ساختما ن داده ها

سرفصل ها

۱-مفاهیم

۲-انواع داده ها

۳-آرایه ها

۴-صف

۵-درخت

۶-گراف

ساختمان داده ها تعریف

ساختمان داده ها به نحوه قرارگیری داده ها به حافظه مربوط می شود.

ما در این درس چی دما ن داده ها در حافظه را بررسی خواهیم کرد .

چیدمان داده هادر حافظه به عملیاتی که روی داده ها می خواهیم ا نجام دهیم تاثیر گذار می باشد. ما داده ها را در حافظه قرار می دهیم که عملیات را روی انها انجام می دهیم.

یعضی از چیدمان برخی عملیات ها مناسب است و برخی عملیات ها مناسب نمی باشد

مثال : دو ساختمان در نظر می گیریم که ده طبقه باشد وخانه یکی از انها --

چیده شده باشد.با ماشین طبقه دهم به سادگی تردد کرد .ولی ایرادی که به این ساختمان وارد است از نظر ایمنی یعنی دزدی وسایر خطرات به اسانی انجام میگیرد در صورتیکه نحوه چیدمان خانه ها در ساختمان برخی از عملیات مناسب ودر برخی عملیات مناسب نمی باشد.

در ساختمان هایی که با ماشین به تمامی خانه های ان می توان مراجعه کرد منا سب است برای انجام کمک رسانی در مواقعی کمک رسانی در موارد ضروری وهمچنین دسترسی اسان و تردد اسانتر از ویژگی های ان می باشدوایرادی برای این ساختمان موجود است از نظر ایمنی دزدی وپارهای از مسائل نیز به سادگی در این ساختمان می توان انجام داد

در صورتیکه در ساختمانی که فقط تردد از طریق راه پله انجام میگیرد امکان دزدی بسیار پایین بوده ودر عوض دسترسی به خانه ها مشکل می باشد کمک رسانی در مواقع ضروری به سختی انجا م می گیرد .

در سادر ساختمانی که با ماشین تا طبقه دهم تردد بکنیم بسیاری از فضا ها را ما-جهت تردد اختصاص می دهیم . یعنی از محل استراحت کاسته ولی در عوض ---امکانات فراهم اورده ایم که دسترسی به خانه ها سریعتر واسانتر شده است .

هزینه وامکان را داده ایم و در عوض سرعت وراحتی را به دست اورده ایم .در د

رس ساختمان داده ها به نحوه چید ن داده ها در حافظه متمر کز می شود

انواع داده – هر داده دارای نوع خاص می باشد.

مثال : از زبان c int b ; char a;

در جدول زیر انواع داده در زبان c نمایش داده می شود.

نوع	فول	رنج
char	**۱بایت = ۸ بیت**	**۱۲۷-۱۲۸**
un signd char.	**۸ بیت**	**۰—۲۵۵**
int.	**۲ بایت = بیت ۱۶**	**۳۲۶۷۸-۳۲۶۷۷**

برای دسترسی با اشنا یی با انواع داده ها در زبان c از طریق فشار دادن کلیدهای شیفت f1 وتایپ int وانتخاب datatip (prt-sysrq) جدول نمایش داده می شود.

ادرس ۲۰۰ بگیریم دویست را نشان میدهد char

c=200

۲۰۰ را در نظر بگیریم ۵۶ را نشان می دهد نه ۲۰۰ را هر عدد بنویسیم از ۲۵۶ کم میکند و برای ما نشان می دهد

هر نوع داده دارای رنج مخصوص به خود می باشد و بایستی در هنگام استفاده از انها رنج انها را در نظر بگیریم

مثال : وقتی تعریف میکنیم کاراکتر a یک داده از نوع کاراکتر می باشد که می توان در ان ۱۲۷ تا ۱۲۸ را ذخیره کرد حال اگر در a عدد ۲۰۰ را قرار دهیم

a=200 دیگر ۲۰۰ در داخل a نخواهد بود.این عمل را با نوشتن برنامه pp1 در

مثال زیر نشان داده ایم

```
<include<stdio.h#
<include<conio.h#
()void main
{
```

```
;()clrscr
;char  b=201
;(printf("Adress is %x ",&b
;(printf("data is %d ",b
getch ( ) ;
}
```

در این مثال مقدار ۲۰۰ در رنج داده از نوع کاراکتر نمی باشد به همین دلیل با اجرای این برنامه می بینیم مقدار ۵۶ به a نسبت داده می شود.

پس اگر داده ها را طوری نسبت دهیم که در رنج نوع داده قرار نگیرد نتیجه درستی دریافت نخواهیم کرد

با نوشتن این دستور int a=205 مقدار ۲۰۵ در متغیر a قرار می گیرد. a یک متغیر از نوع عدد صحیح می باشد که در ان مقدار ۲۰۵ قرار دارده شده طول a دو بایت می باشد مقدار a در یک قسمت از حافظه با ادرس مشخص قرار داده میشود که هنگام مراجعه ونیاز داشت برای سادگی کار در زبان های بالا به جای استفاده از ادرس انها استفاده می کنیم به عبارت بهتر جهت قرار دادن داده ها در حافظه به جای استفاده از ادر-س حافظه از نام انها استفاده می کنیم در مثال فوق a نام دو خانه از حافظه میباشد که دران یک عدد صحیح ذخیره می کنیم به جای استفاده از دو خانه از نام ان یعنی a استفاده می کنیم.

برنامه ای بنویسید که ادرس یک متغیر

pp2

```
# in clued <stdio.h>
# include <conio .h>
{
Int a = 205;
Print f( " Address of a is %x "&a );
Printf ("data is %d",a);
Getch ( ) ;
}
```

جاپ میشود خروجی برنامه Address of a is fff4 data of a is 205

برنامه مجموعه ای از دستوراتی است که روی داده ها اعمال می شود تا خواست کار بربر اورده شود

برنامه = الگوریتم + داده ا لگوریتم ها بر روی داده ها اثر می گذارد داده را پردازش می کند

برنامه : ا لگوریتم هایی هستند که روی یک سری داده ها پردازش انجام واطلاعات مورد نیاز کاربر را تحویل می دهد

فرض کنید که یک ما شین طوری طراحی شده است که می تواند دستورات موجود در جدول زبان ماشین زیر را انجام دهد

از حافظه خوانده در محل ثبات A قرار می دهد	ID	0000
A→		
از حافظه خوانده در محل ثبات B قرار می ده B	ID	0100
A+B را در A قرار می دهد	And	0010
A-B را در A قرار می دهد	sub	0010
Out→ A	out	1110
Stop	HLT	1111

فورمت دستورات

کد عمل	ا درس

0000 1100

LD A 12 A→M [12]

در دستوراتِ sub, add ۱=۱ نیاز به ادرس ندارد

فرض کنید در خانه ۱۲ حافظه عدد ۱۵ و در خانه ۱۴ حافظه ۳۰ وجود دارد برنامه ای بنویسید که ۱۵+۳۰ ------۱۰pord

M [12]=15
M [14]=30
Port 10→ 30+15
Ld A=12
LDB=14
ADDx
Out 10
HLT

برنامه ای که ترجمه کردیم روی ادرس صفر قرار می دهیم

ادرس	محتویات	
0	0000	1100
1	0100	1110
2	0010	****
3	1110	1010
5	1111	****
.		
.		
12	0000	1111
13	****	1111
14	0001	111 0

برنامه به زبان ماشین

0000 1100
0100 1110
0010 ****
1110 1010

1111 ****

معادل این برنامه به زبان C #

```
include <stdio.h>
#include <conio.h>
Void main ( )
{
Clrscr;
int a=15
Int b= 30
Printf ("a+b =%d",a+b);
Getch( );
}
```

ارایه

به مجموعه داده های هم نوع گفته می شودگه یک یک در حافظه قرار داده می شودو دارای نام مشترک می باشد این داده ها با اندیس [ ] با شماره از هم د یگر متمایز می شوند. به عنوان مثال در زبان c `char a [ 10];` از نوع کاراکتر تعریف می کنیم که با اندیس های ۹ تا ۰ مشخص می شوند .یعنی اولین عنصر در صفر[0] a دومین عنصردر[1] a و نهمین عنصردر a[9]

A[0] a [1] a[2] a[9]

فرض کنید ادرس اولین در ۱۰۰۰ باشد دومین ۱۰۰۱ وسومین در ۱۰۰۲ ..... برنامه ای بنویسید که ادرس خانه های ا رایه a[10] در مانیتور چاپ شود

pp4

```
Include <stdio.h>
Void main( )
{
 clrscr ( ) ;
Int I;
Char a[10];
For(i=0;i<10;i++)
Printf("\address of a[ I %d ]is %x".& a[ I ] );
Geth( )
 }
```

Ffec 12 ffed 13 ffee 14 ffef 15 fff0 fff1.

برنامه pp5 را بنویسید که ادرس ا رایه از نوع intjer را در مانیتور چاپ کند.

```
  Pp5
In clued <stdio.h>
Void main(  )
{
Clrscr ( )
Int I;
char  a[10];
For(i=0;i<10;i++)
Print f("address of a[%d]is%x".i;&a[i]);
getch ( );
}
```

Pp6 را طوری بنویسید که ادرس از نوع اعشاری در مانیتور چاپ کند.

```
#include <stdio.h>
#include<conio.h>
Voidmain ( )
{
Clrscr( );
Int I;
Float a[10];
For(i=0;i<10;i++)
Printf("addressofa[%d]is%x,I;&a[ I ] );
Getch ( );
}
```

فرمولی ارائه دهید که بوسیله ان با داشتن ا ولین ادرس اولین خانه ا رایه بتوان ادرس سایر خانه ها را محاسبه کرد

Pp5
I=0, 1,2,3,…..
Address of a [I] =base +i *L)

(طول ونوع داده…..*i) + ادرس شروع= ادرس I مین عضو

طول و نوع داده (2*0) + fff2 = ادرس 0 مین عضو

ادرس چهارمین= Ff f 2 + (4 * 2)= fffA

در بعضی از زبان های برنامه نویسی اندیس ا رایه از یک شروع میشود در این صورت از فرمول زیر جهت محاسبه یا ادرس خانه های یک ارایه استفاده می کنیم

مثال : address of a [I] = base + ((I – 1)*L)

طول ونوع داده * (i- 1)) + ا درس شروع = ادرس i مین عضو

مثال :فرض کنید در یک زبان برنامه نویسی اندیس ها از یگ شروع می شوند از فرمول فوق ادرس خانه های یک ارایه از نوع عدد صحیح را که ادرس شروع انها Ff 00 می باشد به دست اورید

۱- اذرس اولین خانه یکمین خانه ادرس = ff00+(1-1)*2

۲- ادرس پنجمین خانه ادرس پنجمین خانه =ff00+((5-1)*2) = ff08

برنامه pp6 را طوری بنویسید که نمرات یک کلاس ۲۰ نفری را گرفته در یک ارایه قرار دهد و معدل کلاسی را محاسبه نموده در اخرین ا رایه درج نماید سپس محتویات این ارایه به همراه ادرس و نام انها در مانیتور چاپ کند

نام	ادرس	محتویات
Pp6 a[0]	ffc0	16.5

```
# include <stdio.h>
#incLude<conio.h>
Void main()
{
Clrscr();
Int I;
Float a[20],sum
For(i=0;i<20;i++)
Scanf ("%f",a[ i ] );
For(i=0;  i <20; I ++)
Sum=sum+a[ I ];
A[20]=sum / 2
For(i=0
```

ارایه های دو بعدی

در بسیاری از برنامه ها ما نیاز داریم که در جدول ها از ذخیره ونگهداری داده ها " داده هایی که با هم ارتباط دارند استفاده کنیم

مثال: گراف زیر ارتباط ۵شهر را با همدیگر نمایش می دهد از اعدادی که روی بارها نوشته شده اند فاصله بین دو شهر را نشان می دهد

برای ذخیره ونگهداری جداول در کامپیوتر از ا رایه دو بعدی استفاده می کنیم ا رایه دو بعدی مجموعه ای از داده های هم نوع می باشند که به ترتیب وپشت سرهم در حافظه قرار می گیرند ا رایه دو بعدی را به دو ترتیب زیر می باشد

۱- به ترتیب سطری ۲- به ترتیب ستونی

۱- ترتیب سطری : نحوه چید مان داده ها به ترتیب سطری به این ترتیب می باشد که ابتدا سطر اول بعد سطر دوم در حافظه قرار می گیرد

۲- ترتیب ستونی : نحوه جیدمان داده ها به ترتیب ستونی به این ترتیب می باشد که ابتدا ستون اول بعد ستون دوم در حافظه قرار می گیرد.

