طراحی الگوریتم

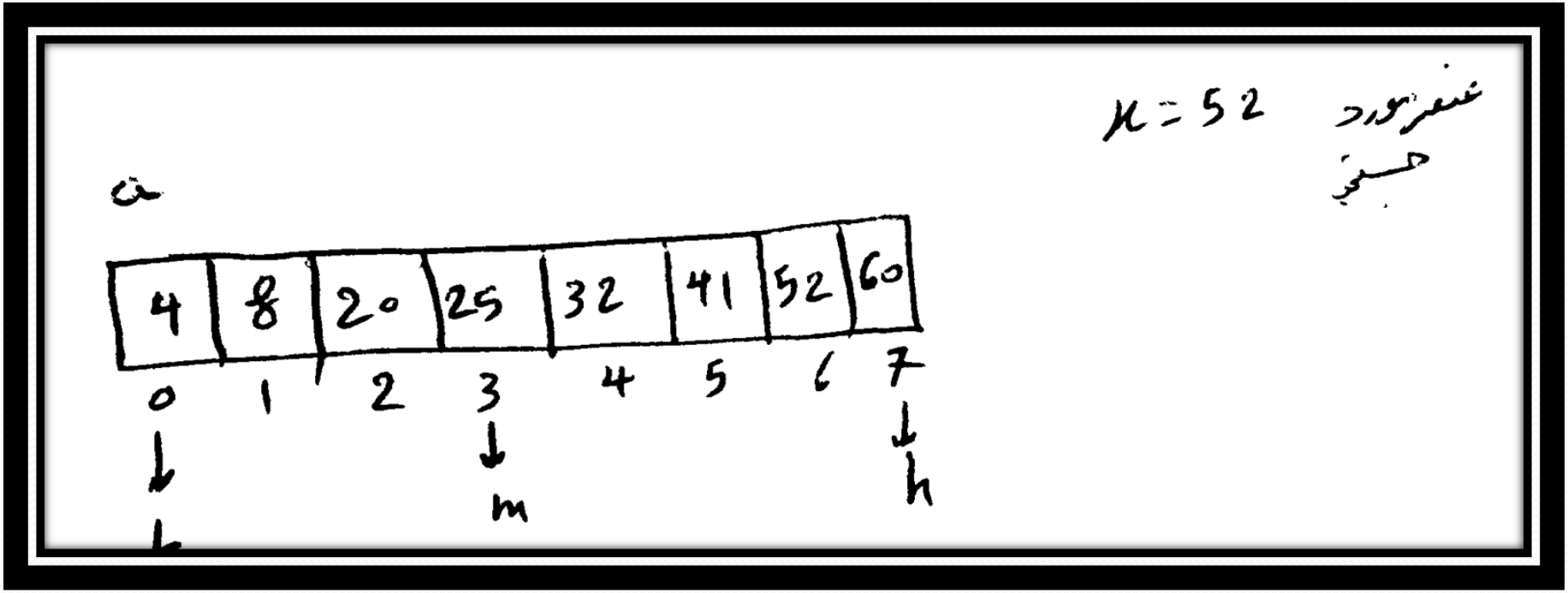
موسسه آموزش عالی پاسارگاد شیراز

جستجوی دودویی

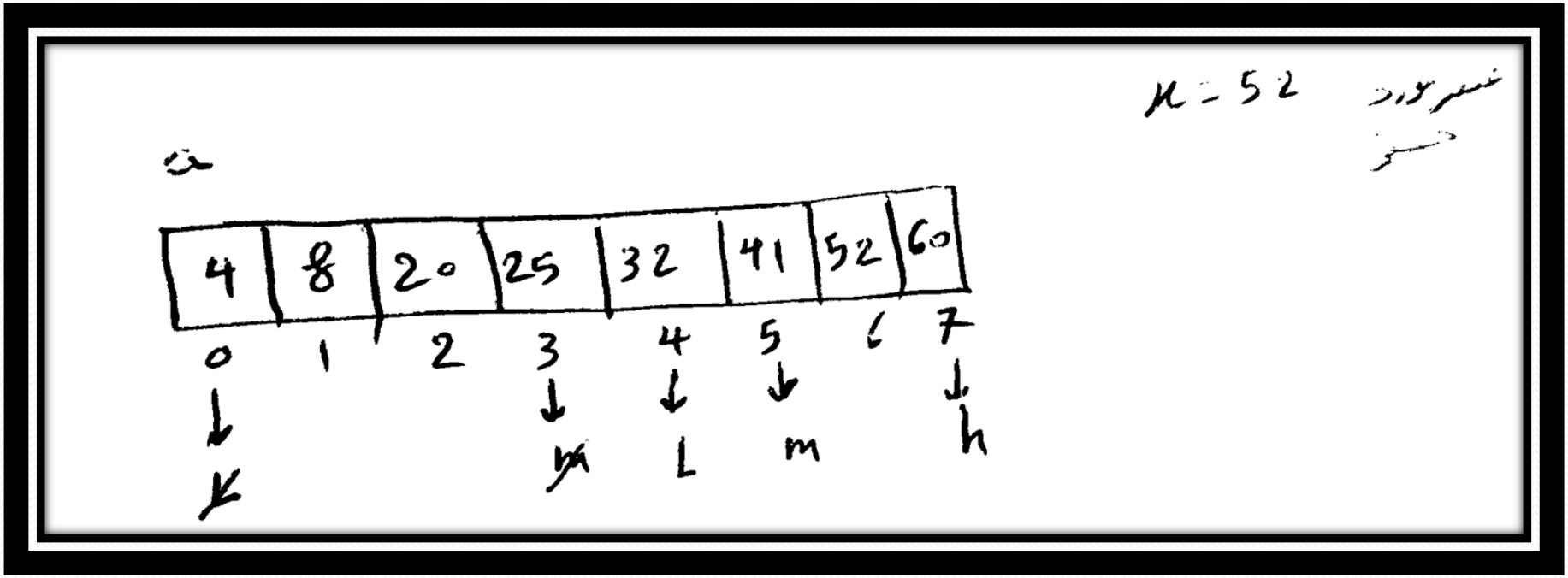
- برای انجام این نوع جستجو باید آرایه حتما مرتب باشد
- فرض کنید آرایه به صورت صعودی مرتب شده
- کد مربوطه به صورت زیر می باشد

```
int b_search(int a[n] , int x)
{
    int L=0, h=n-1, m;
    while( L<=h)
    {
        m=(L+h)/2;
        if(a[m]==x)
            return m; //
        else if(a[m]>x)
                h=m-1;
            else
                L=m+1;
    }
    return -1;
}
```

