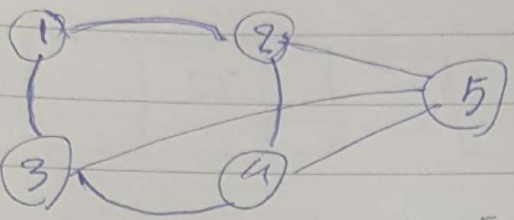
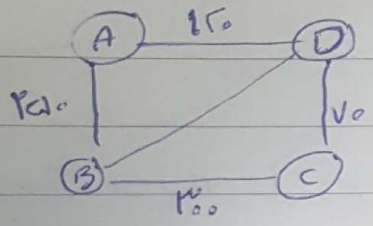
ماتریس مجاورت گراف زیر را رسم کنید.



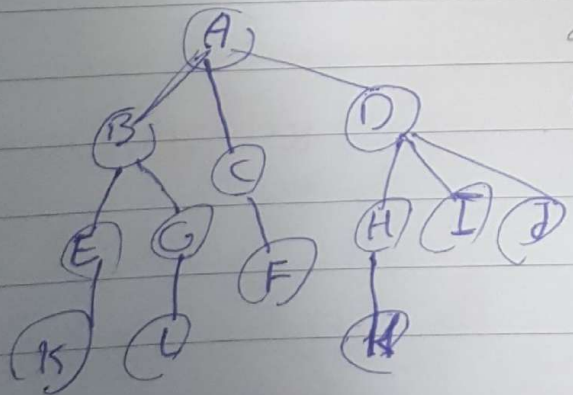
	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	1	1
3	1	0	0	1	1
4	0	1	1	0	1
5	0	1	1	1	0



	A	B	C	D
A	۰	۲۵۰	۰	۱۲۰
B	۲۵۰	۰	۳۰۰	۱۹۵
C	۰	۳۰۰	۰	۷۰
D	۱۲۰	۱۹۵	۷۰	۰

«درخت»

درخت گرافی است که دور ندارد. هر درخت گره‌هایی دارد به نام ریشه و بطور کلی به دو دسته‌ی عمومی و دودویی تقسیم می‌شوند.



درجه: به عناصر موجود در درخت گره گوییم، درجه گره برابر است با تعداد فرزندان آن گره.

زیر درخت‌های یک گره درجه آن گره گوییم.

iman @jozveotamrin

Scanned by CamScanner

Subject:
Date

تعریف برگ: به گره هایی که درجه صفر دارند یعنی گره های پایانی گفته می شود.

درجه درخت: حداکثر درجه گره های هر درخت، درجه آن درخت را مشخص می کند.

طول مسیر درخت، مجموع طول های تمام مسیرها از ریشه به برگ است.

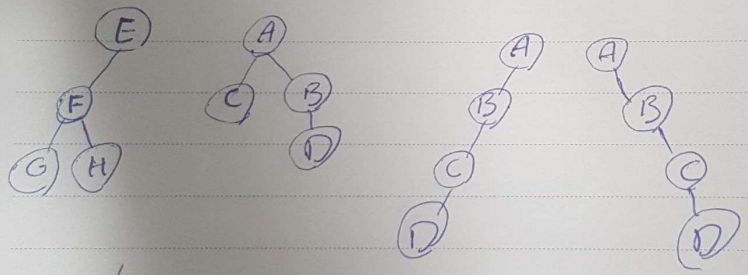
درخت دودویی: درختی است که هر گره حداکثر ۲ فرزند داشته باشد.

درخت K تایی: درختی است که تعداد فرزندان هر گره حداکثر K باشد.

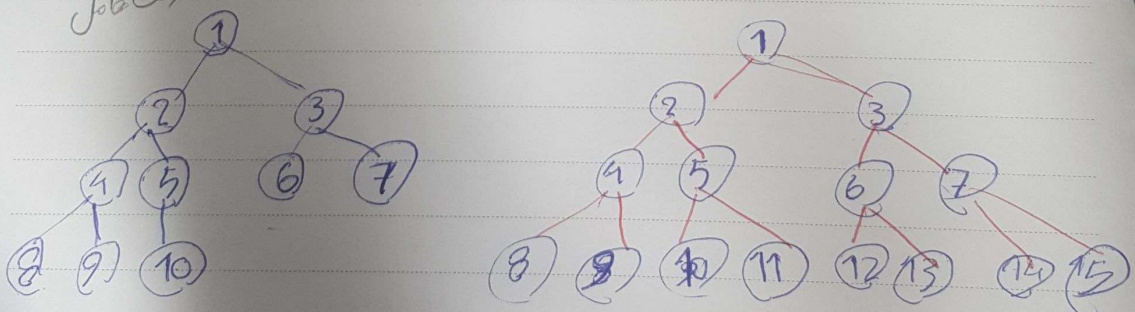
درخت متوازن: درختی است که اختلاف سطح برگ های آن حداکثر یک باشد.

درختی که اختلاف سطح برگ های آن صفر باشد به آن کاملاً متوازن می گویند.

درخت دودویی:



درخت کامل



P4PCO

Scanned by CamScanner

Subject:
Date

درخت کامل n: درختی است که تا درجه n-1 پر و در درجه n از سمت چپ پر باشد

درخت پر: درختی است که تمام گره‌ها به جز برگ‌ها از درجه ۲ باشد.