مثال : داده های موجود در جدول زیر را به ترتیب سطری در ادرس ۱۰۰۰ حافظه قرار دهید

M [1000] = 30	a	b	c
M [1001] = 10	30	10	11
M [1002] = 11	12	40	5
M [1003] = 12	fortrain		
M [1004] = 40			
M [1005] = 5			

برنامه ای بنویسید که ادرس های یک ارایه ۵*۵ را از نوع intejer را در مانیتور نمایش دهید.

```
Pp7
#
#
Void main(0
{
Clrscr( );
Int I,j;
Char aa[ 5 ] [ 5 ]
Int a[5][5];
For(i=0;i<5;i++)
For(j=0;j<5;j++)
Printf("\n addressofa[%d][%d] is %x,I,j& a a [ i ][ j ] );
}
Printf("\n");
}
getch ( );
}
```

مثال: برنامه ای بنو یسید که اعداد اعشاری ۲*۳ را در مانیتور چاپ کند.pp9

برنامه ای به زبان c بنویسید که ادرس دو بعدی برای کاراکتر سطری با بررسی ادرس ها در مثال های فوق ترتیب ذخیره سازی ارایه های دو بعدی در زبان cمشخص کند. فرمول محاسبه ادرس ها دو بعدی ارایه دهید.

#

محاسبه ادرس خانه های ارایه با شروع از اندیس یک و به ترتیب سطری از رابطه با لا حساب می کنیم.

مثال: فرض کنید ارایه ۲*۴ زیر در خافظه به ترتیب سطری قرار داده شده است اندیس های این ارایه از یک شروع می شود اگر ادرس اولین عضو ارایه برابر ۲۰۰۰ باشد ادرس A 2*3 را بدست اورید

```
A  2*L
    N*m
Base=2000                    1.1      3.1      6.1      8.5
L=4                          2.1      1001     7.5      4.5
A 2*3
8=L
J =3
Address= Base+[ (i-1)*m + (j -1)]*L
A2*3  address=2000 + (12-1) *4+13 -11) +4 =204
 2000       1.0
 2004       3.1
 2008       6.1
 2012       8.5
 2016       3001
 2020       1001
 2024       7.5
 2028       4.5
```

```
# include<stdio.h >                    pp8
#  include<conio.h>
()void main
{
Clrscr ( );
;int i,j
int a[5][5 ]
(for(i=0;i<5;i++)
(for(j=0;j<5;j++)
{
printf("Enter distance of %c to %c :",i+65,j+65);
scanf("%d",&a [ I ] [ j ]);
{
Getch( );
}
```

ترتیب سطری

توضیح : اکر اندیس ها از صفر شروع شود رابطه فوق به رابطه زیر تبدیل می شود

A I,j address =Base +[(i*m+j)]*4

فرض کنید ارایه n*m از ادرس شروع Base با طول دادهL به ترتیب ستونی در حافظه نگه داری می شود

فرمولی ارائه دهید که ادرس A i.j را محاسبه کند

1.1	3.1	6.1	8.5
2.1	10.1	7.5	4.5

A I,j address = Base + ((j-1) * n) +(i-1)) *L

مثال: فرض کنید اندیس ۲*۴ از اندیس A 2*4 شروع ۲۰۰۰ با طول ۴ در Base 2000 در حافظه قرار دارد مطلوب است محاسبه A 2*3 به ترتیب ستونی

2000	1.1
2004	2.1
2008	3.1
2012	10.1
2016	6.1
2020	7.5
2024	8.5

2018 c.d A 2*3 Address =2000 + ((3-1) *2)+ (2-1))*4=2020

اگر اند یس ارایه از صفر شروع شود رابطه فوق به صورت زیر خواهد بود

A I,j address =Base +((j*n)+i)) *L

طول ارایه ثابت می باشد.

```
#includ <stdio.h>                pp9
include<conio.h>
void main ( )
{
;int i,j
float a[ 2 ] [  3  ]
;clrscr ( ) ;
for(i=0;i<2;i++ )
for(j=0;j<3;j  ++);
}
printf("\n Adress a[%d][%d] is:%x",i,j,&a[ I ] [ j ]);
printf("\n");
getch ( ) ;
}
```

اشاره گر: اشاره گربه نوعی متغیر گفته می شود یک ادرس را در خود ذخیره می کند. در زبا ن c برای تعریف اشاره گر ا ول نام ا ن را با ستاره شروع می کنیم .

مثال : دستور int* p; متغیرp را از نوع اعداد صحیح وبه صورت اشاره گر تعریف می کند یعنی متغیر p ادرس خانه ای از حافظه را درخود نگه می دارد که در ان خانه یک عدد صحیح می باشد

int* p; int a =2; p=&a;

با نوشتن مجموعه دستورات زیر p به محلی از حافظه اشاره می کند که در ان مقدار ۲ قرار دارد

مثال: برنامه pp10 را طوری یه بنویسید که مقدار یک اشاره گر ومحتویات موجود در خانه ای که p

به ان اشاره میکند نشان دهد

```
#incLude
<stdio.h>
 #in cLude<conio.h>
 Void main ( )
 {
cLrscr( );
 Int*p;
Int a= 10;
P=&a;        & = address
Printf( mgdar  a is %d  address  of is  %x, *p,p);
```

ستاره p* در دستورprintf در واقع مقداری است که p به ان اشاره می کند به عبارت بهترp* مقد اری را که در ادرس

p قرار دارد نشان می دهد p ادرسp* مقدار ان ادرس را نشان می دهد اشاره گر یک ادرس است

مثال: ارایه با طول ثابت ارایه با طول متغیر

در ارایه با طول ثابت تعداد داده ها یی که می خواهیم ذخیره کنیم از اول مشخص بوده ودر حین اجرای برنامه کامپا یلار

مقدار حافظه لازم را برای ا رایه تخصیص می دهد

ولی در ارایه با طول متغیر ابتدا طول داده یا مقدار حافظه مورد نیاز مشخص شده و بعد از مشخص شدن مقدار

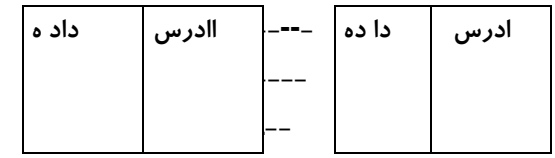
حا فظه مورد نیاز ا ز سیستم را در خواست نموده و این مقدار حافظه را از حافظه اخذ می نماید

```
#in cLude< stdio.h>
#incLude <conio.h
MaLLOC
p   ( int *)  maLLOC  ( SiZe of (mgdar ))
```

مثال: برنامه ای بنویسید که ابتدا تعداد داده مورد نیاز را از کار برسئوال نموده و بعد از گرفتن خانه مورد نیاز به ا ن تعداد داده از کار بر گرفته ودر حافظه ذخیره نماید

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

```
#include<stdio.h>
Void main ( )
{
Int*p
Int I,n
Printf("enter  namber of data");
Scanf("%d",&n);
P=(int*)maLLoc(n*2)
For(i=0; i<n;i++)
{
printf("enter %d , s  namber, I );
Scanf("%d",p[ I ]);
```




```
While(p → next .= NaLL
{
Printf("%d,p ->Data);
P=p →next;
}
```

ارایه با جدول ثابت ومتغیر: ارایه هایی که تا کنون تعریف وبه کار بردیم ارایه با جدول ثابت بودند در کار کردن با این ارایه ها معمو لا دو نوع مشکل وجود دارد یک اگر ارایه با طول **n** تعریف کنیم و نیاز به ذخیره **n+1** ام عنصرباشد ارایه نمی توانیم قرار دهیم به عنوان مثال با تعریف مقصد ۱۰ تا عنصر می توانیم در ارایه قرار دهیم **int a[10]** اگر نیاز به ذخیره ۱۱ ام عنصرباشد نمی توانیم ذخیره کنیم برای حل این مشکل طول ا رایه را می توانیم بزرگ بگیریم در

این صورت مشکل دوم بوجود می اید. اگر طول ارایه را بزرگتر از انچه که مورد نیاز می باشد تعریف بکنیم قسمت هایی از حافظه از دست می دهیم حافظه هرز.

برای رفع این مشکل از ا رایه با طول متغیر استفاده میکنیم برنامه **pp11** یک ارایه با طول متغیر بوجود می ا ورد یعنی

در ابتدای برنامه به تعداد داده ما کاسته می شود سپس با استفاده از **malloc** به اندازه مورد نیاز حافظه در اختیار می گیریم و ادرس شروع ان را در **p** قرار می دهیم

برتامه **pp12** را طوری بنویسید که ابتدا تعداد داده ها را پرسیده سپس یک ارایه با طول همان ایجاد نمائید بعدا به همان تعداد داده ها را دریافت نموده و در ارایه قرار دهید و حاصل جمع داده ها را محاسبه نموده وبه همراه داده ها در مانیتور چاپ کنید

داده ها نوعش **float** باشد **pp12**

```
# include <stdio.h>
#include< conio.h>
# include< stdLib.h>
```

```
Void main ( )
{
Float sum=0;
Float*p,   *first:
Int n,I ;
Clrscr ( );
 print f (" \n enter number  of  Data );
Scan f("%d" &n);
```

ساخت ا رایه با طول n "

```
P= ( float*0 malloc (4*n);
First =p;
 For(i=0;i<n;i++)
{
Printf("\n  enter %d.s Data :",L);
Scanf("%f, p+i);
}
For (i=0;i<n; I++)
{
Printf(" \n%dis of Data is % f "i, *p+ I ));
Sum =+* (p+i);
}
"end offor
Print f ("\n sum is :%f", sum);
}
```

تابع **malloc**

در فایل سرایند **std** تعریف شده وفورمت ان به صورت زیر می باشد

```
( void *) malloc(size);
```

این تابع به اندازه سایز بایت خانه حافظه از سیستم دریافت نموده و ادرس شروع ان را برای ما تحویل می دهد

اگر ما بخواهیم تابعی را تعریف کنیم که مجموع داده های موجود در یک ارایه را توسط یک تابع محاسبه کنیم این کاررا می توانیم به دو روش زیر انجام می دهیم .

