

فرادرس

فراتر از یک کلاس درس  
www.faradars.org

# ساختمان داده ها (توابع بازگشتی)

مدرس:

فرشید شیرافکن

دانشجوی دکتری دانشگاه تهران

(کارشناسی و کارشناسی ارشد: کامپیوتر نرم افزار) (دکتری: بیوانفورماتیک)

## تعریف

تابع بازگشتی (recursive) :

تابعی است که حاوی حداقل یک دستور باشد که خود تابع را صدا بزند. این تابع به تعداد مراحل محدودی اجرا می شود و پس از آن متوقف می شود.

مثال هایی از توابع بازگشتی :

۱- فاکتوریل

۲- مجموع اعداد ۱ تا  $n$

۳- توان

۴- ترکیب

۵- خارج قسمت تقسیم صحیح

۶- آکرمان

۷- هانوی

۸- فیبوناچی

۹- زاد و ولد خرگوش ها

## تابع فاکتوریل

تابع زیر فاکتوریل  $n$  را محاسبه می کند:

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

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

مثال: فراخوانی تابع با مقدار ۴ :

$$f(4)=4*f(3)=4*6=24$$

$$f(3)=3*f(2)=3*2=6$$

$$f(2)=2*f(1)=2*1=2$$

## مجموع اعداد 1 تا n

تابع زیر مجموع اعداد 1 تا n را محاسبه می کند:

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

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

خروجی فراخوانی تابع به ازای n=3 برابر 6 است:

$$sum(3) = 3 + sum(2) = 3 + 3 = 6$$

$$sum(2) = 2 + sum(1) = 2 + 1 = 3$$

## تابع توان

تابع زیر مقدار  $n^m$  را محاسبه می کند: (ورودی ها صحیح و مثبت هستند)

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

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

به طور نمونه تابع را به صورت  $f(3,4)$  فراخوانی می کنیم:

$$f(3,4) = 3 * f(3,3) = 3*3*3*3$$

$$f(3,3) = 3 * f(3,2) = 3*3*3$$

$$f(3,2) = 3 * f(3,1) = 3*3$$

## تابع ترکیب

تابع  $f$  ترکیب  $m$  از  $n$  را محاسبه می کند:

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

$$\binom{n}{m} = 1 \quad \text{if } m = 0 \text{ or } m = n$$

$$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1} \quad \text{if } 0 < m < n$$

فراخوانی  $f(4,2)$ :

$$f(4,2) = f(3,2) + f(3,1) = 3+3=6$$

$$f(3,2) = f(2,2) + f(2,1) = 1+2=3$$

$$f(3,1) = f(2,1) + f(2,0) = 2+1=3$$

$$f(2,1) = f(1,1) + f(1,0) = 1+1=2$$

## تابع div

( محاسبه خارج قسمت تقسیم صحیح )

محاسبه خارج قسمت تقسیم صحیح **a** بر **b**:

$$f(a,b) = \begin{cases} 0 & a < b \\ f(a-b,b) + 1 & a \geq b \end{cases}$$

فراخوانی  $f(11,3)$ :

$$f(11,3) = f(8,3) + 1 = 2+1=3$$

$$f(8,3) = f(5,3) + 1 = 1+1=2$$

$$f(5,3) = f(2,3) + 1 = 0+1=1$$

**تمرین:** با تغییری در این تابع، mod را پیاده سازی کنید.

## تابع آکرمان

تابع زیر به نام آکرمان معروف است:

$$f(a, b) = \begin{cases} b + 1 & a = 0 \\ f(a - 1, 1) & b = 0 \\ f(a - 1, f(a, b - 1)) & a > 0, b > 0 \end{cases}$$

فراخوانی  $f(1, 1)$ :

$$\begin{aligned} &f(1, 1) \\ &= f(0, f(1, 0)) \\ &= f(0, f(0, 1)) \\ &= f(0, 2) \\ &= 3 \end{aligned}$$