Scanned by CamScanner

Subject

Date:

ماتریس های بالا مثلثی و پایین مثلثی:

$$A_{۴\times۴} = \begin{bmatrix} a_{۰۰} & ۰ & ۰ & ۰ \\ a_{۱۰} & a_{۱۱} & ۰ & ۰ \\ a_{۲۰} & a_{۲۱} & a_{۲۲} & ۰ \\ a_{۳۰} & a_{۳۱} & a_{۳۲} & a_{۳۳} \end{bmatrix}$$

$۴\times۴ = ۱۶ \times ۲ = ۳۲$ بایت

	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
B	$a_{۰۰}$	$a_{۱۰}$	$a_{۱۱}$	$a_{۲۰}$	$a_{۲۱}$	$a_{۲۲}$	$a_{۳۰}$	$a_{۳۱}$	$a_{۳۲}$	$a_{۳۳}$

$۱۰ \times ۲ = ۲۰$ بایت

$\frac{n^۲+n}{۲} = \frac{n(n-۱)}{۲}$ ⟵ تعداد عناصر صفر

$n^۲ - \frac{n(n-۱)}{۲} = \frac{n(n+۱)}{۲}$ ⟵ تعداد عناصر غیر صفر

$K = \frac{i(i+۱)}{۲} + j$ نحوه به دست آوردن یک خانه در ماتریس با استفاده از نوار

(نیمه)

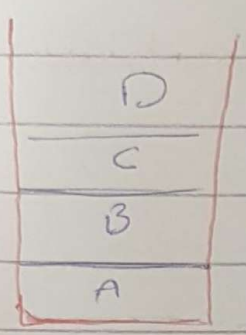
تعریف: لیست یک لیست از عناصر است که عملیات حذف و اضافه کردن عناصر تنها از یک طرف آن صورت می‌گیرد

ساختار پشته به گونه‌ای است که آخرین عنصری که وارد پشته شده اولین عنصری است که از آن خارج می‌شود

MICRO

Scanned by CamScanner

Subject: Date:



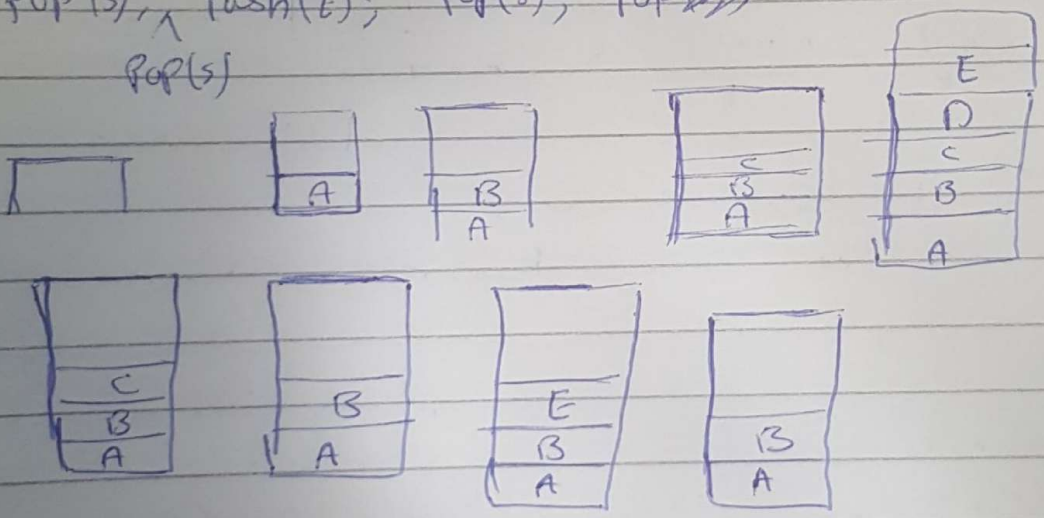
Top = 4
← Top = 3
← Top = 2
← Top = 1
← Top = 0

Push / Top ++ / Insert

Pop / Remove / Top - -

Push(S, A); Push(S, B); Push(S, C); Push(S, F); Push(S, E);

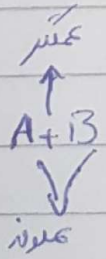
Pop(S); Push(E); Pop(S); Pop(S)



in fix A+B

Pre fix +AB

Post fix AB+



A+B*C

MICRO A = 2
B = 3
C = 9

(A+B)*C = 25

A+(B*C) = 29

Scanned by CamScanner

Subject: Date:

((a*b)+c)-(a/d)

ab*
x
ab*c+
y
ad/
z

Post fix : ab*c+ad/-

pre fix : -+*abc/ad

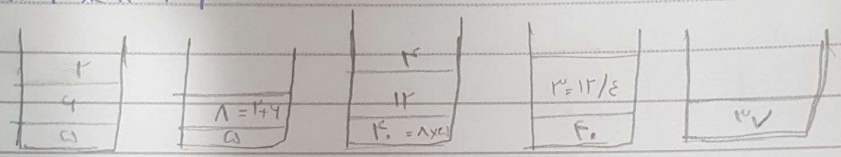
+*abc
اول

/ad
دوم

تابع ضمنی هم پیشگی ندارد

با استفاده از پشته حاصل عبارت زیر را به دست آورید.

۴ ۱۲ * ۶۲ + -



در مثال عبارت Post fix به صورتی که گفته شد عمل کردیم اعداد را به پشته قرار دادیم هر گاه به عملوند رسیدیم ۲ عدد از بالای پشته برداشتیم عمل را بر آنها انجام دادیم و حاصل را به بالای پشته قرار دادیم در نهایت در پشته یک عدد باقی ماند

عبارت را پیاده سازی نمودیم عدد بالای پشته حاصل عبارت خواهد بود.