۱- هنگام فرا خوانی تابع تمام داده هارا به عنوان ارکومان به تابع بدهیم این تابع مجموع ارگومان ها را به ما بر می گرداند

مثال : برنامه ای بنویسید که مجموع سه تا عنصر موجود در یک ارایه توسط تابع محاسبه نموده ودر تابع ا صلی حاصل جمع ان ها را چاپ کند

```
Pp13
#include <stdio.h>
# include <conio.h>
Void main ( )
Int  sum (int a,intb,int c)
{
Return(a+b+ c);
}
Void main ( )
{
Int a[3 ]={ 2,5.7 };
Clrscr ()
Printf("\n  sum of Deta is : %d" , sum (a[0 }=  a[1 ], a[2],
}
```

تابع را مقدار میدهیم صدا می زند **valio** **ball** **call**

روش دوم

به روش اول روش فرا خوانی با مقدار می گویند روش اول وقتی که مقدار عناصر ارایه زیاد باشد نمی تواند مورد استفاده قرار گیرد . به همین دلیل ما از روش دوم که فراخوانی با ادرس مشهود می باشد استفاده می کنیم call by reference

در این فرا خوانی کافی است که ادرس شروع ومقدار عناصر ارایه را به تابع بدهیم تابع مجموع عناصر را محاسبه کردهو ان را برمی گرداند

```
Pp14
#include < stdio .h>
# include < conio.h>
{
Int  s = o,I ;
For (i=0; i<n; i++)
S=s+ p [I ];
Return ( s );
Main ( )
{
Int  a[10] ={1.10,11,12,2,5,10,3,9,16,} ;
sum =(A,10);
& A[0]
Printf("\n sum  of Data is:%d",sum(&A[0 ],10 ));
```

روش وساختمان داده (ارایه با طول متغیر)

موقعی به درد می خورد که ما ابتدای اجرای برنامه تعداد داده ها را به n ام . در صورتیکه در اکثر کاربردها وبرنامه ها ی کار بردی ما تعداد داده ها را در ابتدای اجرای برنامه نمی دانیم مثلا ثبت نام موبایل پس بایستی نوعی ساختمان داده تعریف کنیم گه هر موقع نیاز داشتیم حافظه از سیستم اخذ نموده وبه داده های قبلی اضافه نماید به این نوع ساختمان داده لیست پیوندی می گویند

لیست پیوندی به صورت زیر می باشد هر کدام از خانه های ان دارای دو تا فیلد می باشد جز اول یا فیلد اول داده و فیلد دوم ادرس خانه بعدی می باشد این خانه ها از طریق فیلد های ادرس پیوندی ارتباط بر قرار می کند به همین دلیل به این ها لیست پیوندی می گوییم

با در دست داشتن ادرس لیست پیوندی می توانیم به تمام خانه های لیست دسترسی داشته باشیم با شروع first به خانه های بعدی وبا next به خانه های بعدی

اخرین خانه دارای ادرس naLLi می باشد ومی توانیم به این طریق انتهای لیست را شناسایی کنیم

قرار دادن اطلاعات در لیست پیوندی را درج کردن می گوییم و ملاقات تک تک عناصر موجود در لیست پیمایش ان می گوییم

```
Pp15
#
#
```

```
#in clued<stdl:h.h>
Stract  L L
{
Int  Data
Stact  L L * next;
};
Stract LL * first
In srt ( )
{
First =( stractLL*)maLLoc( 4);
First→Data = 100;
First →Next =(stractLL*)maLLoc (4);
First→next→ Datd = 200;
First→ Next→Next =null;
Pay ( )
{
Printf("1,s Data )s:%d",first→Data)   ;
```

first

data	next
100	

2

200	NuLL

```
 Printf( "2,s Datais: %d",first→Next → Data);
Main ( )
{
Insert ( );
 Pay()
Getch();
}
```

برنا مه ای بنویسید که نام های عسل وعلی ومریم را در یک لیست پیوندی قرار داده با دریافت ادرس شرو

ع لیست فوق نام های موجود در لیست تا رسیدن به انتهای لیست خوانده ودر مانیتور چـاپ کند

تابع **StrCPY** در فایل سرایند **String.H** وفورمت ان به صورت زیر می باشد

این تابع رشته **s2** را در داخل **s1** قرار داده **string(s1,s2);**

دستور این اشاره گر **p** یک خانه در لیست پیوندی جلو می رود **p = p → next ;**

```
#in cLude<stdio.h>
#in clued < stdio .h >
# in clued < std lib .h>
#in clued < string.h >
Struct  .LL.
{
Char nam [10];struct LL * next
}
Void insert (struct LL *L, char nam [10] )
{
Struct  LL  L1;
While( l  i= null)
L1 =l
L = L → next;   = ( L i* ) malloc (12);
```

```
Strucpy ( L → next → nam, nem);
L →next → next += null;
}
Void pag( struct LL  L)
{
While (L,!= nall)
{
L=L → next;
Print f ("\n %s",L → next) ;
}
" while
}" tab
Void main ( )
{
struct  ll*p;                                    p → next = null;          pp16
Insert( p,"ali");
Insert(p," asall");
Insert (p," ma ryam");
Pag ( p);
}
```

مشخصات یک دانشجو شامل فیلد های نام ؛ نام خانوادگی ؛ شماره های دانشجویی . معدل .

باشد ساختمان داده ای تعریف کنید که به توان این داده ها را با لیست پیوندی مدیریت

struct s ll کرد

```
{
Char nam[10];
Char family [10 ];
Char shomaray [10 ];
Ci out   modal;
Struct s1 L * next  };
```

۱-برنامه ای بنو یسید که تعدادی مشخصات را گرفته در لیست پیوندی قرار دهد

۲-برنامه ای بنویسید که داده های موجود را در این لیست پیوندی در مانیتور چاپ کند

۳-برنامه ای بنو یسید که شاگرد اول را از روی معدل شنا سایی کند و در مانیتور چاپ کند

برنامه ای بنو یسید که تعدادی اسم به دلخواه کاربر یعنی در هر مرحله بعد از گرفتن اسم از کار بر سئوال نماید که ایا نام دیگری می خواهید وارد کنید با نه را از کاربر سئوال نموده وبا وارد نمودن yesیا y نام دیگری را بخواهد ویا تایپ نمودن حرف n وارد نمودن اسم ها خاتمه یافته سپس نام دیگری از شما سئوال نموده و اکر ان نام در داخل نام های قبلی وارد شده وجود داشته با شد پیغام

Yes ,no را به ما بد هد.

تابع insert مثل تابع قبلی می باشد

تابع find

```
Find (struct LL,nam )
{
Struct ll;
While( str cmp(l→( s1,s2 ) nam, nam )!= 0  ) &&L → next i= null)
L=L → next;
If str cmp ( L→ nam,nam )=0
Printf (" yes " ) ;
Else
Printf( " no " )
}
Void main ( )
{
Struct LL;
Char   ch ,   str[ 10 ] , ch = 'y ',
While( ch != 'n' )
Scanf ( " %S ",str );
Insert ( L, str );
Printf ( do you want to contenew )
Ch = getch ( ) ;
}
Scanf ( "%s", str );
Find ( L, str );
Getch ( )
}
```

مثال –برای ذخیره ۳۰ عدد صحیح از یک ارایه ۱۵ تایی استفاده نموده اگر این داده ها را در لیست پیوندی قرلر دهیم مطلوب است

۱- محاسبه حافظه های هرز در ارایه

۲- حافظه هرز در لیست پیوندی

۳- شاخص کارایی حافظه نسبت به ارایه

حافظه ارایه در لیست ارایه ۱۲۰ =۳۰- ۱۵۰ ۱۲۰ *۲ =۲۴۰

۶۰= ۲ *۳۰= حافظه هرز ذر لیست پیوندی

داده	ادرس

۲۴۰ ۰ حافظه هرز ارایه = شاخص کارایی حافظه لیست پیوندی نسبت به ارایه برابر است با

۴= ----------=۰----- حافظه ارایه به لیست

۶۰ حافظه هرز لیست

یعنی در استفاده از ارایه در این مثال ۴ برابر حافظه هرز بیشتر مصرف می شود.

برنامه ای بنو یسید که n تا داده را در یک ارایه قرار دهد و سپس انها را مرتب کند ومرتب شده انها را در ما نیتور چاپ کند.

```
#                                                    #
                #
# defin    n 10
Int  a [n], I,j ;
Void main ( )

{
Daryaft( )
Sort f ( );
Print ( ) ;
}
```

یا بصورت دیگر

```
#
#
# defin       n 10
Int  A[n ], I,j ;
Void daryaft ( )
{
For ( I = 0 ; I < n ; I ++ )
{
Print f ( "\n enter A[ %d ]":i) ;
Scanf( "%d", & A [i]);
}
Void  sort ( )
{
Int   temp ;
For ( i=0; i< n- 1; i++)
For( j= i+1;j<n;j++;)
If (A[ i  ] >A[ j ] )
{
Temp = A [ I ] ;
A [I ] = A [ j ] ;
A [ j ] = temp;
}
}
Void print ( )
{
For ( I =0 ; i<n ; I ++ )
Print f ( " \n  A[%d] is % d" ,i , A [ i  ])  ;
}
Getch ( ) ;
}
```

i	تعداد مقایسه
0	N
1	N – 1
2	N – 2
.	.
.	.
N - 1	1

تعداد مقا یسه در مثال فوق بر حسب I شمار نده

مجموع می شود n/2*n

برای محاسبه تمامی مقایسه ها بایستی ستون دوم جدول فوق را باهم جمع کنیم

مجموع مقایسه ها = n + n- 1+ n – 2 + … + 2 + 1

= En I = n * n/2 = n^2 /2 = 1/2 *n^2

اصطلاحا می گوییم تابع فوق دارای مرتبه ۲(n) می باشد و با نماد 0(n ^2) یعنی از ضریب n^2 صر

ف نظر می کنیم بیشتر برای ما توان مطرح و مهم می باشد.

مرتب سازی حبابی

مرتب سازی حبا بی از مرتبه n^2

برنامه ای بنویسید که اطلاعات مربوط به متقاضیان وام به یک بانگ در یک لیست پیوندی قرار دهد pp19

برنامه اصلی به صورت زیر می باشد ۰۰ (تعداد متقاضی مشخص نیست)

```
MAIN ( )
{
DRYAFT ( ) ;
SORT ( ) ;
PRINT ( ) ;
}
```

که در ان اطتاعات را

دریافت کرده ودر لیست قرار می دهد.

