaitlyn Thwack          96
Sofia Nitrous           96
Lucas Fay Jr            96
UNIX> grep 96 data/input-nn.txt | sed 's/^ *//' | sort
Alexander Bstj          96
Grace Globulin          96
Jonathan Blanket Esq    96
Kaitlyn Thwack          96
Layla Option            96
Lucas Fay Jr            96
Madeline Task           96
Sofia Nitrous           96
UNIX>
```

örnek

```
UNIX> bin/ni_sort2 < data/input-nn.txt | head
Addison Paige Chain      0
Eli Gneiss               0
Ella Craftsperson        0
Lilly Gianna Zen         0
Matthew Stiffen          0
Evan Boorish             1
Isaiah Metabolism        1
Mason Fourier            1
Xavier Agave             1
Daniel Berman            2
UNIX> bin/ni_sort2 < data/input-nn.txt > output-2.txt
UNIX> openssl md5 output-*.txt
MD5(output-1.txt)= 4eee1503231b23c0052d9b3c57b1cd50
MD5(output-2.txt)= 4eee1503231b23c0052d9b3c57b1cd50
UNIX>
```

```
UNIX> bin/ni_sort3 < data/input-nn.txt | head
Addison Paige Chain      0
Eli Gneiss               0
Ella Craftsperson        0
Lilly Gianna Zen         0
Matthew Stiffen          0
Evan Boorish             1
Isaiah Metabolism        1
Mason Fourier            1
Xavier Agave             1
Daniel Berman            2
UNIX> bin/ni_sort3 < data/input-nn.txt > output-3.txt
UNIX> openssl md5 output-*.txt
MD5(output-1.txt)= 4eee1503231b23c0052d9b3c57b1cd50
MD5(output-2.txt)= 4eee1503231b23c0052d9b3c57b1cd50
MD5(output-3.txt)= 4eee1503231b23c0052d9b3c57b1cd50
UNIX>
```

*openssl, bilgisayar ağları üzerinden iletişimi daha güvenli hale getiren bir kriptografi yazılım kütüphanesi.

Örnek: Golf

Jose Maria Olazabal	-1 F -8
Davis Love III	-1 F -6
Greg Norman	+1 F -5

and [data/score2.txt](#):

Greg Norman	+1 F +9
David Frost	+3 F +10
Davis Love III	-2 F +11

The **golf** program reads in these two files, and ranks the four players by number of tournaments, and then average score:

```
UNIX> bin/golf score1.txt score2.txt
Greg Norman          :  2 tournaments :    2.00
  -5 : score1.txt
   9 : score2.txt
Davis Love III       :  2 tournaments :    2.50
  -6 : score1.txt
  11 : score2.txt
Jose Maria Olazabal  :  1 tournament  :   -8.00
  -8 : score1.txt
David Frost          :  1 tournament  :   10.00
  10 : score2.txt|
```