## روابطی در تابع آکرمان

در تابع آکرمان روابط زیر برقرار است:

$$f(1, n) = 2 + (n + 3) - 3$$

$$f(2, n) = 2(n + 3) - 3$$

$$f(3, n) = 2^{n+3} - 3$$

$$f(4, n) = 2^{2^{n+3}} - 3$$

به طور نمونه :

$$f(1, 7) = 2 + (7 + 3) - 3 = 9$$

$$f(2, 7) = 2(7 + 3) - 3 = 17$$

$$f(3, 7) = 2^{(7+3)} - 3 = 1021$$

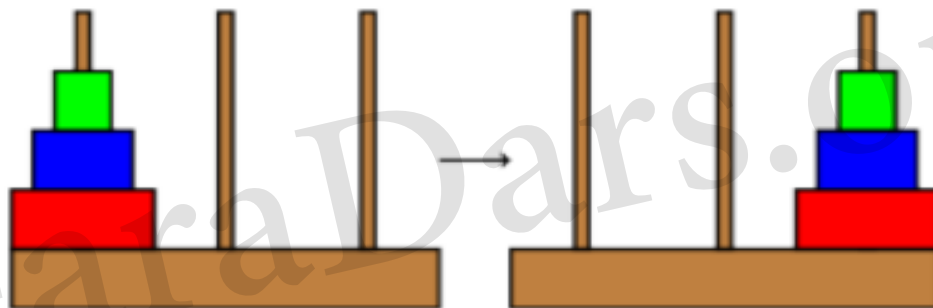
$$f(4, 1) = 2^{2^2} - 3$$

## برج هانوی

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

اندازه مهره ها در میله  $A$  از پایین به بالا کاهش می یابد.

در انتقال مهره ها هرگز مهره بزرگتر بر روی مهره کوچکتر نباید قرار بگیرد. در هر بار فقط امکان انتقال یک مهره وجود دارد که از بالای میله انتخاب می شود.



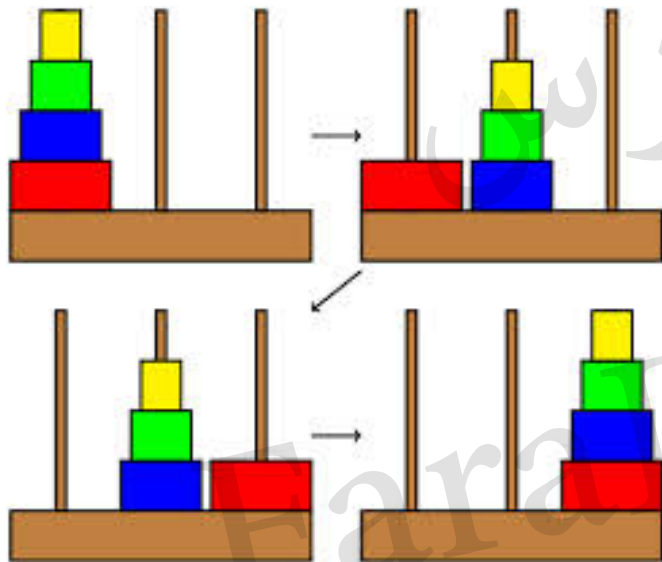
## برج هانوی (ادامه)

در صورت وجود بیش از یک مهره ، مسئله را به صورت زیر حل می کنیم:

(۱) انتقال  $n-1$  مهره از میله A به میله B.

(۲) انتقال مهره  $n$ ام از میله A به میله C.

(۳) انتقال  $n-1$  مهره از میله B به میله C.



## برج هانوی (ادامه)

الگوریتم انتقال  $n$  مهره از **A** به **C** به کمک **B**

```
tower (n , A, B, C)
```

```
{
```

```
  if (n ==1 )
```

```
    A to C;
```

```
  else {
```

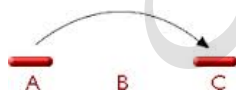
```
    tower ( n-1 , A, C, B);
```

```
    A to C;
```

```
    tower ( n-1 , B, A, C);
```