تابع sort اطلاعات را بر اساس نام انها مرتب و تابع print ان اطلاعات را در مانیتور چاپ می کند

مشخصات متقاضی نام ونام خانوادگی مبلغ در خواستی پس در این صورت ساختمان داده استفاده در لیست پیوندی به صورت زیر خواهد بود

```
Struct  vam
{
Char, name [ 10 ];
Char  family [20 ];
Longint   mablag ;
Struct  vam  *next;
}
first ,* p;
```

تابع دریافت

یک کاراکتر تعریف می کنیم

```
daryaf t   ( )
Char ch = 'y';
{
First =( struct van*) mallocsiz of(struct van));
P =first;
While( ch=='y')
{
Printf("\n enter namber:");
Scanf("%s,p→name);
```

```
Printf("\n enter  f name:");
Scanf ("%s, p→f  name);
Printf("\n enter mablag );
Scanf  ("%Ld",p→ mabLag);
Printf("\n  doyou want +enter new data y/n");
Ch=getch ( );
If (ch =='y')
{
p→next=struct van *)maLLOc (size of (struct von ));
p =p → next;
}
"end of while
P → next= naLL;
}
"end  of daryaft
```

تابع **sort** مرتب سازی

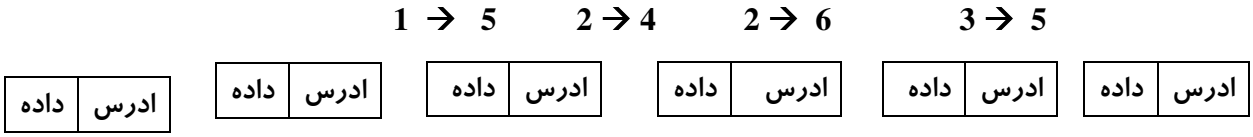
```
Sort ( )
Char temp[20];Longint a;
{
Struct  vam *q;
P = first
While ( p→ next= nuLL)
{
q = p→ next
whiLe ( q→ next % = null)
if ( str cmp (p→ name,  q → name >0 ))
{
Strcpy ( temp, p → neme )
Strc py  ( p → nam, q → nam )
Str c py  ( q → nam, temp ) ;
Strc py  ( temp; p → fname)
Strc  py ( p → fnam ; q → fnam)
Strc  py (  q → fnam , temp ) ;
 A = p → mablag ;
P → mableg  = q → mablag;
Q → ,ablag = a;
}
Q =q → next ;1
}
"end of while q
P = p → next ;
}" end of while q
}
    " end of sort
```

تابع **print** مرتب سازی

```
Print ( )
{
P = first ;
While ( p != null
Print ( "\n name % s  %s\t
Fname,  p → fname
P → mablag ;
P =p → next;
Getch ( )
```

}

حل مسئله از طریق شکل



حل مسئله از طریق شکل

temp = p → next
P → next = q → next
q → next = temp
temp = q → next → next
q = next → next = p → next → next
p → next → next = temp

pp 19 اطلاعات مطرح شده در pp19

می خواهیم توسط ارایه مرتب سازیکنیم بطور متوسطدر هر روز ۳۰ نفر متقاضی به بانگ مراجعه می کند

اگر مثال: pp 19 را با یک ارایه ۳۰۰تا pp پیاده سازی نمائیم شاخص کاررایی رابا لیست پیوندی که ارایه در مثال فوق محاسبه نمائید

محاسیه سایز ساختمان ۱۰ نام + ۲۰ فامیل + ۴ تا برای مبلغ +۲تا ادرس

حافظه هرز در لیست ۶۰= ۲*۳۰

حافظه هرز در ارایه ۹۱۸۰= ۳۴*۲۷۰

شاخص کاررایی حافظه لیست ---------- = ۹۱۸۰

نسبت به ارایه ----------۱۵۳ = ۶۰

یعنی لیست پیوندی در مثال فوق برابر ۱۵۳ برابر حافظه هرز کمتری دارد

اگر دو مثال فوق را باهم مقایسه کنیم اگر روزی قرار باشد ۳۰۲ نفر به بانگ مراجعه کند برنامه ای که با ارایه نوشته شده است ثبت ۳۰۱ مین نفر عاجز خواهد بود در صورتیکه لیست پیوندی به کار خودش ادامه خواهد داد

Serch

ملا قات عناصر موجود در حافظه جهت پیدا کردن عنصر مورد نظر را جستجو می گو ئیم در این جلسه دو نوع جستجو معرفی خواهیم کرد.

جستجوی خطی : جستجوی دو دویی یا باینری

جستجوی خطی را در تمام داده ها می توان انجام داد

در صورتیکه جستجوی باینری را فقط در روی داده های مرتب شده می توان انجام داد قبل از وارد شدن به این بحث بهتر است سا ختمان ریاضی درخت را معرفی نمائیم

ساختمان درخت

درخت به نوعی ساختمان ریاضی گفته می شود که دارای ۲ بچه از نوع خودش باشد
عناصری که بچه نداشته باشد برگ و عنصری که پدر نداشته باشد ریشه می گوئیم .
درختی که هر کدام از عناصر ان دو بچه داشته باشد به ان درخت دو دویی می گوئیم
اگر فاصله همه برگ ها در ریشه برابر باشد درخت حاصل را درخت دودویی کامل می گوئیم
خصوصیات این درخت
ارتفاع یا عمق درخت (در شکل زیر یک درخت کامل دودویی و عمق درخت کامل را نشان می دهد.) فاصله تمام بندها تا ریشه برابر دو می با شد
اگر عمق ۲ باشد تعداد برگ 2n هست (دو به توان n است)
تعداد گره های غیر برگ 2n-1 هست (دو به توان ان منهای یک است)
تعداد کل گره ها 1-- 2n + 1 (دو به توان n بعلاوه یک ومنهای یک هست) m+1
اگر تعداد گر هها در یک درخت m باشد عمق از رابطه زیر بدست می اید Log 2 _1
۲به توان ۳ = ۸ ۸= ۱+ ۷

در جدول زیر مشخصات درختهای دو دویی کامل درج شده است

Log 2 = 5
۲ به توان ۵ =۳۲
۵—۱ =۴

مجموع گرهها	تعداد گرههای برگ	تعداد گرههای غیر برگ	
۷	۴	۳	۲
۱۵	۸	۷	۳
۳۱	۱۶	۱۵	۴
			۵

جستجوی خطی

در جستجوی خطی تک تک عناصر را جهت پیدا کردن عنصر مورد نظر ملا قا ت می کنیم
مثال: برنا مه ای بنو یسید که با استفاده از جستجوی خطی در ارایه یک عنصر مورد نظر را پیدا کند. serch ( )

```
{
Int  a[ 10 ] =       { 1,2.12.20, 50.60 }
Int  I, j ;
Printf ("\n  enter a namber for  serch :");
Scan f ("%d",& j);
For ( i=0;  i<10 ;  I ++)
If ( a [ I ] == j )
{
Print f ("namber is %d is in %d  s",  I,I );
Break;
}
}
```

در برنامه فوق چند تا مقایسه صورت می گیرد بدست اورید

ممکن است بگوئید اولین عدد

مجموع مقایسه ها برابر است با n +۲+۱ است

متوسط مقایسه ها برابر است با

$$\frac{\text{مجموع}}{n} = \frac{n\_\_2 \ o\ (\ n\ )}{n}$$ متوسط مقایسه ها از مرتبه $o\ (n)$ است .

جستجوی خطی را در لیست پیوندی بنویسید که مجموع مقایسه ها و متوسط مقایسه ها را بدست اورید pp۲۳

جستجوی دودویی

معمولا در مورد ارایه های مرتب بکار برده می شود

۱ : ۱۰ : ۱۲ : ۱۴ : ۱۵ : ۳۵ : ۴۰ ؤ۴۵و۶۵

این نوع جستجو را به صورت زیر انجام می دهند

اول داده مورد نظر را با داده وسطی مقایسه می کنند اگر از ان بزر گتر باشد جستجو را در نیمه بالایی انجام می دهند. اگر از ان کوچکتر باشد جستجو را در نیمه پایینی انجام می دهند توجه شود که جستجو در نیمه بالایی و نیمه پایینی نیز با همان عنصر وسطی مقایسه می شود در این نوع جستجو به ازای هر مقایسه نصف داده ها کنار گذاشته می شود

در جدول زیر تعداد ما کسیموم مقایسه ها بر حسب تعداد عنا صر موجود در ارایه ها نوشته شده است

n	ماکذیموم مقایسه
۸	۳
۱۶	۴
۳۲	۵

۶	۶۴
۷	۱۲۸
۸	۲۵۶
۹	۵۱۲
۱۰	۱۰۲۴
۱۱	۲۰۴۸
۱۲	۴۰۹۶

در واقع اگر n تا داده داشته باشیم log 2 n را مقایسه انجام می دهیم

مرتبه جستجو دودویی از مرتبه logn ان بر مبنای ۲ می باشد که با نماد (log 2n) o نشان می دهیم

جستجوی دودویی را فقط در ارایه ها می توان استفاده نمود

جستجوی دودویی در لیست پیوندی اگر مرتب هم باشد نمی توان استفاده نمود .

با توجه به ساختمان لیست پیوندی فقط جستجوی خطی را می توان بکار برد

مثال : برنامه جستجوی دودویی را در ارایه مرتب pp۲۴ بکار ببرید

الگو ریتمی جهت درج یک عنصر جدید در اول لیست پیوندی ارایه نما ئید .

فرض کنید داده های زیر در یک لیست پیوندی قرار دارد می خواهیم یک عنصر جدید در اول لیست بایستی عملیات زیر را به ترتیب انجام دهیم

۱- گرفتن حافظه از سیستم به ادرس p

۲- ایجاد پیوند p به اول لیست p → next = first

۳- قرار دادن عنصر جدید در حافظه اخذ شده : p → Data = " Ahhd "

۴- قرار دادن p به عنوان اولین عنصر ; first = p

ASd	2020	Ali	1030	zinab	NuLL
۱۰۰۰		۲۰۲۰		۱۰۵۰	

Null	

الگوریتمی جهت درج عنصر جدید در انتهای لیست ارایه نمائید

فرض کنید لیست زیر در حافظه قرار دارد جهت درج اطلاعات جدید در انتهای لیست بایستی عملیات زیر به تر تیب انجام دهیم

۱ – پیمایش لیست تا رسیدن به انتها q به اخرین عنصر اشاره می کند

۲ – گرفتن حافظه از سیستم به ادرس p

۳- ایجاد پیوند p به انتهای لیست q → next = p

۴ – قرار دادن عنصر جدید در حافظه p → Data = Ahhd

۵ – قرار دادن p به عنوان اخرین عنصر p → next = Null

zinab	nall

ahhd	Null

جهت درج عنصر جدید در وسط لیست پیوندی ارایه دهید

فرض کنید لیست زیر در حافظه قرار دارد می خواهیم عنصر جدیدی را بعد از عنصر مورد نظر (علی) اضافه کنیم

		null	زینب	۱۰۳۰	علی	۲۰۲۰	اسد

۱ – پیمایش لیست تا پیدا کردن عنصر مورد نظر (p به عنصر مورد نظر اشاره می کند)

۲ – اخذ حافظه جدید به ادرس p

۳- ایجاد پیوند q با عنصر بعدی q → next = p → next

۴- ایجاد پیوند p با q p → next = q

۵ – قرار دادن نام جدید در محل جدید q → data = " ahhd " ;

تمرین

الگوریتم های نوشته شده در همه این برنامه ها را به زبان c پیاده کنید .

