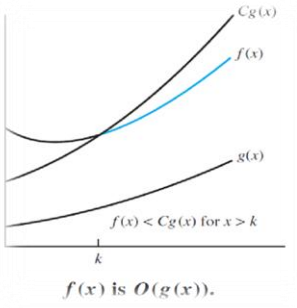
المحاضرة الثانية

The Growth of functions

8\10\2013

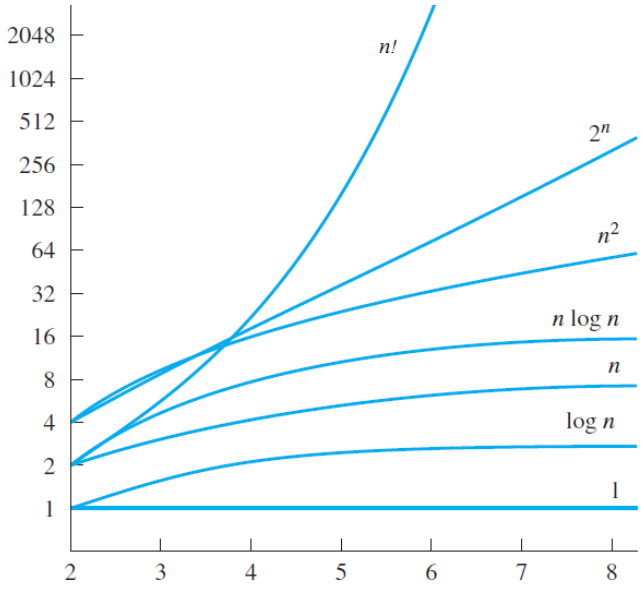
تعاريف: ليكن g و f تابعان معرفان $\mathbb{R}$ولنعرف ما يلي:

- نقول عن تابع fأنه نهمل تقريبا أمام التابع gإذا كان $\frac{f}{g} \xrightarrow[n\to\infty]{} 0$ونرمز لذلك بـ $f = o(g)$
- نقول عن التابع fأنه مكافئ لـgإذا كان $\frac{f}{g} \xrightarrow[n\to\infty]{} 1$ ونرمز لذلك بـ$f \cong g$
- نقول عن التابع fأنه مُسيطر عليه تقاريبا من قبل التابع gاذا كان
  $\exists c > 0 \;\&\; k \geq 0 : |f(x)| \leq cg(x) \;\; ; \;\; x \geq k$
  نرمز لذلك بـ $f = O(g)$
- نقول عن fأنه $f = \Omega(g)$ اذا كان
  $\exists c > 0 \;\&\; k \geq 0 : |f(x)| \geq cg(x) \;\; ; \;\; x \geq k$
- نقول عن fأنه من مرتبة gاذا كان $f = O(g) \;\&\; f = \Omega(g)$
  ونرمز لذلك بـ $f = \Theta(g)$



خواص:

1) $O(O(g) = O(g), \;\; g = O(g)$
2) $cO(g) = O(g)$
3) $O(g_1) + O(g_2) = O(\max\{g_1, g_2\})$
4) $O(g_1 . g_2) = O(g_1) . O(g_2)$
5) $O(1) \subseteq O(\log n) \subseteq O(n^{\epsilon}) \subseteq O(n) \subseteq O(n \log n\ ) \subseteq O(n^c) \subseteq$
   $\subseteq\ O(c^n)\ \subseteq O(n!) \approx O(e^c)$



تاريخ المصطلح *Big-O*

ان أول استخدم للمصطلح *Big-O*قبل اكثر من قرن حيث تستخدم بشكل واسع في علوم الكمبيوتر وتحليل الخوارزميات.أول من قدم هذا المصطلح الرياضي الالماني باول باكمانPaul Bachmann 1892 في كتاب نظرية الأعداد.وعرف هذا المصطلح في علم الكمبيوتر من قبل دونلد كينث Donald knuth كما قدم أيضا كل من *Big-*Θ و *Big-*Ω

تمرين:

اذا كان $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$كثير حدود من الدرجة $n > 0$أثبت أن $f(x) = O(x^n)$

الحل:

انطلاقا من التعريف نكتب

$$|f(x)| = |a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0|$$

$$|f(x)| \leq |a_n x^n| + |a_{n-1}x^{n-1}| + \cdots + |a_1 x| + |a_0|$$

$$= x^n \left[|a_n| + \left|\frac{a_{n-1}}{n}\right| + \cdots + \left|\frac{a_0}{x^n}\right|\right]$$

$$\leq x^n[|a_n| + \cdots + |a_1| + |a_0]$$

$$\leq cx^n \Rightarrow f(x) = O(x^n)$$

أثبت ما يلي:

$$\sum_1^n i = O(n^2) \_1$$

$$\sum_1^n i^2 = O(n^3) \_2$$

$$\sum_1^n i^k = O(n^{k+1}) \_3$$

الخوارزميات العودية

Note
تتميز الخوارزميات العودية بسهولة كتابتها.