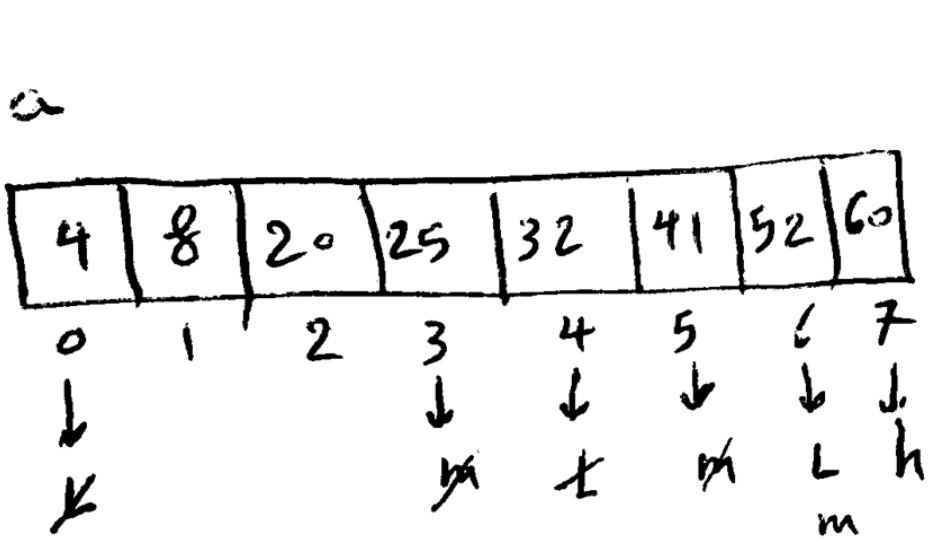
جستجوی دودویی



جستجوی دودویی



جستجوی دودویی



جستجوی دودویی

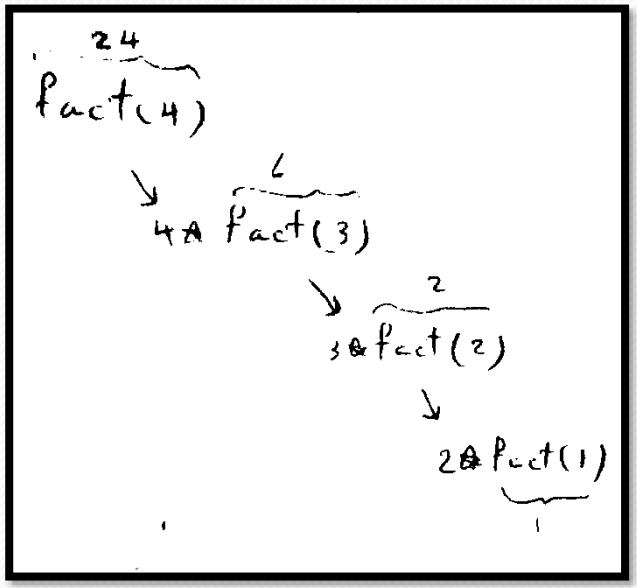
- بهترین حالت O(1)
  - عنصر مورد جستجو دقیقا وسط آرایه باشد

- حالت متوسط و بدترین حالتO(log n)
  - زیرا هر دور آرایه نصف می شود

توابع بازگشتی recursive

- توابعی که برای تعریفشان از خودشان استفاده می شود
  - نکته مهم در این توابع چک کردن شرط پایان می باشد
- مثال تابع فاکتوریل را به صورت بازگشتی بنویسید و نمودار درختی آن را برای عدد ۴ رسم کنید

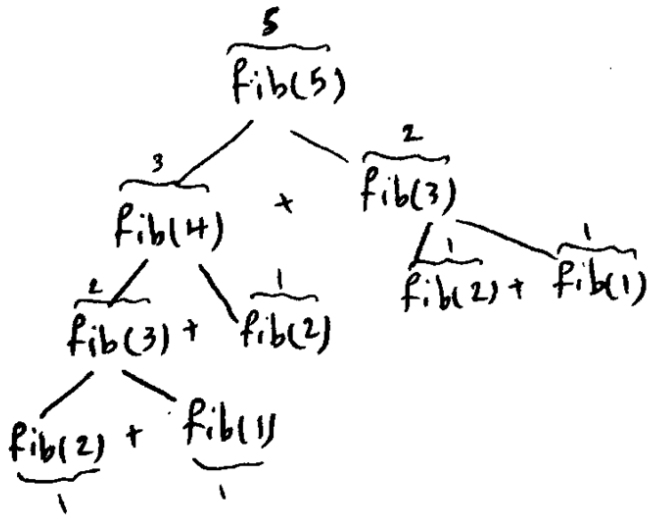
```
unsigned fact(unsigned n)
{
    if(n==0 || n==1)
    {
        return 1;
    }
    else
    {
        return n*fact(n-1);
    }
}
```