۱- الگوریتمی جهت حذف یک عنصر از اول لیست پیوندی ارائه فرمائید .

۱۰۳۰	NULL

۱۰۳۰	

۲۰۲۰	

فرض کنید لیست پیوندی زیر در حافظه قرار دارد می خواهیم اولین عنصر از لیست حذف نمائیم برای این کار اعمال زیر را به ترتیب انجام می دهیم

۱ – قرار دادن ادرس اولین خانه در p p = first ;

۲ – قرار دادن ادرس دومین خانه به عنوان ادرس شروع first = p → next ;

۳ – بر گرداندن حافظه مر بوط به اولین خانه به سیستم first (p) ;

الگوریتمی جهت حذف اخرین عنصر یک لیست پیوندی ارائه فرمائید

فرض کنید لیست بالا در حافظه قرار دارد می خواهیم اخرین عنصر را از حافظه حذف نمائیم

برای این کار باید اعمال زیر را انجام دهیم

۱ – پیمایش لیست تا رسید ن به انتهای ا ن p به ما قبل اخر

۲- ازاد کردن حافظه تخصیص یا فته به اخرین عنصر free (p → next)

۳ – قرار دادن p به عنوان اخرین عنصر لیست p → next = null

الگوریتمی جهت حذف بک عنصر از وسط لیست ارائه فرمائید

فرض کنید لیست زیر در حافظه قرار دارد

null	زینب		احد ۱۰۱۰	۱۰۱۰	علی ۱۰۳۰	۲۰۲۰	اسد ۱۰۰۰

عنصر مورد نظر ahhd را از لیست حذف کنیم

برای این منظور اعمال زیر را به ترتیب انجام می دهیم

۱- پیمایش لیست جهت پیدا کردن عنصر مورد نظر p به عنصر ما قبل عنصر مورد نظر اشاره می کند

۲- قرار دادن ادرس عنصر مورد نظر در q q → next ;

۳- ایجاد پیوند p با عنصر بعدی (حذف عنصر مورد نظر از لیست)

p → next = q→ next

۴ – ازاد کردن حافظه تخصیص یا فته به عنصر مورد نظر free (q)

لیست دایروی با حلقوی

لیست حلقوی به نوعی لیست اطلاق می شود که اخرین عنصر ان به ابتدای لیست اشاره نماید به طوری که در شکل زیر نشان داده می شود

۲۰۲۰		۱۰۱۰		۲۰۲۰		۱۰۰۰	
	زینب		احد	۱۰۱۰	علی	۲۰۲۰	اسد
۱۰۰۰							

ویژه گی های لیست حلقوی

۱- با در دیت داشتن یکی از ادرسهای یکی از خانه های ان می توان به تمام خانه ها ی ان دسترسی پیدا کرد . (یعنی ادرس همه خانه ها به عنوان first استفاده کرد

۲- پیمایش در این لیست از یک ادرس شروع می شود و تا رسیدن به همان ادرس ادامه می یابد

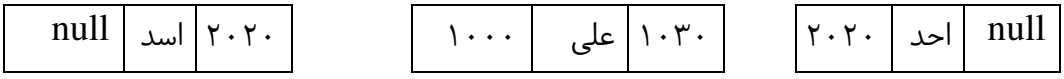
پیمایش در لیست پیوندی همیشه از عنصر اول شروع می شود در صورتیکه در لیست حلقوی پیمایش را می توان از هر عنصری شروع کرد

توضیح : درج کردن در لیست حلقوی مثل درج کردن در وسط لیست پیوندی می با شد ابتدا و انتها ندارد first هم ندارد

لیست دو پیوندی

یکی از ایرا د های لیست تک پیوندی این می باشد که در حین پیما یش نمی توان به عنصر های ما قبل بر گشت برای رفع این عیب ساختمان داده جدید ارائه شده که لیست دو پیوندی نا میده می شود

در این نوع ساختمان دو تا پیوند وجود دارد دو تا ادرس



یک پیوند به عنصر قبلی و یک پیوند به عنصر بعدی اشاره می کند

نحوه تعریف این نوع ساختمان را به زبان c بنویسید

```
Struct   DLL
{
Struct   DLL * pre ;
Char  D ata [20 ] ;
Ftruct    DLL * next;
}
```

تمر ین

برنامه ای به زبان c ارائه دهید که لیست دو پیوندی زیر را ایجاد نموده و اطلا عات مربوط به ان را در او درج نماید

درج در یک لیست دو پیوندی خالی

```
Last    ,first    null
P = (DLL *)  malloc ( size   of ( Dll )
P → Data ;
Last = p ;
```

First = p;
P → next =null
P → pre = null

Null به هیچ جا اشاره ندارد

Pre = قبلی

DLL = لیست دو پیوندی

۱- الگوریتمی جهت درج اطلاعات در ابتدا ی لیست دو پیوندی ارا ئه نمائید

P= (DLL *) maloc (size of (DLL);
P → next , first
First → pre = p ;
First → =p;
First → pre = null;
P = → Data = Data

درج در انتهای لیست

P = (Dll *) maloc (dize of (Dll);
Last → next = p ;
P → pre = Last (Last = Last → next ;
Last = p ;
P → Data = Data
Last → next = null

درج در وسط لیست

P = ('' DLL *) maloc (size of (Dll));
Q → pre → next =p
P → pre =q → pre
P → next = q
Q → pre =p;
P → Data;

حذف از ابتدای لیست

P = first
First = first → next
First → pre = null
Free (p);

حذف از ا نتها ی لیست

P = last;
Last = last → pre
Last → next = null;
Free (p);

حذف از وسط----------

لیست

Q → pre → next = q → next;
Q → next → pre = q → pre;
Free (q) ;

مرتب کردن----------

لیست : پیمایش از انتها

```
DLL * t;
T = LAST;
While ( t != null )
{
Print f ( t → Data );
T = t → pre ;
}
```

پیما یش از ابتدا

```
Dll * t;
T =- first ;
While ( t != null )
{
Print f ( t → Data );
T = t → next ;
}
```

شروع	وارد کردن عدد	بعدی	قبلی	انتها	خارج می شود
first	name	next	pre	Last	NuLL

Pp40

برنا مه ای بنویسید که نام علی و فاطمه را در یک لیست دو پیوندی قرار دهد

```
# in clued < stdio.h >
#include < string.h >
# include < stdlib.h >
Void main ( )
{
Struct dL
{
Struct DL * pre;
Char name [ 20 ] ;
Struct dl * next ;
} ;
Void ins 2 ( )
Struct dl * first , * last;
1- first = ( struct dl * ) malloc (size of ) ; 24----iany
2- first → per = NuLL;
Str cpy ( first → name; "ali " );
4- first → next = ( struct dl * ) malloc ( size of );
5 – first → next → pre = first
6- str cpy (first → next → name," fatmh ");
 7- last = first → next;
8- first →next → next = null;
}
```

Pp41

تابعی بنویسید که با دریافت ادرس شروع یک لیست دو پیوندی و یک نا م را در انتها ی لیست درج نما ید

```
# in clued <stdio.h >
# in clued <string.h>
#include <stdlib.h>
Void ins end (struct dl * first;
Char * name )
{
While (first →next !=null )
First = first ,next
1- first → next= ( struct Dl * ) malloc (size of )
2- first → next → pre = first;
3 – first → next ;
4- first → next → next = null;
5-str cpy first → next → name,name
```

Pp44

برنامه ای بنویسید که با دریافت ادرس انتهای لیست دو پیوندی ان را پیمایش نموده در مانیتور و پرینتر چاپ نماید

```
#
#
Void pay end ( struct dl * last )
{
While ( last  != NULL
Fprinter ( stdout, "\n namr is %S",last →name );
Fprinter ( stdprn,"\n name is %S,last → name );
Last= last →per;
}
```

Pp45

برنامه ای بنویسید که با دریافت ادرس شروع یک لیست دو پیوندی ویک نام ان را در ابتدای لیست درج و در خا تمه ادرس شروع جدید را بر گرداند

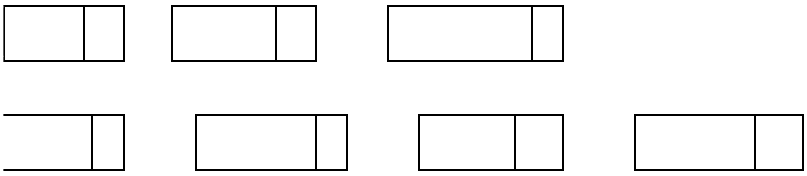
```
#
#
Voidmain ( )
{
Struct DL * ins first ( struct )
{
DL * first  char * name )
First → per = ( struct DL * ) maLLoc (24 );
First → pre → next = first;
First = first → per;
First → per = NULL;
Strcpy ( first → name,name );
Return (first );
```

Pp48

برنامه ای بنویسید که محتوبات موجود در یک لیست را در انتهای لیست دو قرار دهد

فرض کنید ادرس شروع لیست 2 first می باشد

```
#
#
Void (struct  DL * firs 1, struct DL * first )
{
Struct DL * p;
P=first 2;
While ( p → next != NuLL )
P = p → next;
P → next = first 1 ;
}
```



ساختمان داده و پشته یا صف

اکثر در سیستم کامپیوتری و برنامه های کاربردی ساختمان داده های صف پشته مورد نیاز می باشد این نوع ساختمان داده در زیر شرح داده می شود

۱- صف

صف به نوعی ساختمان داده گفته می شود که داده ها در ان به ترتیب ورود به صف پردازش می شوند یعنی اولین داده ای که در صف قرار می گیرد اولین داده ای است که سرویس می گیرد به همین دلیل به این ساختمان داده ساختمان داده fifo می گویند یعنی first , in first out

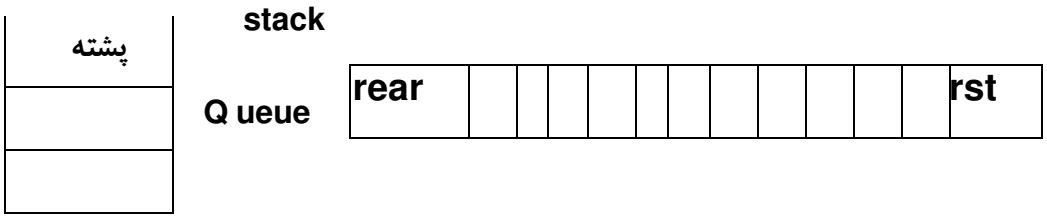
برای هویت این نوع ساختمان داده دو اشاره گریا دو اند یس استفاده می شود که یکی به اول صف اشاره می کند و معمولا front و دومی به انتهای صف اشاره می کند وrear نامیده میشود

۲- پشته

پشته نوعی ساختمان داده می باشد که اطلاعات در ان ساختمان داده به ترتیب عکس ورود انها پردازش می شوند

یعنی اخرین دادهای که وارد پشته می شود اولین دادهای است که پردازش می شود به همین علت به این ساختمان داده LIfo می گویند .Last in first out می گو یند

برای ورود این پشته یک اشاره گر یا یگ اند یس کافی می باشد که معمولا به این نوع داده top می گو یند .