تعريف:نقول عن هدف ما أنه عودي اذا عرف بدلالة نفسه بشكل كلي أو جزئي.

أمثلة عن العودية:

1. مجموعة الأعداد الطبيعية. الصفر عدد طبيعي كل عدد يلي الصفر هو عدد طبيعي (يلي أي بزيادة واحد).
2. العاملي يمكن تعريفها اعتمادا على العودية.
$$n! = n \times n - \times n - 2 \ldots . \times 2 \times 1$$
$$0! = 1$$
3. بعض المتسلسلات مثل اعداد فيبوناتشي.

تمرين(1) :اكتب خوارزمية تقوم بحساب$n!$ حيث n عدد طبيعي معطى واحسب تعقيد هذه الخوارزمية

الحل:

```
Int   factorial(int n)
{if(n==0)  return 1;
        else return (n*factorial(n-1));
    }
```

Note
الاستدعاء الاخير للتابع لا يوجد فيه عملية ضرب لأن قيمة التابع صفر وسيرجع واحد دون أي عملية

حساب تكلفة الخوارزمية
ان عدد مرات استدعاء التابع لنفسه (العمق العودي) $n + 1 =$ مرة وفي كل استدعاء ينفذ التابع عملية ضرب واحدة (عدا أخر استدعاء).⟸ كلفة الخوارزمية $T(n)$هي $T(n) = (n + 1) * 1 - 1 = n$
في حال لم تتمكن من حساب العمق العودي يمكن ان نحسب التكلفة بالطريقة
لدينا $T(n) = 1 + T(n-1)$ وايضا نعلم $T(0) = 0$

$$T(n) = T(n-1) + 1$$
$$= [T(n-2) + 1] + 1$$
$$= [T(n-3) + 1] + 2 = \cdots \ldots \ldots \ldots$$
$$= T(n-n) + n = T(0) + n \ = n = O(n)$$

تمرين(2):اكتب خوارزمية لحسب المقدار $x^n \;; n \geq 0 \;, x \in \mathbb{Z}$

1) طريقة تكرارية

```
Prod=1
  for i=1 to n do
    prod=prod*x;
```

وتكون تكلفة الخوارزمية بالنسبة لعملية الضرب هي $T(n) = n * 1 = n$

2) طريقة عودية

```
Int power(x,n){
        If(n==0)return 1;
         else if(n==1) return x;
           else return(x*power(x,n-1))
         };
```

وتكلفة الخوارزمية هي

$$\left.\begin{matrix} \mathrm{T(n)1 + T(n-1)} \\ \mathrm{T(1) = 0} \end{matrix}\right\} \xrightarrow{\text{السابق للتمرين مشابه بشكل}} \mathrm{T(n) = n = O(n)}$$

نلاحظ ان للطريقتين السابقتين نفس التكلفة أي هي من رتبة $O(n)$لنحالول كتابة خوارزمية اخرى ولكن بتكلفة اقل

```
Int  power(x,n){
        If(n==0)return1;
        else if (n==1) return x;
          else{
              If(n mod 2==0)
                {int z=power(x,n/2);
                   Return z;
                 }\\end of if statement.
                 else{
                     int z=power(x,(n-1)/2);
                      }\\end of else
            }\\end of else
         }\\end of the function.
```

حساب تكلفة الخوارزمية

$$T(n) = T\left(\frac{n}{2}\right) + 1 \quad \& \quad T(1) = 0$$

$$T(n) = T\left(\frac{n}{2}\right) + 1$$
$$= \left[T\left(\frac{n}{4}\right) + 1\right] + 1 \; = \left[T\left(\frac{n}{2^3}\right) + 1\right] + 2 = \cdots = T\left(\frac{n}{2^i}\right) + i$$

$$\Longrightarrow 2^i = n\} \Longrightarrow \underset{T\left(\frac{n}{2^i}\right)=T\left(\frac{n}{n}\right)=T(1)=0}{i=\log n}$$

$$T(n) = 0 + i = \log n$$

انتهت المحاضرة الثانية

يمكن الحصول على مزيد من المحاضرات بزيارة الموقع www.syriamath.com