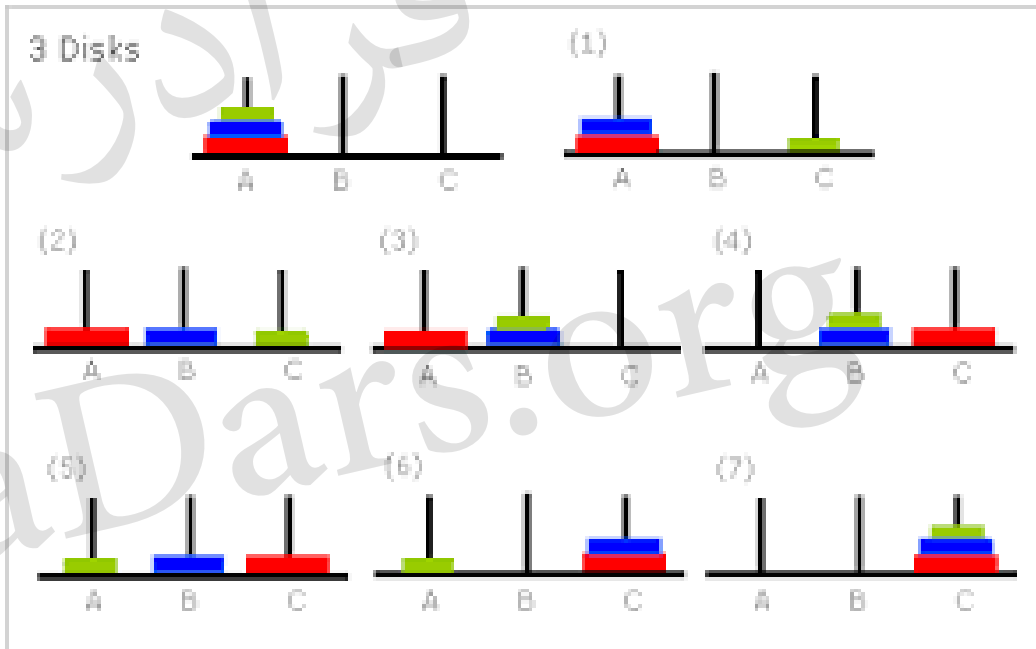
```
  }
```

```
}
```



## برج هانوی (مثال)

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



$$\begin{cases}
 T(1, A, B, C) = A \rightarrow C & (1) \\
 T(2, A, C, B) = \begin{cases} A \rightarrow B & (2) \\ T(1, C, A, B) = C \rightarrow B & (3) \end{cases} \\
 A \rightarrow C & (4) \\
 T(2, B, A, C) = \begin{cases} T(1, B, C, A) = B \rightarrow A & (5) \\ B \rightarrow C & (6) \\ T(1, A, B, C) = A \rightarrow C & (7) \end{cases}
 \end{cases}$$

## تمرین

مسئله برج هانوی را با سه میله  $A$ ،  $B$  و  $C$  در نظر بگیرید که در ابتدا  $n$  مهره به ترتیب در میله  $A$  قرار دارند. مهره کوچک تر بر روی مهره بزرگ تر قرار دارد.

می خواهیم با رعایت مقررات این مسئله، همه مهره ها را به میله  $B$  ببریم با این شرط که **نمی توان** هیچ مهره ای را یک راست از میله  $A$  به  $B$  یا از  $B$  به  $A$  برد.

اگر  $T(n)$  کمینه تعداد حرکت مهره ها برای این مسئله باشد، رابطه بازگشتی برای این مسئله را بنویسید.

FaraDars.org

## تابع فیبوناچی

محاسبه جمله  $n$  ام سری فیبوناچی : ( 0,1,1,2,3,5,8,13,21,34,... )

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

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

محاسبه جمله چهارم سری فیبوناچی :

$$f(4) = f(3) + f(2) = 2 + 1 = 3$$

$$f(3) = f(2) + f(1) = 1 + 1 = 2$$

$$f(2) = f(1) + f(0) = 1 + 0 = 1$$

## زاد و ولد خرگوش ها

در یک جزییره یک جفت خرگوش نر و ماده نوزاد وجود دارد و مدل رشد جمعیت خرگوش ها به صورت زیر است:

(۱) خرگوش ها یک ماه پس از تولد به سن بلوغ می رسند.

(۲) دوران باردای یک ماه است.

(۳) هنگامی که خرگوش ماده به سن بلوغ می رسد حتما باردار می شود.

(۴) خرگوش ماده در هر بارداری، یک خرگوش نر و یک خرگوش ماده به دنیا می آورد.

(۵) خرگوش ها هرگز نمی میرند.

رابطه بازگشتی بنویسید که تعداد خرگوش ها را در شروع ماه  $n$  ام نشان دهد؟



## زاد و ولد خرگوش ها (ادامه)

پاسخ:

تعداد جفت خرگوش ها در شروع ماه  $n$  ام برابر است با مجموع

دو مقدار زیر:

