## تکنیک بازگشتی :

برای مطالعه این بخش توصیه می کنیم که حتماً با استقرای ریاضی آشنا باشید و اصل استقرا را حتماً به دقت بررسی کنید.

در این طرز تفکر مساله ها با فرض حل شده بودن برای تمامی ورودیهای کوچکتر و مساوی ورودی حاضر و یا بالعکس حل می شوند. این ممکن است فکر کنید که طرز تفکر بازگشتی مربوط به توابع است ولی در واقع برای طرح هر دوی توابع و روالها مفید است بیان برای روالها به این صورت است که کار مورد نظر برای تمامی ورودیهای کوچکتر از ورودی حاضر و یا بالعکس حل شده است. منظور از بالعکس در عبارات بالا این است که برای تمامی اعداد بزرگتر از ورودی حاضر این مورد را برای پوشش اعداد منفی بیان کردیم.

ساده ترین مثالی که به ذهن می رسد تابع محاسبه فاکتوریل عدد $n$ است می دانیم که

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

لذا اگر $(n-1)!$ قابل محاسبه باشد $n!$ قابل محاسبه است.

این نوع محاسبه منوط به یک شرط است و آن این است که تعریف تابع به تسلسل منجر نشود و آن به این معنی است که حداقل یک ورودی تابع بصورت صریح ( غیر ضمنی ) تعریف شده باشد. مثلاً $0!=1$ یک تعریف صریح است. لذا تعریف بازگشتی $n!$ بصورت زیر کامل و بدون ابهام می شود:

$$n!=n(n-1),0!=1$$

$$n>0$$

برنامه فاکتوریل بصورت زیر پیاده می شود:

```
function   Factorial( X  : interger ) : interger ;
begin
if    X  < 0  then
result := 1
else
result := X * Factorial( X  −1);
end ;
```

**در راستای توسعه استفاده از روش بازگشتی توابع چند متغیره را در نظر بگیرید. در این نوع توابع هم می توان مفهوم بازگشتی را اعمال کرد. مثال تابع ترکیب را در نظر بگیرید:**

$$\binom{n}{i}=\binom{n-1}{i}+\binom{n-1}{i-1} \qquad i<n, n>1$$

$$\binom{1}{1}=1, \binom{n}{i}=1 \qquad i \geq n$$

**تعریف بالا هم کامل است. تعریف بالا بوضوح متفاوت از تعریف قبلی است.**

**در اینجا شرط تعریف ضمنی هم متفاوت با قبل است شرط آن دو گانه است $n>1$ و $i<n$. لـذا بایـد حالتی که $i \geq n$ است صریحاً تعریف شود و همچنین $n \leq 1$ هم باید صریحاً تعریف گردد. با این شرط بـاز هـم تعریف تابع کامل است:**

```
function   Combination( n ,i : interger ) : interger ;
begin
  if    n ≤ 1  then
    result := 1
  else
  begin
    if    n ≤ i   then
      result := 1
    else
      result := Combination( n −1,i −1) + Combination( n −1,i );
  end ;
end ;
```

دقت شود که هر دو شرط $n \le 1$ و $n \le i$ باید احتمالاً پردازش گردنـد و چـون شـرط $i \le n, \binom{n}{i} = 1$ شامل $\binom{1}{1} = 1$ است فکر نکنید که $n \le 1$ پوشش داده می شود.

کلاً طرح الگوریتمهای بازگشتی شیرین و لذت بخش است و در عین حال تخمین زمان برای آنها بسیار دشوار است. در ضمن حافظه *stack* صرف شده برای این نوع توابع بسیار زیاد است زیرا توابع به دفعات تودر تو فراخوانی می شوند.