الگوریتم های بازگشتی:

الگوریتمی را بازگشتی گویند هرگاه حداقل دارای یک دستورالعمل باشد که خودش را فراخوانی کند.

ویژگی های الگوریتم بازگشتی:

- زیر برنامه خودش ، خودش را صدا بزند (فراخوانی کند)
- یک شرط جهت انجام فراخوانی ها وجود دارد.
- شرط خاتمه
- فرمول یا قانون بازگشتی

نکته

روش غیر بازگشتی را روش پویا نیز میگویند

مثال

برنامه ای بنویسید که عددی را خوانده و با کمک تابع غیر بازگشتی و تابع بازگشتی n فاکتوریل را محاسبه و آن را چاپ کند ؟

```
main ()
{
        int x ;
        cout << fact 1( x ) ;
        cout << fact ( x ) ;
}
//---------- //تابع غیر بازگشتی ----------
{
        int  i , f = 1 ;
        for ( i = 1 ,  i <= n ; i++ )
        f = f  * i ;
        return (f);
}
//---------- //تابع بازگشتی ----------
int fact ( int n )
{
        if ( n <= 1 )
        return 1 ;
        else
        return n * fact ( n-1 ) ;
}
```

روش بازگشتی با روش معمولی سادگی برنامه نویسی آن است و عیب آن مصرف زیاد حافظه است و نیز زمان زیادی است که برای فراخوانی ها صرف می شود.

برنامه ای بنویسید که مقدار a^n را محاسبه کند؟

برنامه ای بنویسید که فیبوناچی n را محاسبه کنید ؟

برج هانوی

سه میله(برج) به نام های A و B و C وجود دارند که در ابتدا بر روی میله A، n حلقه با اندازه های متفائت قرار دارند که بزرگترین حلقه در پایین و به ترتیب تا بالا که کوچکترین حلقه قرار دارد. باید این n حلقه از میله A با کمک میله B به میله C منتقل شود؛ با این شرایط که در هر انتقال فقط یک حلقه منتقل شود و هیچگاه حلقه بزرگ روی حلقه کوچک قرار نگیرد.

برج هانوی

```
Void   Tower(int n, peg A, peg B, peg C)
{
        if (n==1)
            Move top disk on A to C;
        else
        {
            Tower(n-1,A,B,C)
           Move top disk on A to C;
           Tower(n-1,B,A,C)
        }
}
```

برج هانوی

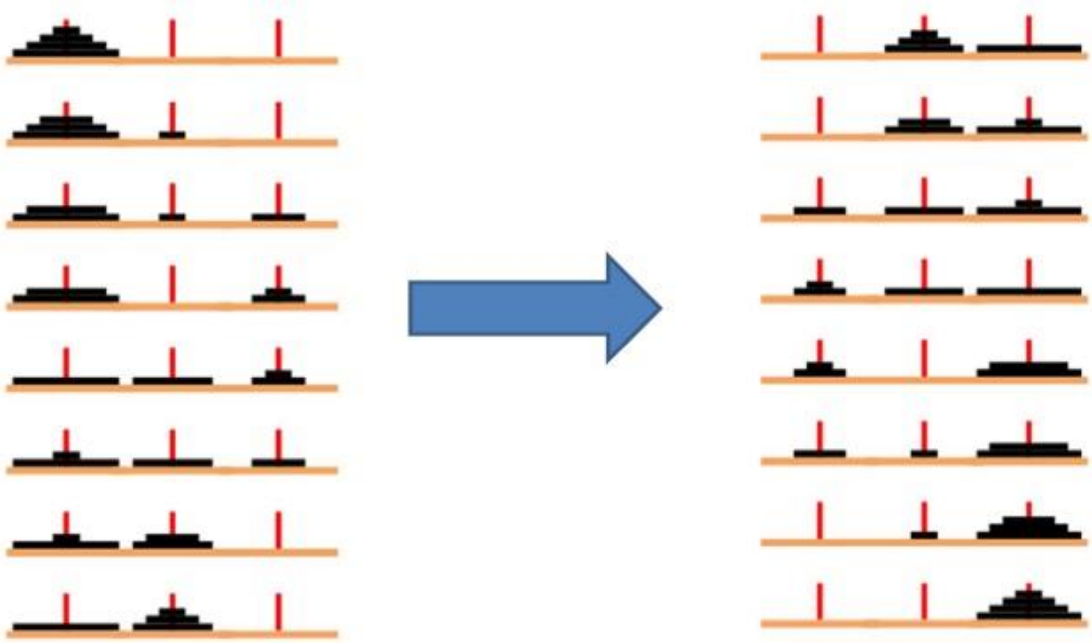
می توان مسئله برج هانوی را به سه مسئله کوچکتر تقسیم کرد:

۱- انتقال n-1 دیسک از میله A به میله B با کمک میله C

۲- انتقال یک حلقه از میله A به C

۳- انتقال n-1 حلقه از میله B به میله C با کمک میله A

برج هانوی



برج هانوی

اگر تعداد انتقال مسئله برج هانوی را برای n دیسک بابر T(n) فرض کنیم با توجه به ۳ مرحله نوشته شده داریم:

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

پس رابطه بازگشتی مسئله برج هانوی را می توان به صورت زیر نوشت. دقت کنید $T(1)=1$ زیرا برای یک دیسک، یک انتقال انجام می شود:

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

می توان با جایگذاری و استقراء رابطه بازگشتی فوق را حل کرده و به یک رابطه غیربازگشتی برسیم:

$T(1)=1$

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

$T(3)=2T(2)+1=7=2^3-1$

.

.

.

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