**الف-** تعداد جفت خرگوش های ماه قبل

**ب-** تعداد جفت نوزادان (که برابر است با تعداد جفت خرگوش های

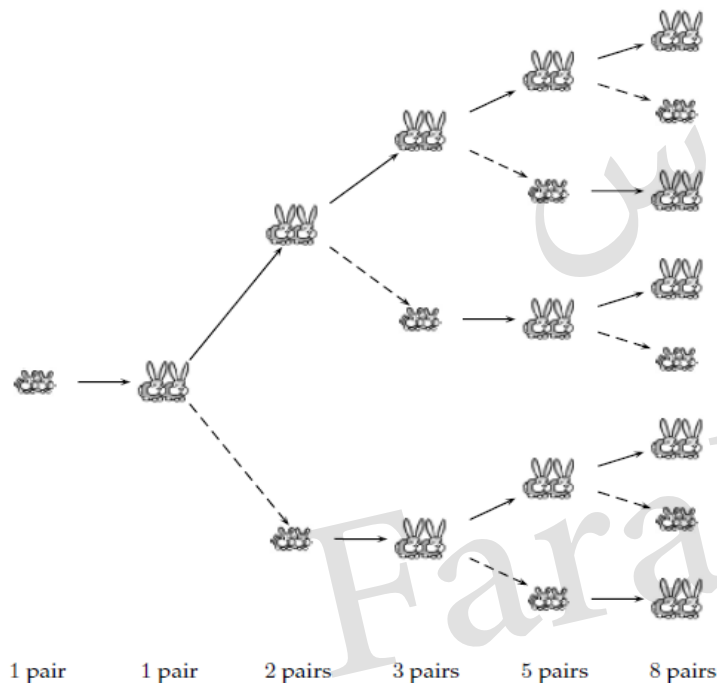
بالغ ماه قبل که آن هم برابر است با کل خرگوش های دو ماه قبل)

بنابراین داریم:

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

$$f(1) = 1, f(2) = 1$$

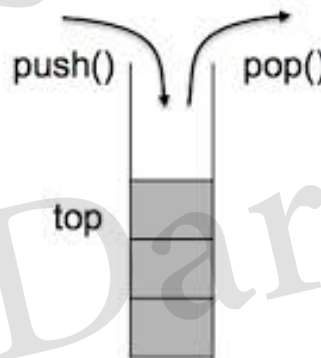
$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233$$



## کاربرد پشته در زیر برنامه های بازگشتی

تعریف پشته: (stack)

پشته ساختمان داده ای است که حذف و اضافه از بالای آن انجام می شود. پشته را **FILO** می نامند، چون آخرین عنصر وارد شده در آن، اولین عنصری است که از آن برداشته می شود.



عملیات در پشته و کاربردهای آن در فصل پشته به طور مفصل بررسی خواهند شد.

## تمرین

خروجی  $f(1,1)$  را مشخص کنید.

$$f(x,0) = f(x+1,0) + f(x+1,1), \text{ if } x < 3$$

$$f(x,1) = 2f(x+1,0) + f(x+1,1), \text{ if } x < 3$$

$$f(3,0) = 1$$

$$f(3,1) = 0$$

## مثال

خروجی زیر برنامه زیر، به ازای فراخوانی **f(1)** چیست؟

$$f(1) \Rightarrow f(2) \Rightarrow f(3)$$

```
f(n){  
    if (n==3)  
        exit( );  
    else{  
        n=n+1;  
        f(n);  
        cout<<n;  
    }  
}
```

|   |
|---|
| 3 |
| 2 |

بنابراین خروجی برابر **32** خواهد بود.

## مثال

خروجی  $f(4)$  چه می باشد؟

```
f(n){  
  if (n > 2)  
  {  
    f(n-1);  
    cout << 'a' ;  
    cout << 'b' ;  
  }  
}
```

 $f(4) \Rightarrow f(3) \Rightarrow f(2)$ 

|   |
|---|
| a |
| b |
| a |
| b |

پاسخ: خروجی **abab** می باشد.

## مثال

خروجی زیر برنامه زیر، به ازای فراخوانی  $f(1,4)$  چیست؟

$$f(1,4) \Rightarrow f(2,5) \Rightarrow f(3,6)$$

```

f(n,m){
    if (n==3)
        exit( );
    else {
        n=n+1;
        m=m+1;
        f(n,m);
        cout<<n;
        cout<<m;
    }
}

```

|   |
|---|
| 3 |
| 6 |
| 2 |
| 5 |

بنابراین خروجی برابر **3625** خواهد بود.

## تمرین

خروجی  $f(4)$  چیست؟

```
f(int x){  
    if (x)  
        g(x-1);  
    cout<<x;  
}
```

```
g(int y){  
    if (y)  
    {  
        cout<<y+1;  
        f(y-1);  
    }  
}
```

## رابطه بازگشتی برای توابع بازگشتی

یک رابطه بازگشتی برای تعداد ضرب ها در تابع فاکتوریل بنویسید.