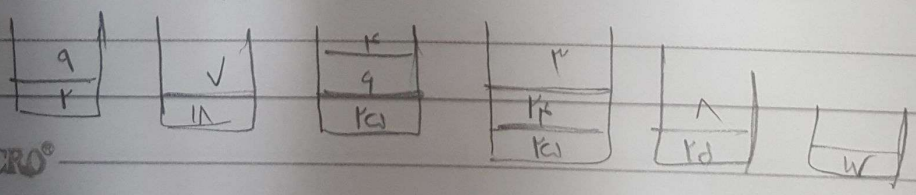
ابتدا به Post fix تبدیل کنید سپس با استفاده از پشته حاصل عبارت زیر را به دست آورید.

[[(۴*۹)+۷]-[(۶*۴)/۲]]

۴۹*۷+

۶۴*۲/

۴۹*۷+۶۴*۲/-

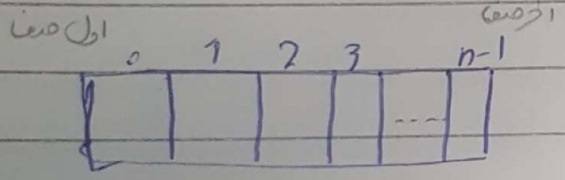


MICRO

Scanned by CamScanner

Subject: Date:

صف : صف لیست مرتبی است که عمل اضافه کردن از یک طرف آن به انتهای صف (rear) و عمل خواندن یا حذف کردن از طرف دیگر (front) انجام می‌شود. ساخت صف به صورت FiFo می‌باشد یعنی اولین عنصری که وارد شده، اولین عنصری است که خارج می‌شود. برای پیاده‌سازی صف به ۲ متغیر front, rear نیاز هستیم که اولی مشخص کننده‌ی شماره‌ی خانه‌ی خالی قبل از اولین عنصر است و rear به آخرین عنصر اشاره می‌کند.




```
<type> q [n];
front , rear : -1 ... n-1        داشته تغییرات
front = rear = -1;               مقدار اولیه
if ( front = = rear)             شرط خالی بودن صف
if ( rear = = n-1)               " " پر " "
```

مشکل اصلی صف معمولی این است که یک وقت‌هایی اتفاق می‌افتد که rear به انتها رسیده دیگر نمی‌تواند چیزی را اضافه کند در حالی که صف واقعاً خالی است.

MICRO

Scanned by CamScanner

Subject:

Date:

«لیست پیوندی»

لیست پیوندی دنباله‌ای از تعدادی عناصر تشکیل شده که هر عنصر در این لیست node نامیده می‌شود و هر node

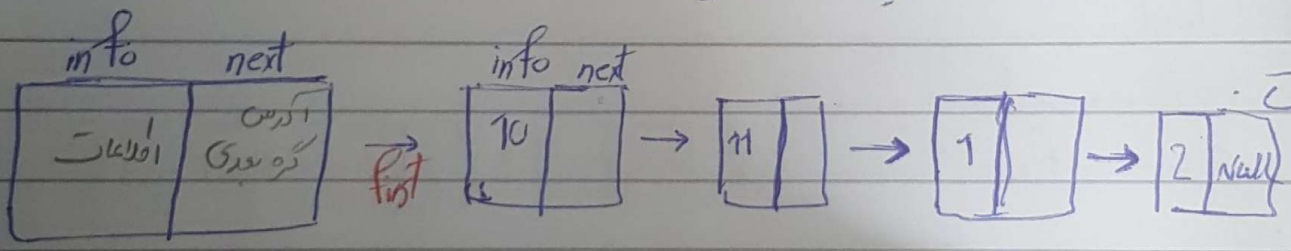
شامل ۲ فیلد اطلاعات که شامل اطلاعات واقعی و آدرس گره بعدی. فیلد اطلاعات داده مربوط به آن node

را ذخیره می‌کند و فیلد آدرس گره بعدی است با توجه به نوع پیاده‌سازی لیست پیوندی نیاز به لیست هدی

عناصر در حافظه نیست و هر node به آدرس گره بعدی در حافظه اشاره

می‌کند. برای دسترسی به یک لیست پیوندی از یک اشاره‌گر به نام first استفاده می‌شود که آدرس اولین node

لیست را نگه می‌دارد. اشاره‌گر node آخر لیست به Null اشاره می‌کند تا انتهای لیست



«پیمایش یک لیست پیوندی»

برای پیمایش یک لیست پیوندی با استفاده از گره‌ها به این صورت عمل می‌کنیم که با اشاره به اولین node لیست که

first مشخص می‌شود پیمایش را آغاز کرده و تا جایی این روند را ادامه می‌دهیم که به اشاره‌گر null برسیم.

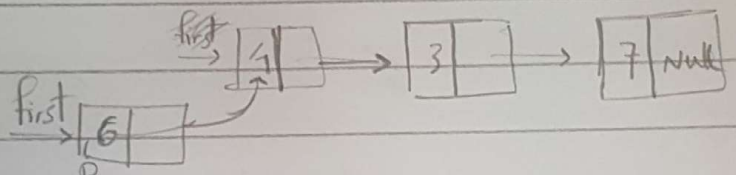
MICRO
@jozveotamrin
Scanned by CamScanner

Subject: Date:

```
P = first
while (P != null)
{
Process (P -> info ) //  پردازش گره
P = P -> Next  //  گره بعدی
}
```

