



جامعة المنارة
كلية الهندسة
قسم الهندسة المعلوماتية

مقرر الخوارزميات و بنى المعطيات 1

جلسة العملي الثانية

(الفصل الثاني 2023-2024)

تحليل خوارزمية البحث الثنائي والترتيب الفقاعي

تحليل خوارزمية البحث الثنائي:
تمرين 1: أوجد درجة تعقيد الخوارزمية التالية:

```
found =false;
first = 0;
last = n-1 ;
while ( (first <= last) && (!found) )
{
    mid = ( first + last ) / 2 ;
    if ( key > A[mid] )
        first= mid +1;
    else if ( key < A[mid] )
        last =mid-1;
    else
        found =1;
}
```

يمكن كتابتها على شكل شيفرة زائفة كما يلي:

1. set Found equal to false.
2. set First equal to 1.
3. set Last equal to n.
4. while First <= Last and not Found do the following :
 - a. calculate Mid = (First +Last) div 2.
 - b. if Key >A[Mid] then
 - b1. set First equal to Mid +1.
 - c. else if Key < A[Mid] then
 - c1.set last =Mid-1
 - d. else set Found equal to true.

statement	# of times executed
1	1
2	1
3	1
4	$2+\log_2 n$
a	$1+\log_2 n$
B	$1+\log_2 n$
b1	0
C	$1+\log_2 n$
c1	$1+\log_2 n$
D	0
Total:	$9+5\log_2 n$

وعليه فإن الزمن يكون :

$$T_B(n) = 9 + 5\log_2(n)$$

$$9 + 5\log_2(n) < 10 + 5\log_2(n)$$

$$9 + 5\log_2(n) < \log_2(2^{10}) + 5\log_2(n)$$

ونلاحظ أنه من أجل كل $n > 2$ فإن:

$$9 + 5\log_2(n) < \log_2(n^{10}) + \log_2(n^5)$$

$$9 + 5\log_2(n) < \log_2(n^{15})$$

$$9 + 5\log_2(n) < 15 \cdot \log_2(n)$$

وبالتالي يمكن استخدام BON للتعبير عن درجة تعقيد الخوارزمية كالتالي :

$$T_B(n) = O(\log_2(n))$$

حيث : $n_0 = 2$, $C = 15$, $f(n) = \log_2(n)$

تحليل خوارزمية الترتيب الفقاعي:
تمرين 2: أوجد درجة تعقيد الخوارزمية التالية:

```
Scan=1
while(scan<=n-1)
{
    pos=0;
    while(pos<n-1)
    {
        if(A[pos]> A[pos+1])
        {
            temp = A[pos] ;
            A[pos] = A[pos+1];
            A[pos+1]= temp ;
        }
        pos++;
    }
    scan ++ ;
}
```

1. set scan equal to 1
2. while scan less or equal to n-1 do the following:
 - a. set pos equal to 1.
 - b. while pos less or equal to n-1 do the following:
 - b1. if $A[pos] > A[pos+1]$ then do the following:
(* swap $A[pos] \leftrightarrow A[pos+1]$)
 - b11. set temp equal to $A[pos]$
 - b12. set $A[pos]$ equal to $A[pos+1]$
 - b13 set $A[pos+1]$ equal to temp
 - b2.set pos equal to pos+1
 - c. set scan equal to scan+1

Statement	# of times executed
1	1
2	N
A	N-1
B	$N \times (N - 1)$
b1	$(N - 1) \times (N - 1)$
b11	$(N - 1) + (N - 2) + \dots + 2 + 1 = \sum_{scan=1}^{scan=N-1} (N - scan) = \frac{(N) \times (N - 1)}{2}$
b12	$(N - 1) + (N - 2) + \dots + 2 + 1 = \sum_{scan=1}^{scan=N-1} (N - scan) = \frac{(N) \times (N - 1)}{2}$
b13	$(N - 1) + (N - 2) + \dots + 2 + 1 = \sum_{scan=1}^{scan=N-1} (N - scan) = \frac{(N) \times (N - 1)}{2}$
b2	$(N - 1) \times (N - 1)$
C	N-1
total:	$\frac{9}{2} N^2 - \frac{7}{2} N + 1$

بالتالي نستطيع التعبير عن زمن التنفيذ باستخدام تدوين BON كما يلي:

$$T(n) = \frac{9}{2} N^2 - \frac{7}{2} N + 1 \Rightarrow T(n) = O(n^2)$$

حيث أن :

$$\frac{9}{2} N^2 - \frac{7}{2} N + 1 \leq \frac{9}{2} N^2 \text{ for all } N \geq 1$$

أي أن $n_0 = 1$, $C = 9/2$, $f(n) = n^2$

وظيفة:
بفرض لديك

found = loc = first = 0; last = n-1, m=3,l=10 ;

أوجد درجة تعقيد النص البرمجي التالي:

```
for (int i=0;i<n;i++)
    m=m*4;
for (i=0;i<n-1;i++)
    l=l*2
while ( (first <= last) && (!found) )
{
    mid = ( first + last ) / 2 ;
    if ( key > A[mid] )
        first= mid +1;
    else if ( key < A[mid] )
        last =mid-1;
    else
        { found =1;loc=mid; }
}
```