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

حل:

برای یک  $n$  معین تعداد ضرب هایی که انجام می شود برابر است با تعداد ضرب های انجام شده در فراخوانی  $(n-1)$  به اضافه عمل ضرب  $n$  در  $fact(n-1)$ . بنابراین اگر تعداد ضرب های انجام شده برای یک مقدار معین  $n$  را با  $T(n)$  نمایش دهیم، داریم:

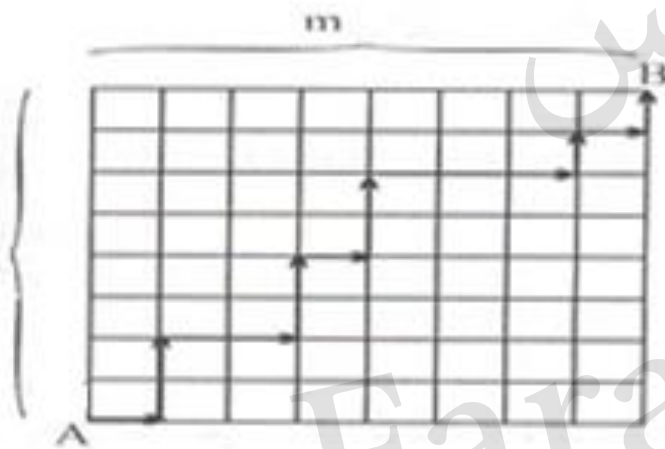
$$T(n) = 1 + T(n-1)$$

وقتی  $n=0$  باشد، هیچ ضربی صورت نمی گیرد:  $T(0) = 0$



## حرکت در صفحه شطرنجی

در صفحه شطرنجی زیر می خواهیم از نقطه  $A$  به نقطه  $B$  برویم. حرکت های مجاز فقط به سمت راست یا بالا می باشد. یک رابطه بازگشتی برای  $T(n, m)$  (تعداد مسیرهای ممکن از  $A$  به  $B$ ) بنویسید.



$$T(n, m) = T(n, m - 1) + T(n - 1, m)$$

$$T(0, m) = 1$$

$$T(n, 0) = 1$$

جواب این رابطه بازگشتی برابر است با:  $\binom{m+n}{m}$

## مثال

اگر  $T(n)$  تعداد ستاره های چاپ شده توسط  $f(n)$  باشد، رابطه بازگشتی  $T(n)$  را مشخص کنید.

```
f(n){  
  if (n>=2)  
  {  
    f(n-1);  
    f(n-1);  
    f(n-2);  
    cout<<"*";  
  }  
}
```

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

$$T(0) = 0$$

$$T(1) = 0$$

## رابطه بازگشتی برای مسئله برج هانوی

اگر به تابع برج هانوی نگاه کنیم، مشاهده می شود که تابع دو بار خودش را فراخوانی می کند:

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

جواب این رابطه برابر است با:

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

## تمرین

برای هر یک از مسئله های زیر یک رابطه بازگشتی بنویسید.

(۱) تعداد دنباله های 0 و 1 به طول  $n$  که هیچ دو صفر متوالی ندارند.

(۲) تعداد روش های فرش کردن یک صفحه شطرنجی با اندازه  $n \times 2$  با موزاییک های  $1 \times 2$

(۳) فرض کنید کدی سه تایی شامل ارقام 0 و 1 و 2 با طول  $n$  زمانی معتبر است که تعداد صفرهای ظاهر شده در آن زوج باشد. رابطه بازگشتی بنویسید که تعداد کدهای به طول  $n$  که معتبر هستند را محاسبه کند.

## روش های حل رابطه های بازگشتی

رابطه های بازگشتی را می توان به روش های زیر حل کرد:

- ۱- حدس
- ۲- تکرار با جایگذاری
- ۳- قضیه اصلی
- ۴- درخت بازگشت
- ۵- روش های حل رابطه های بازگشتی همگن و ناهمگن

این روش ها در درس **تحلیل و طراحی الگوریتم** بررسی می شود.

این اسلاید ها بر مبنای نکات مطرح شده در فرادرس  
«مجموعه فرادرس های ساختمان داده ها»  
تهیه شده است.

برای کسب اطلاعات بیشتر در مورد این آموزش به لینک زیر مراجعه نمایید.

**[faradars.org/fvds9402](https://faradars.org/fvds9402)**