توابع بازگشتی recursive

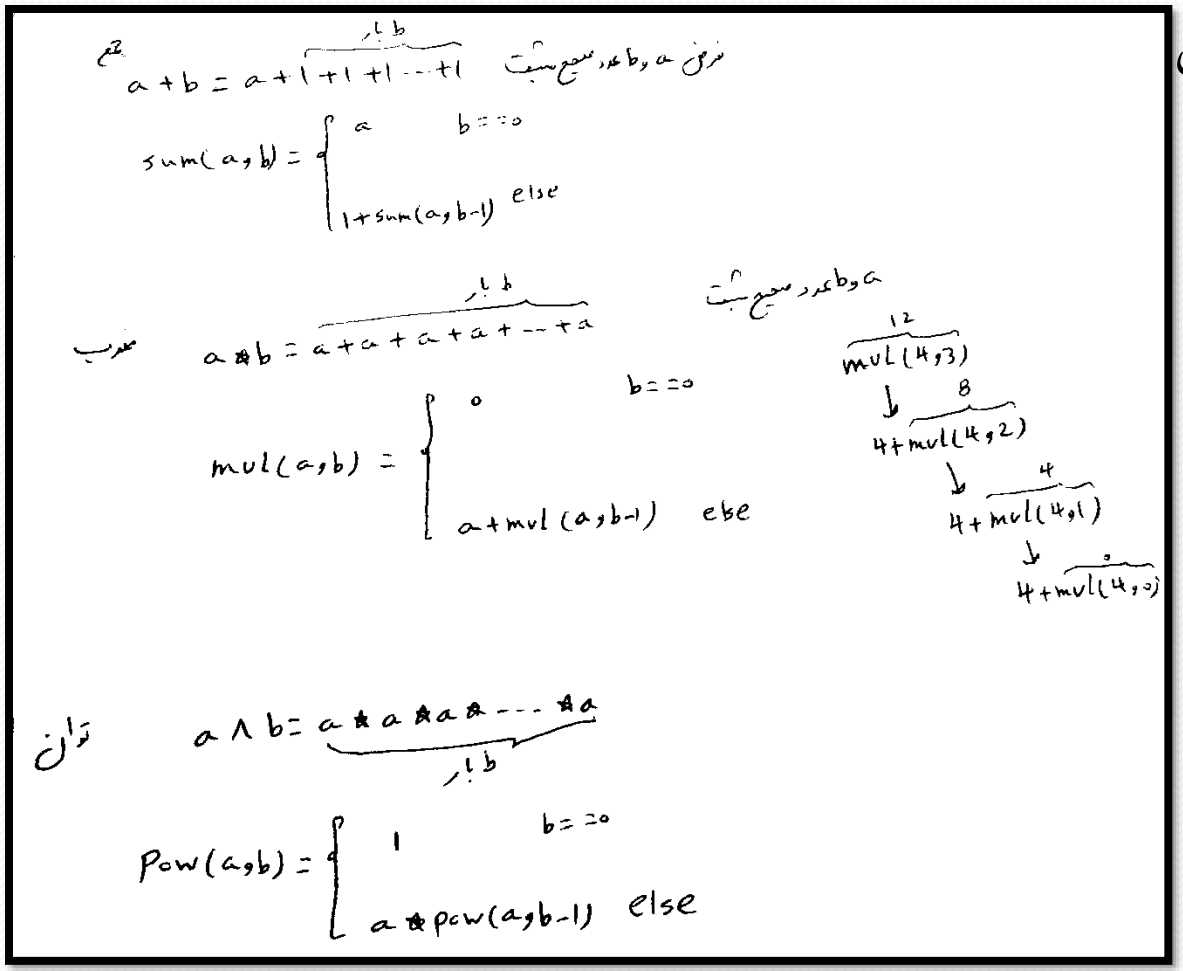
- مثال جمله **n**ام سری فیبوناچی
  - **1,1,2,3,5,8,13,21,…**

$$fib(n) = \begin{cases} 1 & n==1 \;||\; n==2 \\ fib(n-1)+fib(n-2) & n>2 \end{cases}$$