Top به اخرین عنصر وارد شده به پشته اشاره می کند

هر دو ساختمان داده فوق را می توان با ارایه و لیست پیوندی ایجاد و مدیریت کرد

معمولا در ساختمان داده صف جهت سرویس دهی متقاضیان از یک منبع استفاده می کنیم در صورتی که از پشته در مسا ئل مخصوص ریاضی وفراخوانی توابعی استفاده می کنیم

وقتی یک تابع در داخل تابع دیگر فرا خوانی می شود وبه همین ترتیب چند ین تا بع تو در تو فرا خوانی شود ابتدا با یستی اخرین تابعی که فرا خوانی شده است اجرا شود تا به توان تابع قبلی را اجرا نمود

برای مدیریت توابع باز گشتی و توابع تو در تو ما مجبور هستیم از ساختمان داده پشته استفاده نمائیم. در این صورت ادرس های باز گشت توابع در پشته قرار می گیرد چون

این ادرس ها به ترتیب محفوظ مورد استفاده قرار می گیرند

یعنی با یستی اخرین تابع فراخوانی شده اجرا شود تا بتوا نیم نام قبلی را اجرا نمائیم

مثال های زیر این موضوع را مشخص می کند

۱- pp49

برنامه ای بنویسید که کاراکتر ها را یک یک و پشت سر هم و تا وارد نمودن cly enter در ما نیتور چاپ نماید

```
# in clued <stdio.h>
# include <conio.h>
Void main ( )
{
Char ,ch [ 20 ];
Int top → = - 1;
Do
{
Fprinter  F ( std out , "\n enter a char:" );
Top + +;
```

```
Ch [ top ] = getch ) ;
}
While ( ch [ top ] !=13 );
While ( top != _1 )
{
Top _ _;
Put ch (ch [top ] );
}
```

Pp51

توبع باز گشتی و توابع تو در تو توسط پشته پیاده سازی می شود

برنامه pp51را طوری بنویسید که به کمک توابع باز گشتی کلید ها را یک یک پشت سر هم از صفحه کلید دریافت نموده و با وارد نمودن کلید اینتر به صورت معکوس عمل نماید (مثل برنامه ۴۹ یعنی می خواهیم برنامه ای بوسیله توابع باز گشتی بنویسیم pp51

```
void main ( )
{
Char  ch;
Ch = getch ( );
If ( ch != ( 13 )
Main ( ) ;
Put ch ( ch );
}
Void main ( )
{
F ( ) ;
}
```

با مقایسه نتیجه pp51,pp49 متوجه می شویم که توابع باز گشتی حتما توسط پشته پیاده سازی و مد یریت می شوند

توضیح این که تابع mainخود می تواند یک تابع باز گشتی باشد در این صورت می توان برنامه pp52 را نوشت

Pp52

برنامه ای بنویسید که با استفاده از فراخوانی توابع پشته اختصاص داده شده به برنامه را پر نماید

راهنمائی اگر یک تابع باز گشتی بخواهد خودش را بی نهایت فرا خوانی نماید در این صورت پشته اختصاص داده شده به برنامه پر خواهد شد

برنامه ای برای این منظور بنویسید که پر شده پشته را نشان دهد

```
Pp52                                                    void main
( )
{
Printf ( " \n000 End of stack );
Main ( ) ;
```

برنامه **pp54** را طوری بنویسید که در ان برنامه کلید های وارد شده از صفحه کلید را وارد نموده ودر یک صف قرار دهد و سپس به ترتیب ورود در مانیتور چاپ نماید

این برنامه را به کمک ارایه پیاده سازی می کنیم

```
Struct  LL
{
Char  ch ;
Struct LL * next ;
} ;
Void main ( )
{
Struct LL * front,* rear *pfront = ( struct LL * ) malloc ( 3 );
Front = rear= getch ( ) ;
Rear → ch = getch ( );
Do
{
Rear → next = ( struct LL ) malloc ( sizeof ( structll ));
Rear = rear → next;
Rear → ch = getch ( );
{
While ( rear → ch ! = rear )
{
While ( front ! =rear )
{
Putch ( front → ch );
P = front;front = front → next;
Fr ee ( p );
}
}
```

Pp55

برنامه ای بنویسید که پیاده سازی صف توسط ارایه انجام گیرد

```
#
#
Void main ( )
{
Char  Ch [ 20 ] ;
Int rear,front;
Rear = front = 0
,do
[
CH [REAR ] =GETCH ( );
REAR ++;
}
WHILE ( CH [REAR—1] ! =13
WHILE (ENTER !=rear )
{
putch ( ch [ front ] );
```

```
front ++
}
}
```

توضیح این که پیاده سازی صف با استفاده از ارایه عددی با یستی از ان ارایه بصورت حلقوی استفاده شود یعنی پر شدن ارایه صف اطلاعات را از ابتدای ارایه قرار دهد (صف حلقوی) در این صورت **rear= front** می شود اگر صف خالی باشد یا پر باشد یعنی در صف های حلقوی در دو حالت **front = rear** صف پر باشد یا خالی باشد مثل مثال **pp55**

قرار دادن در ساختمان داده را **push** وبر داشتن از ان را **pop** می گویند.

فرض کنید یک رکورد اطلاعا تی به طول ۵۰ توسط ۳ تا ساختمان داده ارایه لیست تک پیوندی یا دو پیوندی پیاده سازی می شود بطور متوسط ۱۰۰ رکورد در برنا مه پردازش می شود انواع الگوریتم های پیچیدگی زمانی و مکانی ساختمان های فوق را بررسی نمائید جهت مدیریت این برنامه با ارایه یک ارایه ۳۰۰ تایی نیاز داریم در این صورت متوسط

حافظه هرز

حافظه هرز در صد ۲۰۰

٪ ۶۶ = -------

۳۰۰

حافظه مفید ۶۶

٪ ۳۳ = ------ -۱ با تک پیوندی

۱۰۰

۲ بایت	۵۰ طول اطلاعات

حافظه هرز ٪ ۴ = ۲/ ۵۲

,حافظه مفید

٪ ۹۶ =۵۲/ ۵۰

لیست دو پیوندی

حافظه مفید ٪ ۹۲ = ۵۴/ ۵۰

حافظه هرز ٪ ۸ =

۵۴/ ۴ = ٪ ۹۲ - ۱

۲	۵۰	۲

جدول مقایسه ای پیچد گی مکانی

حافظه مفید	حافظه هرز	نوع ساختمان

ارایه	٪ ۶۶,۶۶	٪ ۳۳,۳۳
لیست تک پیو ندی	۴٪	٪ ۹۶
لیست دو پیو ندی	۸٪	٪ ۹۲

امکان پر شد گی	دسترسی	جستجو	اضافه	حذف	هرز	مفید	نوع سا ختمان
وجود دارد	تصا د فی	دودویی o(Longin)	O (n)	O (n)	۶۶,۶۶	۳۳,۳۳	ارایه مرتب
وجود ندارد	ترتیبی یکطر	O (n)	O (1)	0 (1)	٪ ۴	٪ ۹۶	لیست تک پیوند
وجود ندارد	ترتیبی دو طرفه	O (n)	O (1)	O (1)	٪ ۸	٪ ۹۲ ٪۴	لیست دو پیوند

درخت

به نوعی ساختمان داده اطلاق می شود که هر واحد سا ختمانی می توا ند به دو نوع واحد سا ختمانی از نوع خود اشاره کند

هر درخت با ریشه ان مشخص می شود .

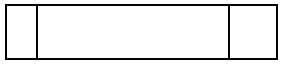
در درخت زیر a به دو گره پیو سته اشاره می کند b,c بچه های گره a هستند

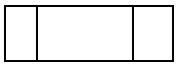
برای نگهداری درخت در حا فظه و ایجاد ا ن از واحد سا ختمان زیر استفاده می کنیم

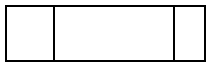
```
# in clued < stdio.h>
#include<conio.h>
#include<stdbli.h>
#include<
Struct   t
{
Struct*Left;
Char  name [ 10 ];
Struct  t * Right;
```

ریشه هر درخت با Root مشخص می شود.

بر نا مه ای بنویسید که با درخت زیر علی و حسن وحسین را در حا فظه ایجاد نما ید







```
# in clued
#include <
Struct  t
{
Struct t * left;
Char name [ 10 ];
Struct t * Right;
Void main ()
Struct t *root;
Root = ( struct t * ) malloc ( 14 );
Str cpy( root→ name,"ali");
Root →left = ( struct  t* )malloc ( 14 );
Root → left → right = null;
Root → left → left =null;
Str cpy ( root → left → name,"hasn"
Root→ right = stract * ) malloc ( 14 );
Root → left = null;
Root → right → right = null;
Str cpy ( root → right → null," hosing;
Pay ( root );

}
```

این تابع جهت پیما یش یک درخت

تابع **p** پیما یش درخت و چاپ " در ما نیتور

انواع پیما یش درخت

درخت را به سه طریق زیر می توان پیما یش نمود

۱- میان ترتیب

۲- پیش ترتیب

۳- پس ترتیب

۱- **پیمایش میان ترتیب** درخت در سه مرحله زیر انجام می گیرد

پیمایش پیمایش زیر درخت چپ

ملا قات ریشه

پیما یش زیر درخت راست

پیمایش پیش ترتیب درخت در سه مرحله زیر انجام می گیرد

ملا قات ریشه

پیمایش زیر درخت چپ

پیما یش زیر درخت راست

پیما یش پس ترتیب درخت در سه مرحله انجام می گیرد

پیما یش زیر درخت چپ

پیما یش زیر درخت را ست

ملا قات ریشه

درخت زیر را به سه تر تیب زیر پیما یش نموده و حاصل عبارت زیر را بنویسید

پیمایش سه نوع

۱- پیش ترتیب **abdfgecij** **a**

۲- میان ترتیب **fdgbeaicj**

۳- پس ترتیب **fgdebijca** **j** **b**

c

d **e** **i**

f **g**

پیش : علی وحسین وحسن

میان : حسن و علی وحسین

پس : حسنو حسین و علی

Ali

hasn **hosin**

درخت زیر را به سه ترتیب پیما یش نموده و یبارت زیر را بنو یسید

۴ -- ۲ + ۳	**in fix**	**پیما یش میان ترتیب**
۱ -- ۴ + ۲ ۳	**پسو ندی post fix**	**پیمایش پس تر تیب**
-- + ۳ ۲ ۴	**pre fix**	**پیمایش پیش ترتیب**