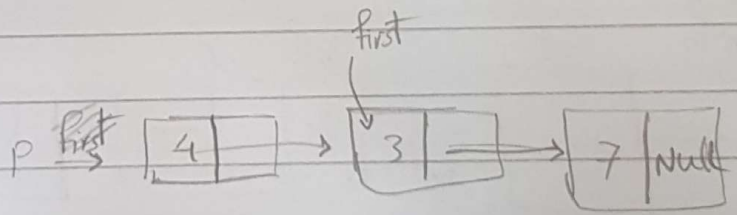
اضافه کردن node به اول لیست

```
P = getnode ();
P -> info = 6;
P -> next = first;
first = P;
```



حذف یک node از اول لیست

```
P = first
first = P -> next
x = P -> info
freenode (P)
```



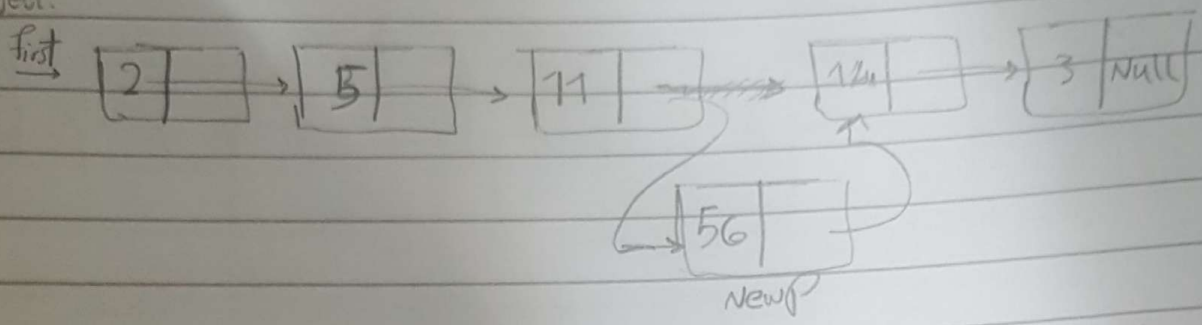
x = 4

اضافه کردن یک node به وسط لیست

```
NewP = getnode ();
NewP -> info = 56;
NewP -> Next = P -> Next;
P -> Next = NewP;
```

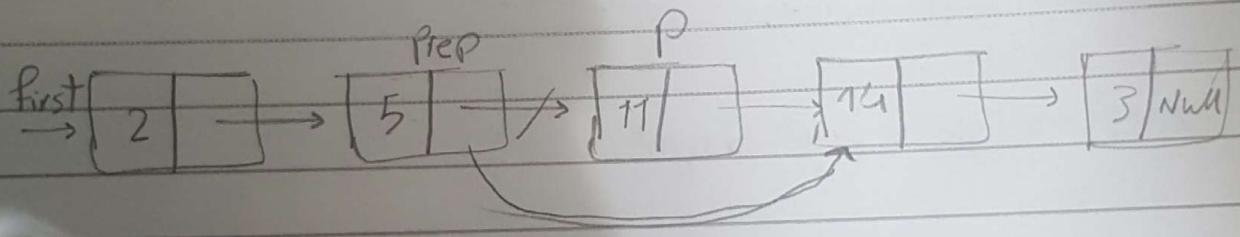

Scanned by CamScanner

Subject: Date:

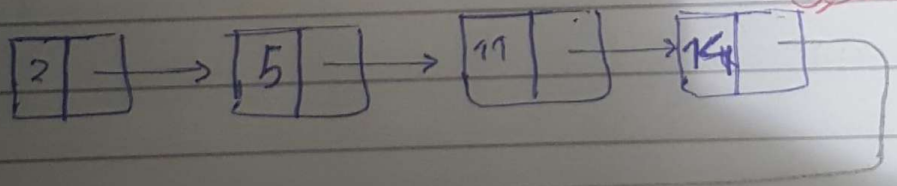


حذف کردن یک node از وسط لیست:

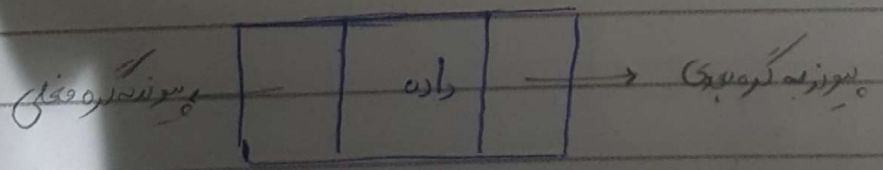
```
PreP -> Next = P -> Next
x = P -> info
freenode(P)
```



لیست پیوندی حلقوی:



لیست پیوندی ۲ پیوندی:

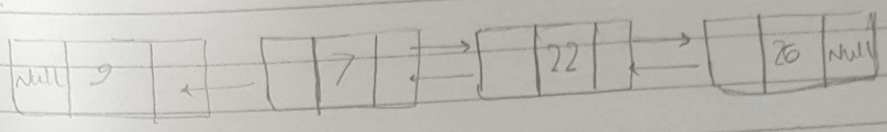


ساختار گره لیست دو پیوندی

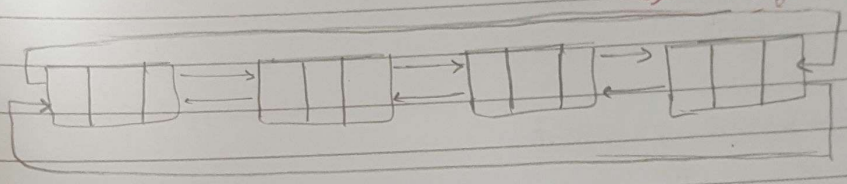
```
struct node
{
    node * left;
    int info
    node * right;
}
```