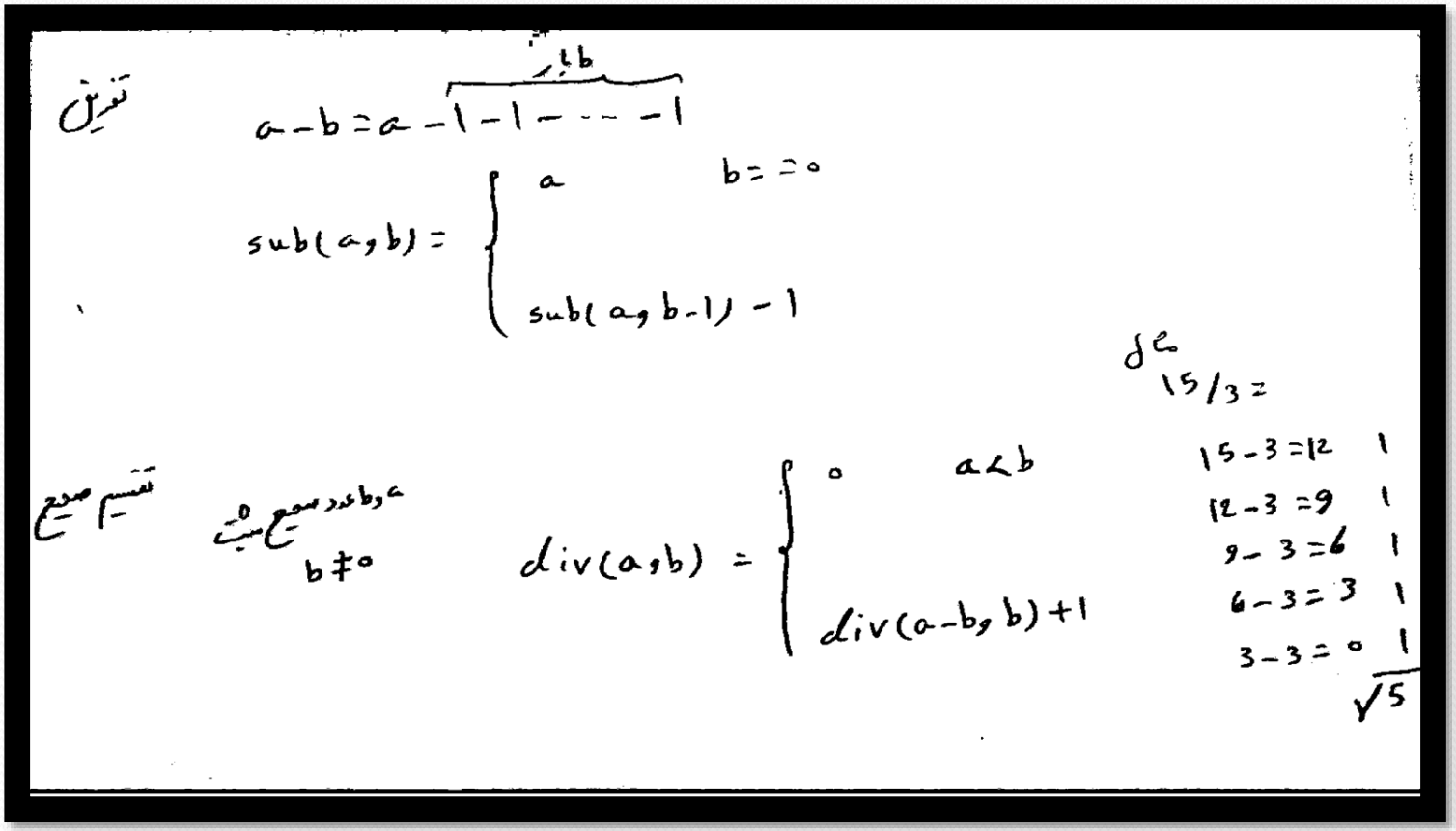


توابع بازگشتی recursive

- برخی توابع بازگشتی



توابع بازگشتی recursive



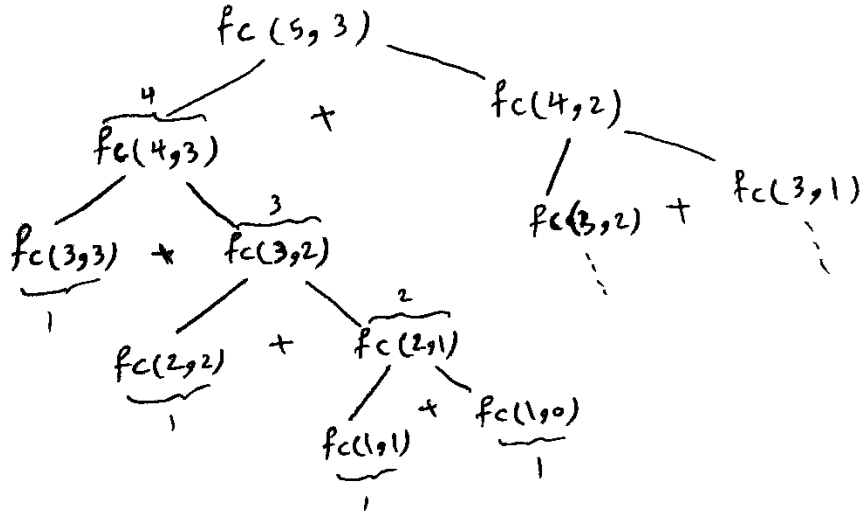
توابع بازگشتی recursive

ترکیب

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \qquad \binom{n}{n} = 1 \qquad \binom{n}{0} = 1$$

$$fc(n,k) = \begin{cases} 1 & n==k \,||\, k==0 \\ fc(n-1,k) + fc(n-1,k-1) & \end{cases}$$

نمودار درختی



توابع بازگشتی recursive

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

unsigned fc(unsigned n, unsigned k)
{
    if (n==k || k==0)
        return 1;
    else
        return fc(n-1,k) + fc(n-1, k-1)
}
```

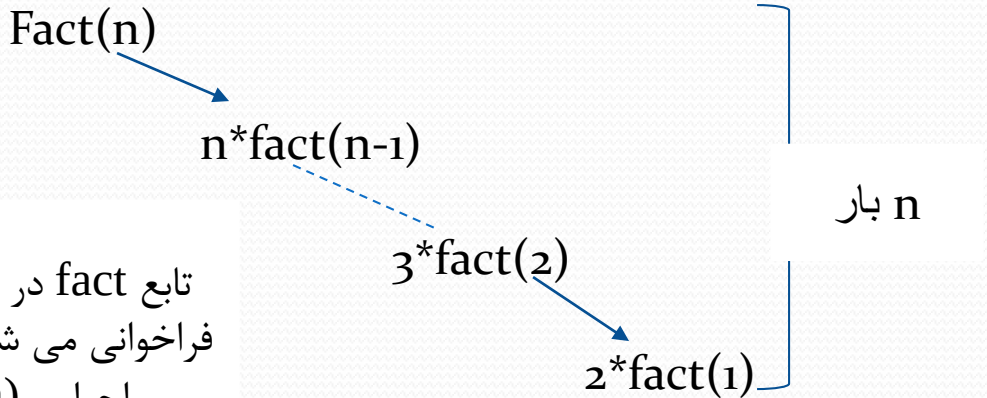
مرتبه اجرایی توابع بازگشتی

- برای بدست آوردن مرتبه ی اجرایی توابع بازگشتی باید بررسی کنیم که تابع بازگشتی به ازاء ورودی n چند بار خودش را فراخوانی می کند.
- روش کلی برای به دست آوردن مرتبه اجرای توابع بازگشتی استفاده از ریاضیات می باشد به این صورت که ابتدا رابطه ی بازگشتی را به صورت معادلات بازگشتی نوشته و سپس معادله ی بازگشتی را حل می کنیم تا مرتبه ی اجرایی به دست آید.
- به عنوان مثال در مورد سری فیبوناچی معادله ی بازگشتی به صورت زیر است.

$$Fib(n)=Fib(n-1)+Fib(n-2) \Rightarrow a_n= a_{n-1} + a_{n-2}$$

مرتبه اجرایی توابع بازگشتی

- برای به دست آوردن مرتبه ی اجرایی توابع بازگشتی می توان از روش های دیگری مثل نمودار درختی و یا روش جایگزینی استفاده کرد.
- مثال : به کمک روشهای نمودار درختی و جایگزینی مرتبه ی اجرایی تابع فاکتوریل را به دست آورید.



تابع fact در این نمودار n بار فراخوانی می شود بنابراین مرتبه اجرایی o(n) میشود

مرتبه اجرایی توابع بازگشتی

- روش جایگزینی :