-- **میان : ۴ – ۲ + ۳**

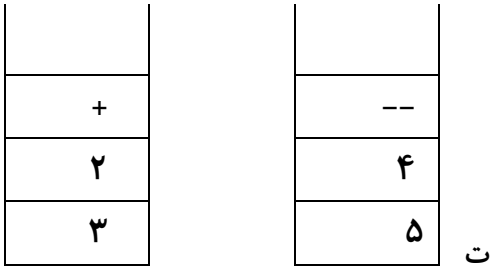
۴ + پس : -- ۴ + ۲ ۳

۲ ۳ پیش : ۴ ۲ ۳ + --

ا تگو ریتم محا سبه حا صل عبارت پسو ندی با استفاده از پشته

۱- عبارت را مرور می کنیم تا جایی که به علا مت پشته

دو تا علا مت را بر داشته



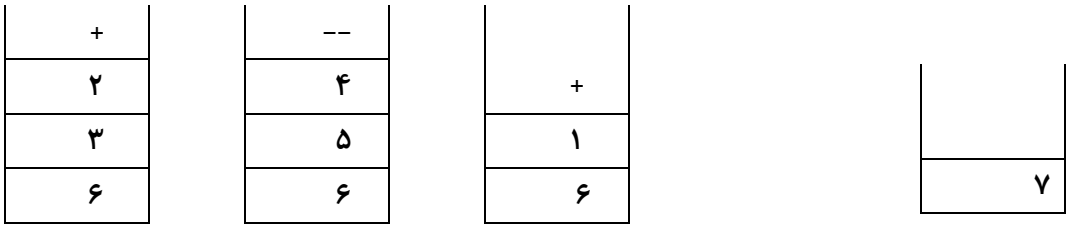
ت

تمرین

محاسبه مقدار عبارت پیشو ندی (لهستا نی)

ارائه ا لگوریتم محا سبه مقدارعبارت پیشو ند

حاصل عبارت زیر را با استفاده عبارت پسو ندی به دست ا ورید



Pp 61

بر نا مه ای بنو یسید که نا م های درخت زیر را در حا فظه ایجاد نموده و ان را به سه تر تیب در ما نیتور چاپ کند

```
#
#
#
#
Struct  t
{
Struct t  t*Left;
Char name [ 10 ];
Struct t*right;
```

```
};
Struct tree
{
T *root;
};
Insert (tree*t,char
Payper (t*root)
{
If (root != null )
{
Printf ( "\n%s",root → name );
Payper ( root → right );
Payper (root → Left );
}
}
Paypost (t*root)
{
If (root ! =nuLL )
{
Paypost ( root → right );
Paypost ( root → Left );
Printf ("\n %s",root → name );
Struct  t
{
Struct  t* left;
Char datd;
Struct t *right
};
Struct tree;
};
Void  in (t*tr,int data)
{
If(tr→ data >data
If (tr→ right==null)
{
Tr → right= (t*) malloc (siz of ( t );
Tr → right → right =null;
Tr→ right →left  = null;
Tr → right → data =data;
}
Else
In(tr → right,data );
Else
If (tr→left== null )
Tr→ left =(t* )malloc (siz of(t));
Tr →left →left → null;
Tr→ left → right →null;
Tr →left →data =data;
}
Else
```

```
In (tr→ left,data );pay in()tr→left );
Printf("\n%d",tr→ data );
Payin (tr→ right )'
}
}
Void payper (t*tr )
{
If(tr! = null
{
Printf("\n%d",tr→data);
Payper(tr → right )
}
}
Void main (void)
{
Clrscr ( );
Tree*t;
T = (tree* ) malloc ( size of (tree ));
T → root =null;
Insert (t ,4 );
Insert (t,55 );
Insert ( t,4 );
Payin (t→ root );
Getch ( );
}
```

Pp62

بر نا مه ای بنو یسید که درخت با لا را ایجاد نموده و به سه ترتیب در ما نیتور چاپ نما ید.

```
#
#
#
Void insert (tree*tr,int data )
{
If (tr → root == null )
{
Tr→ root = (t* )maLLoc ( size of ( t ));
Tr → root → Left= nuLL;
Tr → root → right =null;
Tr → root → data =data;
}
Else
```

```
In (tr → root,data );
}
Void paypos (t*tr );
{ ifr ! =null )
{
Pay pos 9 tr →left );
Paypos ( tr → right );
Print f ("\n%d",tr →data );
} }
Void payin (t*tr )
{
Voidpayin (t*tr )
{
If
```

درخت جستجوی دودویی

به نوعی درخت گفته می شود که بچه های زیر درخت چپ از ریشه کوچکتر و بچه های زیر درخت راست از ریشه بزرگتر می با شند

مثال : درخت زیر یک درخت دودویی می باشد

الگوریتم جستجو در درخت جستجوی دودویی

۱- عنصر مورد نظر را با ریشه مقا یسه می کنیم ا گر برا بر با شد پیدا شده و در غیر ا ین صورت

۲- عنصر مورد نظرا ز ریشه بزرگتر باشد زیر درخت را ست ادا مه می یا بد.

۳- ا گر کوچکتر با شد جستجو را در زیر درخت چپ ا دا مه میدهیم.

برنا مه ای بنویسید که اطلا عات را از صفحه کلید دریا فت نموده در یک درخت جستجوی دودویی قرا ردا ده و سپس عنصر مورد نظر را جستجو نمائیم.

ا لگوریتم درج در یک درخت دودویی

الگوریتمی جهت درج یک عدد در درخت دودویی ا را ئه دهید

مثا ل: فرض کنید یک رکورد اطلا عا تی شا مل ۱۰۰ با یت با شد بطور متوسط چنین پردازش ۲۰۰ رکورد مورد ا ستفا ده قرا رمی گیرد پیچیدگی مکا نی وزما نی ا نوا ع سا ختما ن دا ده ها شا مل ا را یه لیست تک پیوندی دو پیوندی ودرخت در یک جدول درج نموده ا نها را با هم مقا یسه نما ئید

هنگام پیا ده سا زی برنا مه بوسیله ا را یه ا ز یک ا را یه ۵۰۰ خا نه ای ا ستفا ده می نما ئیم

اعما ل پر شدگی	دسترسی	جستجو	اضا فه	حذ ف	حافظه هرز	حافظه مفید	**نوع ساختما ن**
وجود دا رد	تصادفی	O(Log^n)2	O(n)	o(n)	۳۰۰ ----- ٪ ۵۰۰	۲۰۰ ----- ٪ ۵۰۰	ارایه مرتب
وجود ندارد	ترتیبی یک طرفه	O (n)	O(1)	O(1)	۲ ---- ٪ ۱۰۲	۱۰۰ -------٪ ۱۰۲	لیست تک پیوندی
وجود ندارد	ترتیبی دو طرفه	O (n)	O(1)	O (1)	۴ ----٪ ۱۰۴	۱۰۰ ------٪ ۱۰۴	لیست دو پیوندی
وجود ندا رد	ترتیبی یک طرفه	O(log^n)2	O(Log^n)2	Log^n2	۴ ---- ٪ ۱۰۴	۱۰۰ ----- ٪ ۱۰۴	درخت جستجوی دودویی

هرم Heap

نوعی درخت می با شد که پدر ا ز هر دو بچه ا ش بزرگتر می با **شد max Heap**

ا گر پدر از هر دو بچه اش کوچک با شد **min Heap** می **گوئیم**

ا ز سا ختما ن دا ده Heap جهت مرتب سا زی استفا ده می کنیم روش حا صل را Heap sort می گویند

ا لگوریتم Heap sort

۱- **ا گر Heap** خا لی باشد ریشه را حذ ف و در ا را یه قرا ر بده

۲- ا ز زیر درخت ها ی با قی ما نده یک **Heap** جدید بساز برو به یک

پدر از بجه ها بزرگتر با شد ۸۰" ۱۱۰" ۱۵۰ " ۲۰۰

حذ ف ریشه

۲

پیچیدگی زما ن ا جرای Heap sort

برای حذف نیز هما ن می با شد (loq^n)o می با شد

۲

پیچیدگی زمان و حذف (Loq^n) o

۲

الگوریتم ا ضا فه کردن

عنصرجدید را به عنوان یک برگ Heap ا ضا فه می کنیم

ا ین عنصررا درخت تا جا یی با لا می بریم که درخت حا صل Heap باشد

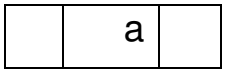
پیچیدگی زمان اجرای ا ضا فه کردن در (loq^n)o می با شد Heap-→

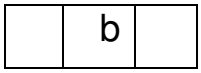
۲

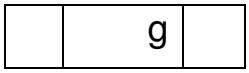
نخ کشی در درخت

در درخت ها می توا ن به جای اشا ره گر هایnull در برگها ا درس با زگشت به گرهها ی با لائی را طوری قرار داد که پیما یش درخت به سا دگی (بدون ا ستفا ده از توابع بازگشتی نظیر لیست های پیوندی ان ها را پیما یش کرد تنها می توا ن چند نوع نخ کشی داشته با شیم که ۱- نخ کشی پیش ترتیب

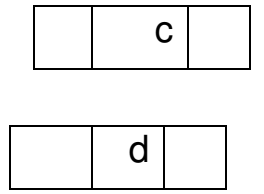
۲- نخ کشی میان ترتیب

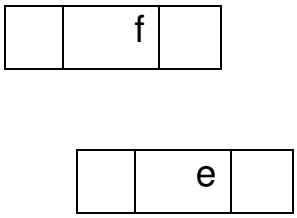






۳- نخ کشی پس ترتیب





dcebfag

نخ کشی inorder

ا شا ره می کنیم null دیگر نیست .
مثال درخت زیر را با نخ کشی inorder نخ کشی کنید
با ایجاد کردن پیوندهای

مثال : درخت فوق را به صورت پیش ترتیب وپس ترتیب نخ کشی نما ئید.

مرتب کردن ادغا می sort merg

قبلا با روشها ی مرتب شازی حبا بی وانتخا بی ا شنا شده ا یم ا کنون روش جدید مرتب کردن تحت عنوا ن مرتب کردن ادغا می را ا را ئه می دهیم
برای یاد اوری در ا لگوریتم مرتب سا زی حبا بی عنا صرموجود در سا ختما ن داده را دوبه دو با هم مقایسه می کردیم ودر صورت نیاز به جا بجایی عناصررا جا به جا می کردیم در هر مرحله مطمئنا بزرگترین عنصر ۰(یا کوچکترین عنصر) در مکا ن نهایی خود قرار می گرفت پیچیدگی زما ن ا ین ا لگوریتم o(n^2) می با شد.
در sort انتخا بی در هر مرحله بز رگترین یا کوچکترین عنصر را پیدا می کنیم وا ن را در مکا ن ا صلی خود قرا ر می دهیم سپس از عنا صر با قی ما نده بزرگترین و کوچکترین عنصر را پیدا نموده در مکان اصلی قرار می دهیم

پیچیدگی زما ن ا جرا ا ین نوع مرتب سازی نیز دارای (n^2)o یعنی ا گر بخوا هیم یک ساختما ن داده با هزار عنصرا طلا عا تی توسط یکی از ا ین روشها مرتب نما ئیم یک میلیون دوم (1000000) o مقا یسه انجام دهیم یعنی مرتبه ا ین ا لگوریتم
(1000000)O میباشد.

