تابع بازگشتی(recursive) :

تابع بازگشتی دارای دستوری است که تابع را فراخوانی می کند و تعداد دستورات محدودی را اجرا می کند و دارای شرط خاتمه است.

مثالهایی از تابع بازگشتی:

فاکتوریل

مجموع اعداد 1 تا n

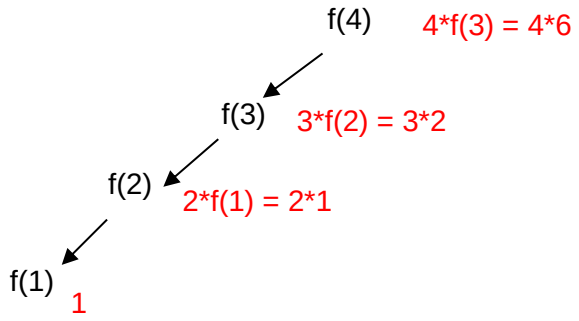
توان

فیبوناچی

تابع بازگشتی فاکتوریل:

$$f(n)=\begin{cases} 1 & n=1 \\ n*f(n-1) & n>1 \end{cases}$$

```
long int f(int  n)
{  if(n==1)
       return 1;
    else
        return n*f(n-1);
}
```



تابع بازگشتی مجموع اعداد 1 تا n :

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

```
long int sum(int  n)
{  if(n==1)
       return 1;
    else
        return n+ sum(n-1);
}
```

sum(4) 4+sum(3) = 4+6

sum(3) 3+sum(2) = 3+3

sum(2) 2 +sum(1) = 2+1

sum(1) 1

تابع بازگشتی توان :

$$pow(n,m)=\begin{cases} n & m=1 \\ n*pow(n,m-1) & m>1 \end{cases}$$

```
long int pow(int  n, int m)
{  if(m==1)
       return n;
    else
        return n* pow(n,m-1);
}
```

pow(3,4) 3*pow(3,3) = 3*27

pow(3,3) 3*pow(3,2) = 3*9

pow(3,2) 3*pow(3,1) = 3*3

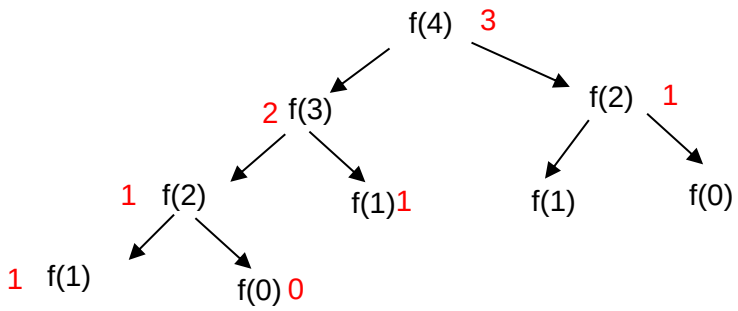
pow(3,1) 3

تابع بازگشتی فیبوناچی

0,1,1,2,3,5,8,13,21,34,....

$$f(n)=\begin{cases} 0 & n=0 \\ 1 & n=1 \\ f(n-1) * f(n-2) & n>=2 \end{cases}$$

```
int f(int  n)
{  if ((n==0) || (n==1))
       return n;
    else
        return f(n-1)+ f(n-2);
}
```



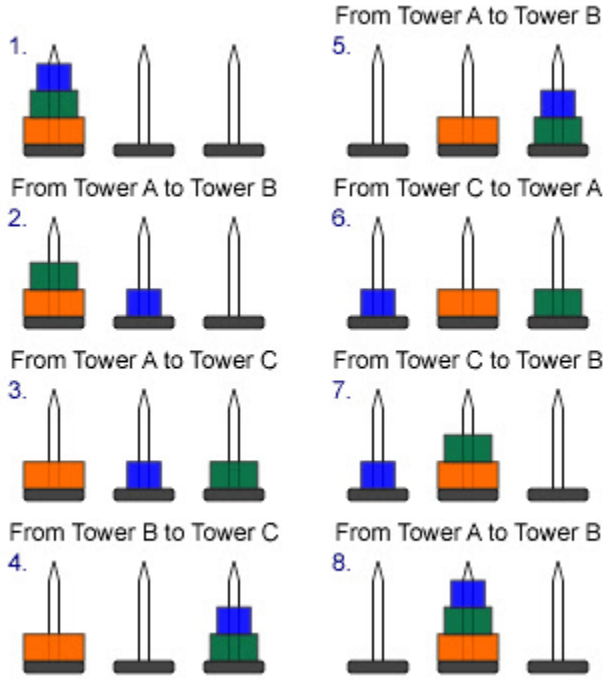
برج هانوی

سه میله ی A,B,C وجود دارد که n مهره روی میله ی A قرار دارد. هدف انتقال n مهره به میله ی C است که برای این کار از میله ی B کمک گرفته می شود.