Scanned by CamScanner

Subject: Date:

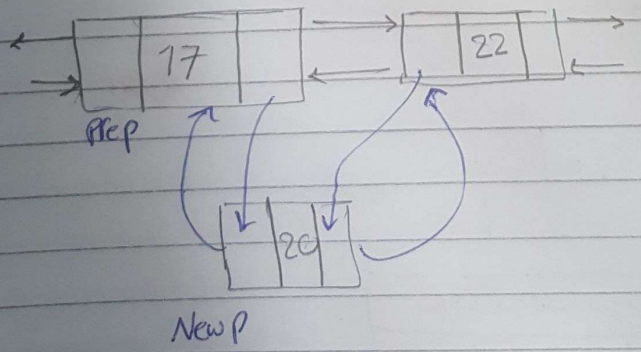


لیست دوپیوندی حلقوی:



درج یک عضو در لیست دوپیوندی:

```
NewP -> Left -> Prep;
NewP -> right = Prep -> right
Prep -> right -> Left = newP;
Perp -> right = newP
```

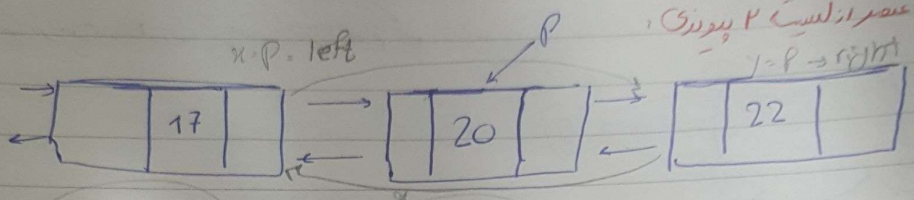


MICRO

Scanned by CamScanner

Year. Month. Day. Subject.

حذف یک عنصر از لیست ۲ پیوندی:



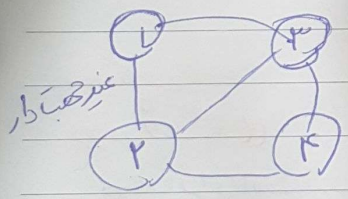
p → right → left = p → left ;

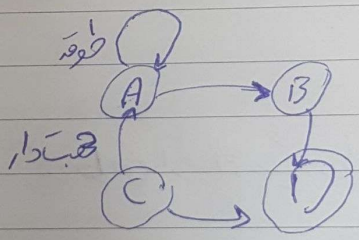
p → left → right = p → right ;

free p();

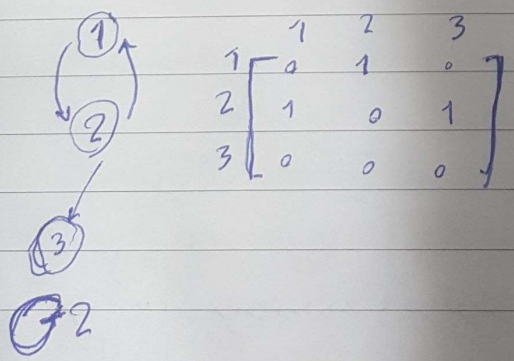
«گراف»

$G = (V, E)$ مجموعه‌ای از یال‌ها و رأس‌ها





«ماتریس مجاورت»

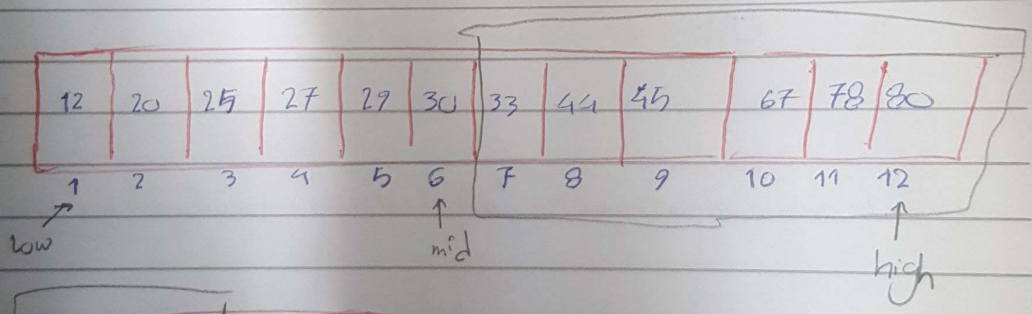


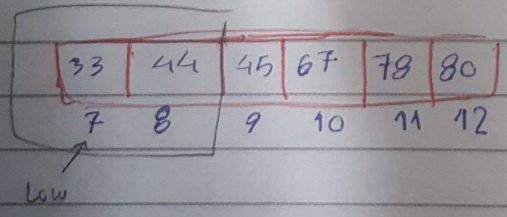
iman

Scanned by CamScanner

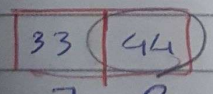
Subject: Date:

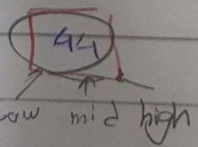
```
{ low = 0; high = n-1;
while (low <= high) {
mid = (low + high)/2;
if (K < x[mid])
    high = mid - 1;
else if (K > x)[mid] )
if(K = m) low = mid + 1
else { cout << "The number" << K << "is in carray";
exit (0); }
} cout << "The number is not in array";
```





K = 44





Scanned by CamScanner

Subject: Date:

نمایش چندجمله‌ای:

$$f(x) = 5x^4 + 3x^2 + 7x - 6$$

$$f(x) = 9x^{1000} + 2$$

ضریب	9	2
توان	1000	0

ضریب	5	3	7	-6
توان	4	2	1	0

روش هورنر:

$$ax^3 + bx^2 + cx + d$$

$$\Rightarrow a \cdot x \cdot x \cdot x + b \cdot x \cdot x + c \cdot x + d$$

$$((a \cdot x + b) \cdot x + c) \cdot x + d$$

$n = 2$ $\quad O(n)$

ماتریس اسپارس:

به ماتریسی اسپارس گوییم که تعداد عناصر صفر آن بسیار زیاد باشد و برای ماتریس ها به جای ذخیره سازی یک ماتریس

فقط عناصر غیر صفر آن را ذخیره می کنیم.