اکنون مرتب سازی ادغامی را توضیح می دهیم وبه کمک هم مرتبه ا ن را به دست می ا وریم . در این روش داده ها را به دو قسمت مسا وی تقسیم می کنیم و سپس دو تا قسمت را مرتب نموده دو قسمت را باهمدیگر ادغا م می کنیم

{ 100,20,} {30, 200} {22,101} {110,10}

تا رسیدن به دو عضوی تقسیم می کنیم بعد دوبه دو ا نها رابا هم مرتب نموده در هر مرحله ا ن را دو به دو ا دغام می کنیم

{ 200,100,30,20 } {110,101,22,10 }
{ 200 ,110, 101 ,100 ,30,22, 20 ,10 }
برنا مه ا صلی { 100,20,30.200,22.101,110,10 }

محاسبه پیچیدگی زما ن ا جرای مرتب سا زی ا دغا می

فرض کنید ۲ تامجموعه که هر کدا م دارای L تا عنصر با شد می خوا هیم ا دغا م کنیم برای ا ین منضور در بهترین حا لت مقدا ر مقا یسه ها L تا ا ست وبد ترین حا لت دارای L تا می باشد.

بهترین حا لت L موقعی است که برای ادغا م تک تک عنا صر نیاز به مقا یسه داشته با شند مثال : {200,150,100,50,} { 210,160,110,60 }

بهترین حا لت ۴ مقایسه

در بد ترین حا لت با ۲ تا L

در بهترین حا لت L

تعداد عنا صر هر مجموعه	تعداد مجموعه	مرحله
2	N/2	۱-
4	n/4	۲-

۳-	n/8	8
۰	۰	۰
۰	۰	۰
m	n/ 2^m = 1	یک مجموعه n

N =2^m➔Log2^n= Log2^n = m

تعداد مقایسه ها در کمترین حا لت	تعداد مقایسه در بهترین حا لت
n/4 x 2	n/4 x4 n
n/8 x4	n/8 x8 n
n/16 x8	n/16 x16 n
1/2 n	1/2 x2n n
+ n/2 m	N تاm
n/2 Log^n 2	nLog^n 2

پیچیدگی زما نی الگوریتم ادغا می در بهترین n/2 Log^n ۲

// // // // // بدترین حا لت n Log^n ۲

که هر دو ا نها ا ز مرتبه o (nLog^n) ۲

روش مرتب سازی سریع quick sort

که در ا ن در هر مرحله یک عدد محل نها یی خودرا به دست می ا ورد.که محل نها یی داده ای که محل نها یی خود را به دست ا ورده داده ها را به دو قسمت تقسیم می کند تا مرتب شدن تما م داده ا لگوریتم مرتب سا زی سریع را به دو صورت ا عما ل می کنیم

داده های زیر را می خواهیم به روش quick sortرا مرتب سا زی می کنیم در مرحله اول عدد ۵۰ محل نهایی خود را در داده ها بدست می اورد بعد از مشخص شدن محل نها یی ۵۰ داده ها را به دو قسمت تقسیم می کند که بایستی ا لگوریتم quick sort را در هر دو قسمت با ید مجددا بکار ببر یم

مرتب سا زی صعودی

50, 100,44,200,20,25,12
12,100,44,200,20,25,50,
12,50,44,200,20,25,100
12,25,44,200,20,50,100,
44> 50
12,25,44,50,20,200,100,
12,25,44,,20,50,200,100,
12,25,44,20,50,200,100

بعد از مشخص شدن عدد ۵۰ داده ها به دو قسمت تقسیم می شوند که ا لگوریتم quick sortرا در هر دو قسمت بکا ر ببریم

محاسبه پیچیدگی زمان ا جرا مرتب سا زی سریع

بهترین پیچیدگی زمان اجرا مرتب سا زی سریع موقعی بدست می ا ید که داده ای مکا ن نها یی ا ن مشخص می شود داده ها را به دو قسمت مسا وی تقسیم کند در ا ین صورت پیچیدگی زما ن ا جرا برا بر (nLog^n) o خوا هد بود از مرتبه (nlog^n) o خواهد بود

۲ ۲

بد ترین حا لت ا لگوریتم در مرتب سازی سریع موقعی ا تفا ق می ا فتد کداده ای مکا ن نها یی ان مشخص می شود نتوا ند داده ها را تقسیم کند یعنی در ابتدا یا در انتها ی داده ها قرار گیرد. در این صورت روش مرتب سازی سریع هما نند روش مرتب سازی انتخا بی می با شد که دارا یمرتبه (n^2) o می با شد

چرا داده ها را مرتب سا زی می کنیم ؟

ما روش ها ی مرتب سا زی را تمرین نموده تا بوسیله ا ن داده ها را مرتب سازی بکنیم که جدا عملیا ت جستجو به سا دگی انجا م کیرد و ایا می توا ن روشی ارائه داد که بدون مرتب سازی سریعتر بتوا ن داده مورد نظر را پیدا کرد ؟ جواب بلی ا ست

مطابق ا صل قا نون زمان و حا فظه اکر حا فظه صرف بکنیم می توان زما ن را بدست ا وریم.روشی تحت عنوا ن در هم سا زی وجود دارد که می توا ن داده ها را بدون مرتب سازی و توسط تا بع در هم ساز در حا فظه قرا ر داد تابع در هم ساز در وا قع اطلا عا ت فرد را دریا فت نموده و ا درس ا ن حا فظه را به ما تحویل می دهد

فرض کنید ۷۰ میلیون ا یرا نی اطلا عات ا نها در هارد دیسک ذخیره شده اگر تا بع تعریف کنیم که با دریافت کد ملی فرد شماره sctor در حافظه تحویل دهد که مشخصا ت ا ن درsctor قرار دارد در ا ن صورت نیا زی به مرتب سازی وجود ندارد.

در استفاده از توابع در هم ساز ا کثر موارد ما مجبور می شویم چندین برا بر حا فظه هرز داشته با شیم و L در عوض نیازی به مرتب سا زی و عملیا ت جستجو نمی با شد
ا ین روش دارا ی ا ین مشکل نیز می با شد که در برخی موارد پیاده سازی ممکن است دو عنصر ا طلا عا تی بخوا هند در یک جا قرار گیرند تصادم نمی گوئیم روش ها ی متفا وتی جهت رفع تصا دم وجود دارد که مطا لعه ان به دانشجوی عزیز وا گذار می شود

Radix sort مبنا یی

روش مرتب سازی مبنا یی radix روش مرتب سازی ا عداد و لغات در فرهنگ لغا ت گفته می شود

اگر ما بخوا هیم یک کتا ب را تا یپ کنیم اگر برا ی حروف از کد اصلی ا ن ا ستفا دهبکنیم این کتا ب ۸ مگا با یت حا فظه اشغا ل خوا هد کرد اگر ما بتوا نیم روش را ا را ده دهیم که هما ن کتا ب را در سا یز کو جکتر ذخیره کند روش مفید خوا هد بود و این ا مکانپذیر می با شد چون حروف انگلیسی ۲۶ تا می با شد برا ی کد کردن ا ن ها کافی است از ۵ بیت ا ستفا ده کنیم (و یا حتی کمتر) در صورتیکه در کد ا صلی برا ی ذخیره کردن ۸ بیت می با
شد بطور ساده اگر ما به جا ی کد اصلی از یک کد ۵ بیتی جهت ذخیره نمودن حروف استفا ده نمئیم در ا ین صورت کتا ب ۸ مگا با یتی (با کد ا سکی) ذخیره خوا هد شد

روش کد گذاری منبع sors in codin

دارا ی مبا حث خا ص خود را در تئوری ا طلا عا ت دارا می با شد یکی از ا ین روش ها ا لگوریتم haf man می با شد در روش ا لگوریتم haf man کد به حروف متنا سب با تکرار حروف می با شد یعنی اگر حرف a تعداد دفعا ت تکرار زیا د است با یستی کد با طول کمتری اختصا ص دهیم و بر عکس sors in codin مطالع به دانشجوی محترم وا گذار می شود

درج درخت در ارا یه

می توان درخت را به صورت زیر در ا را یه قرا ر داد

۱- ریشه را در خا نه یک قرا ر می دهیم [1] a
۲- بچه جپ را در خا نه [2i] a قرار می دهیم
۳- بچه را ست را در خا نه [I] a قرار می دهیم

A [I], a [1] , a [2 I] , a [2i + 1]

A [1]= 100
A [2] = 50
A [3] =200
A [4] =30
A[5] = 60
A [6] = 150
A [7] = 210
A [8] = 0

مسئله برج ها ی ها نوی

مسئله برج ها ی ها نوی به این صورت ا ست که حلقه ها ی مرتب موجود در a را به b انتقا ل دهیم با این شرط که همیشه با یستی حلقه کوجکتر روی حلقه بزرگتر نیستند بر عکس مجاز نمی با شد

ا ین مسئله را با ا ستفاده از پشته و یا توا بع با ز گشتی می توا ن حل نمود

H (n -1 a ,aux,b)
H (n-1) aux,a,b)

گراف

گراف یک سا ختمان گسسته ریا ضی ا ست که را بط بین ا عضای یک مجموعه را نشا ن میدهد

فرض کنید گرا ف می خوا هد ارتباط بین ا عضای مجموعه abcd را نما یش دهد به اعضای v گره می گوئیم

مجموعه ای به نا م E جهت نما یش ا رتبا ط ا عضا ی v به کا ر می رود به عنوا ن مثا ل از a به b را بطه ای وجود دا شته با شد دو تا یی مرتب abعضو مجموعه E خوا هد بود به مجموعه با لا می گوئیم یک گراف با دو مجموعه v,f مشخص می شود و ا ن را به صورت G (v,f) نما یش می دهیم



برای تعریف این گراف به کا مپیوتر می توا ن از یک جدول (ارایه دو بعدی) که به ا ن ما تریس مجا ورتی می کویند استفا ده می کنیم

ا ین جدول یک جدول ۴X۴ خوا هد بود

	a	b	c	d
A	0	1	0	0
B	0	0	1	0
C	0	0	0	0
d	0	0	0	1

در ما تریس فوق تعداد عنا صر غیر صفر ا ندک می با شد به ا ین ما تریس ۰ماتر یس خلوت می گو یند

تمرین : ماتریس زیر را با ا ستفا ده از لیست های پیوندی نما یش دهید

ساختما ن های بکا رگرفته شده را به زبا ن c بنویسید

```
Struct  Q
{
Char.ch;
Struct E*first;
Struct Q*next;
};
Struct E
{
Char  ch;
Struct E*next;
};
```