```
if (n==1 || n==0)
    return  1;
else
    return  n*fact(n-1);
```

$$T(n)=\begin{cases}1 & n==1 \;||\; n==0\\ T(n-1)+1 & n>1\end{cases}$$

$$T(n)=T(n-1)+1=T(n-2)+1+1=T(n-3)+1+1+1=.....=T(n-n+1)+1+1+1...+1=T(1)+1+1+1....+1=n$$

$T(n-1)=T(n-2)+1$

$T(n-2)=T(n-3)+1$

بنابراین مرتبه اجرایی $o(n)$ میشود

مرتبه اجرایی توابع بازگشتی

- مثال

$$T(n)=\begin{cases}1 & n==1 \;||\; n==0\\ T(n-1)+T(n-1)+1 & n>1\end{cases}$$

$T(n)=2T(n-1)+1=2*(2T(n-2)+1)+1 = 2^2T(n-2)+2+1=2^3T(n-3)+2^2+2+1=2^n+2^{n-1}+\ldots+2^2+2+1$

$T(n-1)=2T(n-2)+1$

$T(n-2)=2T(n-3)+1$

تصاعد هندسی

تعداد جملات تقریبا n

$q=2$

مرتبه اجرایی

$o(2^n)$

مجموع جملات $= \frac{\text{جمله اول}(q^{\text{تعداد جملات}}-1)}{q-1} = \frac{1*(2^n-1)}{2-1} = 2^n$

مرتبه اجرایی توابع بازگشتی

- مثال

$$T(n)=\begin{cases}1 & n<5\\ T(n)=T(n-2)+1 & n>5\end{cases}$$

$$T(n)=T(n-2)+1=T(n-4)+1+1=T(n-6)+1+1+1=\ldots.=T(4)+1+1+1\ldots.+1=\frac{n-4}{2}+1$$

$T(n-2)=T(n-4)+1$

$T(n-4)=T(n-6)+1$

$n-(n-4)=4$

بنابراین مرتبه اجرایی $o(n)$ میشود

مرتبه اجرایی توابع بازگشتی

- تمرین

$$T(n)=\begin{cases}1 & n<7\\ 3\,T(n-1)+1 & n>7\end{cases}$$

$$T(n)=\begin{cases}1 & n<5\\ T(n-3)+1 & n>5\end{cases}$$

$$T(n)=\begin{cases}1 & n<2\\ T(n-1)+n & n>2\end{cases}$$

مرتبه اجرایی توابع بازگشتی

- مقایسه ی زمان اجرای تابع فیبوناچی به دو روش بازگشتی و غیر بازگشتی

```
unsigned fib1(unsigned n)
{
    if (n == 1 || n == 2)
    {
            return 1;
    }
     else
    {
            return fib1(n - 1) + fib1(n - 2);
     }
}
```