شکل ذخیره سازی برای ماتریس اسپارس:

در این روش در سطر اول تعداد سطر، تعداد ستون و تعداد عناصر غیر صفر ماتریس قرار می گیرد و در سطرهای بعدی

مختصات و مقدیر غیر صفر ماتریس قرار می گیرد.

MICRO

Scanned by CamScanner

Subject: Date:

۱۵	۰	۰	۲۲	۰	-۱۵
۰	۱۱	۳	۰	۰	۰
۰	۰	۰	-۶	۰	۰
۰	۰	۰	۰	۰	۰
۹۱	۰	۰	۰	۰	۰
۰	۰	۲۸	۰	۰	۰

بایت ۷۲ = ۲×۳۶ = ۶×۶×۲

⇓

۶	۶	۸
۰	۰	۱۵
۰	۳	۲۲
۰	۵	-۱۵
۱	۱	۱۱
۱	۲	۳
۲	۳	-۶
۴	۰	۹۱
۵	۲	۲۸

۹×۳ = ۲۷×۲ = ۵۴ بایت

جمع ۲ ماتریس اسپارس زیر را بنویسید.

A

۳	۶	۵
۰	۵	۷
۱	۲	۱۸
۱	۴	-۳
۲	۳	۱۹
۲	۵	-۱۱

B

۳	۶	۳
۰	۲	-۶
۱	۲	-۱۸
۳	۰	۴

۳	۶	۶
۰	۲	-۶
۰	۵	۷
۱	۴	-۳
۲	۳	۱۹
۲	۵	-۱۱
۳	۰	۴

چون ۶ تا سطر داریم این را به عنوان ۶ می‌نویسیم

چون ستون ۰ از ۵ ... است زودتر می‌نویسیم

۱۸ + (-۱۸) = ۰

چون صفر شده و ۱ ۲ برابر است نمی‌نویسیم

MICRO

Subject: Date:

تمرین ۲ ماتریس حل را حساب کنید.

A

$$\begin{bmatrix} ۴ & ۶ & ۵ \\ ۰ & ۵ & ۷ \\ ۱ & ۲ & ۱۸ \\ ۱ & ۴ & -۳ \\ ۲ & ۳ & ۱۹ \\ ۳ & ۰ & -۱۱ \end{bmatrix}$$

B

$$\begin{bmatrix} ۳ & ۶ & ۳ \\ ۰ & ۲ & -۶ \\ ۱ & ۲ & -۱۸ \\ ۳ & ۰ & ۴ \end{bmatrix}$$

$$\begin{bmatrix} ۴ & ۶ & ۹ \\ ۰ & ۲ & ۹ \\ ۰ & ۵ & ۷ \\ ۱ & ۲ & ۳۶ \\ ۱ & ۴ & -۳ \\ ۲ & ۳ & ۱۹ \\ ۳ & ۰ & -۱۵ \end{bmatrix} \begin{matrix} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{matrix}$$

چون آن سطر ۰ است ← $۰ - (-۶) = ۶$

$۱۸ - (-۱۸)$ ←

$-۱۱ - (-۴) = -۱۵$ ←

نکته: جمع، تفریق و ضرب ۲ ماتریس فقط (اسپارس) الزاماً صفر نیست.

مثال)

$$\begin{bmatrix} ۱ & ۱ & ۱ \\ ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ \end{bmatrix} + \begin{bmatrix} ۰ & ۰ & ۰ \\ ۱ & ۱ & ۱ \\ ۰ & ۰ & ۰ \end{bmatrix} = \begin{bmatrix} ۱ & ۱ & ۱ \\ ۱ & ۱ & ۱ \\ ۰ & ۰ & ۰ \end{bmatrix}$$

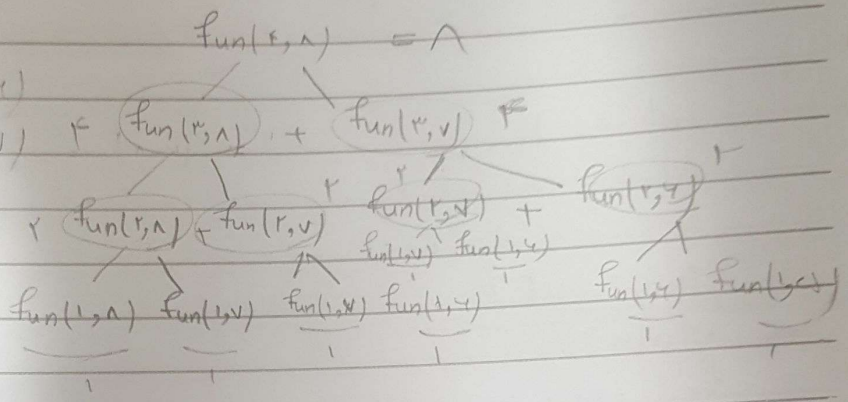
$$\begin{bmatrix} ۱ & ۰ & ۰ \\ ۱ & ۰ & ۰ \\ ۱ & ۰ & ۰ \end{bmatrix} \times \begin{bmatrix} ۱ & ۱ & ۱ \\ ۰ & ۰ & ۰ \\ ۰ & ۰ & ۰ \end{bmatrix} = \begin{bmatrix} ۱ & ۱ & ۱ \\ ۱ & ۱ & ۱ \\ ۱ & ۱ & ۱ \end{bmatrix}$$