مهره ها در میله ی A از بزرگ به کوچک مرتب شده اند. در انتقال مهره ها هرگز نباید مهره ی بزرگتر بر روی مهره ی کوچکتر قرار گیرد. در هر بار انتقال، فقط یک مهره جابجا می گردد.

حل مسئله:

1) **انتقال n-1 مهره از میله A به میله B**
2) **انتقال مهره ی n ام از میله ی A به میله ی C**
3) **انتقال n-1 مهره از میله ی B به میله ی C**




```
tower(n,A,B,C)
{
 if(n==1)
 A  to C;
else{
        tower(n-1,A,C,B);
        A  to C;
        tower(n-1,B,A,C);
}
}
```

انتقال 3 مهره از میله ی A به میله ی C به کمک میله ی B: T(3,A,B,C)

$$T(2,A,C,B)=\begin{cases} T(1,A,B,C)=A\to C & (1) \\ A\to B & (2) \\ T(1,C,A,B)=C\to B & (3) \end{cases}$$

$A\to C \quad (4)$

$$T(2,B,A,C)=\begin{cases} T(1,B,C,A)=B\to A & (5) \\ B\to C & (6) \\ T(1,A,B,C)=A\to C & (7) \end{cases}$$

رابطه ی بازگشتی برای توابع بازگشتی:

یک رابطه ی بازگشتی برای تعداد ضربها در تابع محاسبه ی فاکتوریل بنویسید؟

```
long int f(int  n)
{  if(n==1)
       return 1;
   return n*f(n-1);
}
```

حل: برای یک n معین تعداد ضربهایی که انجام می شود برابر است با تعداد ضربهای انجام شده در فراخوانی f(n-1) به اضافه یک عمل ضرب دیگر.

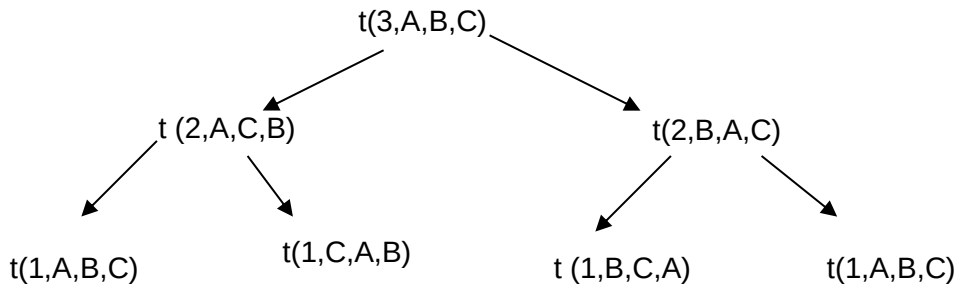
T(n)=T(n-1)+1

وقتی n=0 باشد، هیچ ضربی انجام نمی شود: T(0)=0

رابطه بازگشتی برای مسئله ی برج هانوی

تابع دوبار خودش را فراخوانی می کند.

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

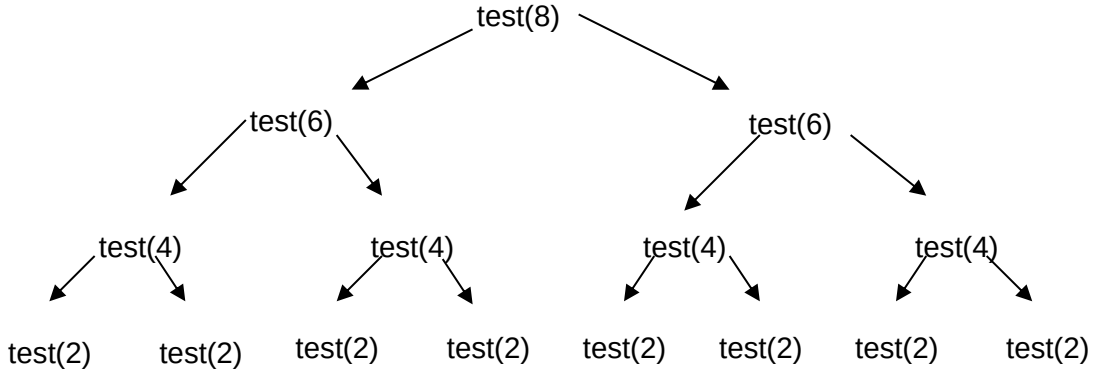


$O(2^n)$

```
int test (int  n)
{
  if (n<= 2)
      return  1;
  else
     return  test(n-2)  * test(n-2);
}
```

از کل بازه n، دنباله n-2، n-4 ، ... تا 2 (نصف بازه) مورد توجه قرار می گیرد.

در هر بار اجرا، تعداد احضارها دو برابر می شود



$O(2^{n/2})$

مثال: مقدار تابع بازگشتی زیر را به ازای a(1,1) بدست آورید.

```
int a(int n, int m)
{ if(n==0)
    return  m+1;
  else if (m==0)
   return a(n-1,1);
  else
   return a(n-1, a(n,m-1));
}
```

a(1,1)

=a(0, a(1,0))

=a(0, a(0,1))

=a(0,2)

=3