```
unsigned fib2(unsigned n)
{
   if (n == 1 || n == 2)
   {
            return 1;
    }
     else
     {
            int f1 = 1, f2 = 1, f3;
            for (int i = 3; i <= n; i++)
            {
                f3 = f1 + f2;
                f1 = f2;
                f2 = f3;
            }
            return f3;
      }
}
```

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

قضیه اول

$$T(n)= \begin{cases} C_1 & \text{شرط پایان} \\ aT(\frac{n}{b}) + C_2 & else \end{cases}$$

مرتبه اجریی =

$$\begin{cases} \theta(log_b^n) & a == 1 \\ \theta\left(n^{log_b^a}\right) & a \neq 1 \end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}2 & n==1\\ T(\frac{n}{5})+3 & else\end{cases}$$

a=1

b=5

$$\theta(log_5^n)$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}1 & n==1\\ 3T(\frac{n}{5})+1 & else\end{cases}$$

a=3

b=5

$$\theta\left(n^{log_5^3}\right)$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}4 & n==1\\ 3T(\frac{2n}{5})+1 & else\end{cases}$$

a=3

b=$\frac{5}{2}$

$$\theta\left(n^{log_{\frac{2}{5}}^{3}}\right)$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

قضیه دوم

$$T(n)=\begin{cases} C_1 & \text{شرط پایان} \\ aT(\frac{n}{b}) + C_2 n^k & else \end{cases}$$

مرتبه اجریی =

$$\begin{cases} \theta\left(n^{log_b^a}\right) & log_b^a > k \equiv a > b^k \\ \theta\left(n^k\right) & log_b^a < k \equiv a < b^k \\ \theta\left(n^k \log n\right) & log_b^a = k \equiv a = b^k \end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}2 & n==1\\ 3T(\frac{n}{5})+7n^2 & else\end{cases}$$

a=3
b=5
k=2

$$\text{مرتبه اجریی} = \begin{cases}\theta\left(n^{log_5^3}\right) & log_5^3 > 2 \equiv 3 > 5^2\\ \theta(n^2) & log_5^3 < 2 \equiv 3 < 5^2 \ \surd\\ \theta(n^2 \log n) & log_5^3 = 2 \equiv 3 = 5^2\end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}1 & n==1\\ 2T(\frac{n}{2})+n & else\end{cases}$$

a=2
b=2
k=1

$$\text{مرتبه اجریی} = \begin{cases}\theta\left(n^{log_2^2}\right) & log_2^2 > 1\\ \theta(n) & log_2^2 < 1\\ \theta(n\log n) & log_2^2 = 1 \quad \surd\end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}4 & n==1\\ 5T(\frac{n}{2})+n^2 & else\end{cases}$$

a=5
b=2
k=2

$$\text{مرتبه اجرایی}=\begin{cases}\theta\left(n^{log_2^5}\right) & log_2^5>2 \quad \surd\\ \theta(n^2) & log_2^5<2\\ \theta(n^2\log n) & log_2^5=2\end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

قضیه اصلی

$$T(n)=\begin{cases} C & \text{شرط پایان} \\ aT(\frac{n}{b}) + f(n) & else \end{cases}$$

مرتبه اجرایی =

$$\begin{cases} \theta\left(n^{log_b^a}\right) & \theta\left(n^{log_b^a}\right) > \theta(\text{f(n)}) \\ \theta(\text{f(n)}) & \theta\left(n^{log_b^a}\right) < \theta(\text{f(n)}) \\ \theta(\theta(\text{f(n)}) * \log n) & \theta\left(n^{log_b^a}\right) = \theta(\text{f(n)}) \end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}3 & n==1\\ 2T(\frac{n}{3})+n^2+1 & else\end{cases}$$

a=2

b=3

$\theta(f(n))=n^2$

$$\text{مرتبه اجرایی}=\begin{cases}\theta\left(n^{log_3^2}\right) & \theta\left(n^{log_3^2}\right)>n^2\\ n^2 & \theta\left(n^{log_3^2}\right)<n^2 \quad \surd\\ \theta(n^2*\log n) & \theta\left(n^{log_3^2}\right)=n^2\end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}10 & n==1\\ 8T(\frac{n}{2})+n^3+n & else\end{cases}$$

a=8

b=2

$\theta(f(n))=n^3$

$$\text{مرتبه اجرایی} = \begin{cases}\theta\left(n^{log_2^8}\right) & \theta\left(n^{log_2^8}\right) > n^3\\ n^3 & \theta\left(n^{log_2^8}\right) < n^3\\ \theta(n^3 * \log n) & \theta\left(n^{log_2^8}\right) = n^3 \quad \surd\end{cases}$$

قضایای مربوط به پیدا کردن مرتبه اجرای توابع بازگشتی

- مرتبه اجرایی تابع زیر را حساب کنید

$$T(n)=\begin{cases}1 & n==7\\ 5T(\frac{n}{4})+\log n+3 & else\end{cases}$$

a=5

b=4

$\theta(f(n))=\log n$

$$\text{مرتبه اجرایی}=\begin{cases}\theta\left(n^{log_4^5}\right) & \theta\left(n^{log_4^5}\right)>log\,n \surd\\ \log n & \theta\left(n^{log_4^5}\right)<log\,n\\ \theta(\log n*\log n) & \theta\left(n^{log_4^5}\right)=log n\end{cases}$$