جامعة المنارة
كلية الهندسة
قسم الهندسة المعلوماتية

مقرر الخوارزميات وبنى المعطيات1

جلسة العملي الأولى

(الفصل الثاني 2024-2025)

https://manara.edu.sy/

Algorithm Complexity تعقيد الخوارزميات

الغاية من الجلسة:

- ✓ فهم أنواع التعقيد الزمني للخوارزميات .
- ✓ القدرة على حساب التعقيد الزمني للخوارزمية .

التمرين الأول:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب عدد العمليات . 2- احسب تعقيد الخوارزمية .

```
Void main()
{
  int m=1,p=3,L=5;        // 3

  for (int i=1;i<=n;i++)  // 1+n+1+n = 2n+2

        {
              m=m+1;   // n
              p=p*2;   // n
              L=L+3;   // n
        }
}
// T(n) = 5n+5    عدد العمليات الكلي:
```

ملاحظة : إن عدد العمليات في رأس حلقة for: تعليمة إسناد + n+1 عملية مقارنة في الشرط + n عملية زيادة ل i = **1+n+1+n**

لحساب درجة تعقيد الخوارزمية نبحث عن التابع f(n) و الثوابت c,n0 كما يلي:

$$T(n) \le C.f(n) \;; for \quad all \quad n \ge n_0$$

$$T(n)=5n+5 \Rightarrow 5n+5 \le 6n$$

$$5 \le 6n-5n$$

$$5 \le n$$

وعليه فإن: $c=6$ $n_0=5$ $f(n)=n$

بالتالي نقول أن الخوارزمية ذات درجة تعقيد $O(n)$

التمرين الثاني:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب عدد العمليات 2- احسب تعقيد الخوارزمية .

```
Void main()
{
  int m=1,p=3,L=5;                // 3

  for (int i=1; i<=n; i++)         // 1+n+1+n=2n+2
       for (int j=1; j<=n; j++)  // n(2n+2)=2n^2+2n
       {
             m=m+1;          // n^2
             p=p*2;          // n^2
             L=L+3;          // n^2
       }
}
```

عدد العمليات الكلي: **// $T(n)= 5n^2+4n+5$**

لحساب درجة تعقيد الخوارزمية نلاحظ أن : $5n^2+4n+5 \le 6n^2$ وذلك كما يلي:

n	$5n^2+4n+5$	$6n^2$
1	14	6
2	33	24
3	62	54
4	101	96
5	150	150

وعليه فإن: c=6 n_0 =5 f(n)=n^2

بالتالي نقول أن الخوارزمية ذات درجة تعقيد **$O(n^2)$**

التمرين الثالث:

لدينا هاتين الخوارزميتين لطباعة عناصر القطر الرئيسي في مصفوفة ثنائية والمطلوب: قارن بين تعقيد الخوارزميتين و حدد الخوارزمية الأفضل.

for(int i=0; i<n; i++) **//2n+2**
cout<<x[i][i]; **// n**
 	
// T(n) = 3n+2
 	
//Complexity : O(n)
	for(int i=0; i<n; i++) **// 2n+2**
	for(int j=0; j<n; j++) **//n(2n+2)= 2n²+2n**
	if(i==j) **// n²**
	cout<<x[i][j]; **// n**
	// T(n) = 3n²+5n+2
	
	//Complexity : O(n²)

بالمقارنة بين تعقيد الخوارزميتين نجد أن الخوارزمية التي تعقيدها O(n) هي الأفضل.

التمرين الرابع:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب عدد العمليات   2- احسب تعقيد الخوارزمية .

```
Void main()
{
 int m=1,p=3,L=5;        // 3
 for (int i=n; i>=1; i/=2)  //  1+log n +1+log n =2+2log n
 {
 m=m+1;                  //     log n
 p=p*2;                  //     log n
 L=L+3;                  //     log n
 }
 }
// T(n) = 5log n +5
//Complecity : O(log n)
```

التمرين الخامس:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب عدد العمليات   2- احسب تعقيد الخوارزمية .

```
Void main()
{
 int m=1,p=3,L=5;            // 3
 for (int i=n; i>=1; i/=2)     // 1+log n+1+log n = 2+2log n
 for (int j=1; j<=n; j++)  // (1+n+1+n)log n =(2n+2)log n=2n*log n +2log n
 {
 m=m+1;                      // n*log n
 p=p*2;                      // n*log n
 L=L+3;                      //  n*log n
 }
 }
// T(n)= 5n*log n + 4log n + 5
//Complexity : O(n * log n)
```

جامعة المنارة
MANARA UNIVERSITY

التمرين السادس:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب عدد العمليات   2- احسب تعقيد الخوارزمية .

```
Void main()
{
 int m=1,p=3,L=5;                // 3
 for (int i=1; i<=n/2; i++)   // 1+n/2 +1+n/2 = 2+n
 for (int j=1; j<=n; j++)     //  (1+n+1+n)*n/2 = n+n^2
{
 m=m+1;                        //    n*n/2
 p=p*2;                        //     n*n/2
 L=L+3;                        //     n*n/2
 }
}
// T(n) = 5/2 n^2 + 2n+ 5
//Complexity : O(n^2)
```

التمرين السابع:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب عدد العمليات   2- احسب تعقيد الخوارزمية .

```
Void main()
{
 int m=1;              // 1
 for (int i=n; i>=1; i/=2)    // 1+log n+1+log n = 2+2log n
 for (int j=1; j<=n; j++)     //  (1+3+2)* log n=6log n
 {
 If (j==3)            // 3log n
 break;               // log n
 }
 m=m+9;           //   1
 }
// T(n) = 12 log n +4
//Complexity : O(log n)
```

التمرين الثامن:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب عدد العمليات   2- احسب تعقيد الخوارزمية .

```
function(int n)
{
if (n==1) return;                   // 1
for (int i=1; i<=n; i++)            // 1+n+1+n=2n+2
for (int j=1; j<=n; j++)            //  (1+1)n =2n
{
cout<<"*"  ;                        //  n
break;                              // n
}
}

// T(n) = 6n +3
//Complexity : O(n)
```

التمرين التاسع:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب تعقيد الخوارزمية .

```
void function(int n)
{
int count = 0;
for (int i=n/2; i<=n; i++)
 for (int j=1; j<=n; j = 2 * j)
 for (int k=1; k<=n; k = k * 2)
 count;++
}
```

//Complexity : $O(n * \log^2 n)$

التمرين العاشر:

لتكن لدينا الخوارزمية التالية والمطلوب:

1- احسب تعقيد الخوارزمية .

```
void function(int n)
{
 int count = 0;
for (int i=n/2; i<=n; i++)
for (int j=1; j+n/2<=n; j = j++)
 for (int k=1; k<=n; k = k * 2)
 count++;
}
```

Complexity : $O(n^2 * \log n)$