MICRO

Scanned by CamScanner

Subject:

Date:




```
float sum(float list[], int n)
{
float s = 0;
int i;
for(i=0; i<n; i++)
s = s + list[i];
return s;
}
```

	0
	0
	1
	0
	n+1
	n
	1
	0
	2n + 2

```
for(i = 1; i<=m; i++)
for(j = 1; j<=n; j++)
{
X = X + 1;
}
```

	m+1
	m(n+1)
	m x n

(m+1) + (m(n+1)) + m x n

Scanned by CamScanner

Subject: Date:

تحلیل پیچیدگی زمانی در حالات بهترین، بدترین و متوسط

```
int a[n];
int s=0;
for(i=0;i<n;i++)
s = s+a[i];
```

```
int a[n], i;
for (i=0; i<n; i++)
if (x==a[i])
{
cout<<"Yes";
exit(0);
}
cout<<"No";
```

$B(n)=1$ Best

$W(n)=n$ Worst

$A(n)=\frac{n+1}{2}\simeq\frac{n}{2}$

$B(n)=W(n)=T(n)=n$

الف) `x = x+1;` $\Rightarrow O(1)$

$O(n)$

مرتبه اجرای الگوریتم

ب)
```
{
y = 2*15;
x = n+1; ⇒ O(1)
z = a+y; }
```
$\Rightarrow O(n^2)$

ج) `for(i=0; i<n; i++)` $\Rightarrow O(n)$

`x = x+1;`

MICRO

Scanned by CamScanner

Subject:

Date:

۱) for(i = ۲; i <= n-۴; i = i+۲)
x = x+1;

$\frac{n^2}{2} - 2$ $O(n)$

۵) for(i = ۱; i <= n+۵; i++)
for(j = ۲; j <= n; j = j+۲)
x = x+1;

$\frac{(n+5)\times n}{2} = \frac{n^2+5n}{2}$ $O(n^2)$

for(i = ۵; i^۲ <= n+۹; i++) $n+3$
for(j = ۷; j <= ۱۰۰; j = j+۳) ۹۹۲
for(k = n-۲; k > ۲; k = k-۲) $\frac{n}{2}$

۹۹۲ * n * $\frac{n}{2}$ = ۹۹۲ $\frac{n^2}{2}$ $O(n^2)$
ضریب ثابتی ندارد

x = ۰; $n = 2^k$
i = n;
while (i > ۱) {
x = x+1;
i = i/۲;
}

i	x
2^k	۰
2^{k-1}	۱
2^{k-2}	۲
2^{k-3}	۳
۲	$k-1$
۱	k

$n = 2^k = 2^{k-1}$ $O(\log_2^n)$

MICRO

Scanned by CamScanner

Subject: Date:

```
i = 1
while (i<n)
{
  عملیات ;
  i = *k;
}
```

$\Rightarrow O(\log_k^n)$

```
i = n;
while (i>1)
{
  عملیات ;
  i = /k;
}
```

$O(\log_k^n)$

به ازای هر ضرب و تقسیم باید log استفاده کنیم

```
i = n+3;
while(i>4)
{
  عملیات ;
  i = -c;
}
```

$\Rightarrow O(n)$

«رشد توابع»

$f(n) = \frac{1}{n}$

$g(n) = \frac{1}{n^2}$

$\lim_{n\to\infty} \frac{\frac{1}{n}}{\frac{1}{n^2}} = \lim \frac{n^2}{n} = \lim n = \infty$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

$f(n) > g(n)$ رشد

$\frac{1}{n} > \frac{1}{n^2}$ رشد

MICRO

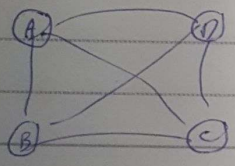
Scanned by CamScanner

Subject: Date:

$$\frac{1}{n^2} < \frac{1}{n} < 1 < \log n < \sqrt{n} < n < n\log n < n^2 < n^3 < \dots < 2^n < 3^n < \dots < n! < n^n$$

(مسئله فروشنده دوره‌گرد (TSP

در مسئله فروشنده دوره‌گرد یک فروشنده می‌خواهد مسیر خود را از یک شهر شروع کند و از تمام شهرهای موجود فقط یک بار عبور کند و مجدداً به همان شهر برگردد. هدف مسئله این است که این فروشنده کمترین مسافت ممکن را طی نماید.



$B \to A \to D \to C \to B$

$A \to C \to D \to B \to A$

$n = 1000 \qquad 2^n = 2^{1000} = (2^{10})^{100} = (1.024)^{100} \simeq (1000)^{30} = (10^3)^{100} = 10^{300}$

Scanned by CamScanner

Subject: Date:

(۱ آبی)

① جستجوی خطی (مرتب و نامرتب)

② جستجوی دودویی (مرتب)

مرتب ← صعودی
نزولی مرتب شده اند

```
# dfine n 10
# include <iostream.h>
 void linear - search (int x [ ], int K)
 {
   int i;
  for( i=0; i<n; i++)
  if ( K = = x[i]) break;
  if (i = =n) Cout << "The number is not exist in carray";
  else cout << " The number is " << i+1 << " the element ";
 }
 int main()
 {
 int x [n], K, i;
 Cout << "enter 10 numbers ";
  for ( i =0; i<n; i++)
    Cin >> x[i];
    Cout << " Enter a number to sarch ";
     Cin >> K;
 MICRO linear - Sarch(x,K);
      return 0;
     }
```

Scanned by CamScanner

Subject: Date:

تابع ضرب را به صورت تابع بازگشتی بنویسید $a*b = \begin{cases} a & \text{if } b=1 \\ a + a*(b-1) & \text{if } b>1 \end{cases}$

```
int multiply (int a, int b)
{
if(b==1) return a,
else
return (a + multiplay (a, b-1));
}
```

تابع فیبوناچی را به صورت تابع بازگشتی بنویسید

```
int Fiba (int n)
{
if ((n==1) || (n==2))
return 1;
else
return Fiba(n-1) + Fiba(n-2);
}
```

Fiba(4) = ۳

Fiba(۳) Fiba(۲)

Fiba(۲) Fiba(۱)

۱ ۱

Fib(۵)

Fib(۴) Fib(۳)

Fib(۳) Fib(۲) Fib(۲) Fib(۱)

Fib(۲) Fib(۱)

Fib(۲) Fib(۱)

Scanned by CamScanner

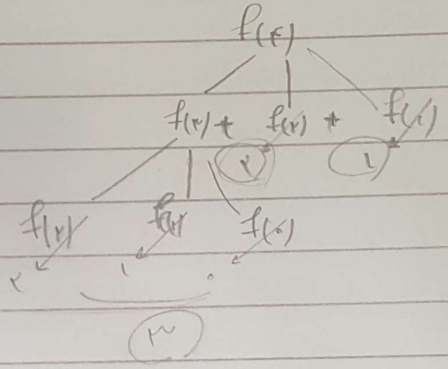
w3school.com

Subject: Date:

درخت بازگشتی تابع $f(4)$ کدام است؟

```
int f(int n){
  if(n<2) return n;
  else
  return f(n-1) + f(n-2) + f(n-3);
}
```

$f(4) = ?$




```
int p(int x, int y)
{
if(y==1)
return x;
return (x + p(x, y-1));
}
```

مقدار $p(2,3)$ چیست؟

$p(2,3)$

$x=2, y=3 \Rightarrow 2 + p(2,2) = 4$

$x=2, y=2 \Rightarrow 2 + p(2,1) = 4$ (2)

$x=2, y=1 \Rightarrow 2$

```
int fun(int a, int b)
{
if(a==1 || b==0 || a==b)
    return 1;
else
    return fun(a-1, b) + fun(a-1, b-1);
}
```

؟ $fun(4,3)$

$fun(4,3)$

$a=4, b=3$

$(3,3) + (3,2)$

MICRO

Scanned by CamScanner

«ساختمان داده»

Subject: Date:

«توابع بازگشتی»

$5! = 5 \times 4 \times 3 \times 2 \times 1!$

$5! = 5 \times 4!$

$4! = 4 \times 3!$

$3! = 3 \times 2!$

$2! = 2 \times 1!$

$n! = n \times (n-1)!$

```
int fact1(int n)
{
int i, f = 1;
for(i=1; i<=n; i++)
f = f*i;
return(f);
}
```

غیربازگشتی

```
int fact(int n)
{
if (n<=1) return 1;
else return n*fact(n-1);
}
```

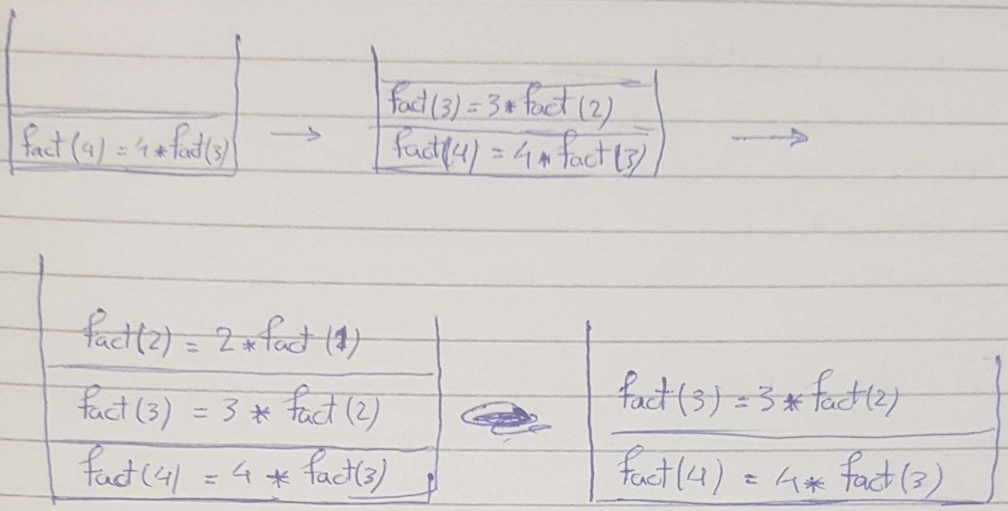
بازگشتی

MICRO

Scanned by CamScanner

Subject:

Date:



FiFo

First in First out

یعنی

Queue

LiFo

last in first out

یعنی

Stack

MICRO

@jozveotamrin

Scanned by CamScanner