Salman
م/ محمد فوزي
0507979205
mfhassan@yahoo.com

Big O

I

هي طريقة لتقييم الخوارزميات

ولقتصر على حساب عدد الخطوات

خطوات التنفيذ step execution

جملة الإعلان لا تحسب

int X ; لا تحسب

int Y = 20 ; تحسب

لا تحسب public void swap(int x, int y)

{ لا يحسب

} لا تحسب

write Method to Calculate

$$total = 1 + 2 + 3 + \text{----} \; n$$

$$\sum_{i=1}^{n} i \qquad 151$$

```
public int Sum (int n)
{
    int i, total;          // عداد
    total = 0;                                   [1]

    for ( i = 1 ; i <= n ; i++ )                 [n+1]   // مقارنات   عدد مرات المقارنة
        total = total + i ;                      [n]     // مرات التنفيذ

    Return total;                                [1]
}
```

وفي حالة وجود = نضع (النهاية - البداية +1)

عدد مرات المقارنة
=
النهاية - البداية + 1
في حال وجود $\boxed{=}$ لنزيد واحد

$$= 1 + \underbrace{n+1}_{\text{مقارنات}} + \underset{for}{\underline{n}} + 1$$

$$= 2n + 3$$

$$O(n)$$

ما بداخل for يتم
تنفيذه بمقدار 1
أقل من المقارنة

$$Sum = 1 + 2 + 3 + 4 .. n$$

قانون

$$\sum_{i=1}^{n} \frac{n(n+1)}{2}$$

متواليه حسابيه
151 رسمية
106 الرسمية

Public int sum (int n)

{

int total ;

total = n*(n+1)/2 ; $\boxed{1}$

return total ; $\boxed{1}$

}

يدرس for

$$= O(\overset{\text{اهم}}{2}) = O(1)$$

$O(1)$ اى رقم مهما كانت قيمته تتحول الى $O(1)$

Const.

$$O(50) = O(1)$$

$$O(1000) = O(1)$$

$$O(1000\,000) = O(1)$$

باستخدام for

مجموع الاعداد من ١ الى 20

```
int i, total;
total = 0;                          1
for (i=1; i<=20; i++)               21
{
   total = total + i;               20
}
S.o.p (total);                      1
```

$20-1+1+1=21$

$$= 1 + 21 + 20 + 1 = O(43)$$

$$= O(1)$$

$X = 0$ $\boxed{1}$

for (i = -2 ; i < n ; i ++) $\boxed{n+3}$

مقارنة $n-(-2)+1$ قانون

{

$X++$; $\boxed{n+2}$ أقل بخطوة

}

s.o.p (X) ; $\boxed{1}$

نفترض $n = \underline{3}$

i
-2
-1
0
1
$\underline{2}$
3 خراب ما لزوم n

$$\underset{n+3}{6}$$

$n-(-2)+1$

نفترض $n = 5$

i
-2
-1
0
1
2
3
4

$= \boxed{1} + \underline{n+3} + \underline{n+2} + 1$

$= 2n + 6 + \boxed{1} = O(n)$ ✓

$2n + 7 = O(n)$

```
int fun1( a[], n, m)
{
    for( i=1; i<=n; i++)            n+1
    {
        for( j=1; j<=m; j++)        m+1
        {
 n *       m* S = S + a[i][j];
        }
    }
    return s;      [1]
}
```

بتبسيط نجعل $n=m$

$n+1 + n(m+1) + n*m + 1$

$n+1 + \underbrace{nm+n}_{n(m+1)} + nm + 1$

$n=m$ للتبسيط

$n+1+n^2+n+n^2+1$

$2n^2+2n+2 = O(n^2)$

int s=0; $\boxed{1}$ الشرط-البداية+1

for (i=3 ; i<n ; i++) $n-2$

s = s+2 $n-3$

max - initial +1

$n-3+1 = n-2$

s.o.p (s) $\boxed{1}$

نفترض $n = 5$

$n=5$

i

3

4

5 خراب الشرط

$n=8$

i

3 — 1

4 — 1

5 — 1

6 — 1

7 — 1

8 خراب — 1

$6-8 = -2$

$n-2$

مقارنات for

$n-2$

ينقص منه واحد

$n-3$ تنفيذ

$\cancel{1} + n-\cancel{2} + n-3 + \cancel{1}$

$= 2n-3 = O(n)$

النهاية - البداية + 1 + 1 = عشان يساوي

```
int s=0;            1
int i=1;            1
while(i<=n)         n+1
{
    s= s+i;   1
    i++;      1     * n
}
return s            1
```

$n-\cancel{1}\cancel{+1}+1$

$$1+1+\underbrace{n+1}+\overbrace{\underbrace{n}+\underbrace{n}}^{2n}+1$$

$$=3n+4=O(n)$$

big Oh

$O(1)$

$O(\log n)$ $\qquad n = \underline{512} = 2^8$

$\qquad \log 2^8 = 8$

$O(n)$

$O(n \log n)$

$O(n^2)$

$O(n^3)$

الأسوأ $\quad O(2^n)$

for (i=1 ; i<=n ; i = i*2)

$\underline{\underline{\log n}}$

for (i=$\underset{\underline{100}}{n}$; i>=1 ; i = i/2)

$\log A^{100} \quad \log 2^7$

محمد جواد VIP

CS212 (MT1) Name ID

14

Question 1 (25 Points)

Find the total number of primitive operations and Big Oh notation of the following methods.

	Statements	S/E ضعوہ	Frequency تکرار	Total اجمالی
1	int displaySum(int n) {	~~0~~	~~0~~	~~0~~
2	int sum=0;	1	1	1
3	for(int i=1;i<=n;i=i++) { n + 1	1	n+1	n+1
4	for(int j=1;j<=12;j=j+3) { 5	1	5n	5n
5	sum=i+j; * 4	1	4n	4n
6	System.out.println("i="+i+"j="+j+"sum="+sum);	1	4n	4n
7	}	0	0	0
8	}	0	0	0
9	}	0	0	0
	Total Operations →			14n+2
	Big Oh → O(n	O(n)		

8

J 1 4 7 10 13 تکرار

	Statements	S/E	Frequency	Total
1	public static int fun1(int n) {	0	0	0
2	for(int i=1; i<=5; i++) { 6	1	6	6
3	if(n <10)			
4	System.out.println("Aslamm-o-Alaikum");	1	5	5
5	~~else~~	0	0	0
6	5* System.out.println("Walaikum-Aslaam");	0	0	0
7	}	0	0	0
8	}	0	0	0
	Total Operations →		11	11
	Big Oh → O(1)	O(1)		4

6

big

يسمح لنا بإهمال العوامل الثابتة Constant factor
و الحدود الادنى lower order و التركيز على
الجزء الرئيسي في الدالة الذي يسبب النمو

مثال

أثبت أن... $5\underline{n^4} + 3n^3 + 2n^2 + 4n + 1 = O(\overline{n^4})$

المبرر

$5n^4 + 3n^3 + 2n^2 + 4n + \underline{1} \le (5+3+2+4+1)n^4 =$

$\underline{c}\, n^4 \qquad \underline{15}\, n^4$

for $c = \underline{15}$, when $n \geq n^0 = 1$

$n_0 \geq 1$

مثال

$5n^2 + 3n \log n + 2n + 5$ is $O(\underline{n^2})$

الاثبات

$\log_2 1 = 0$

$\underline{5}n^2 + \underline{3}n \log n + \underline{2}n + \underline{5} \le (5+3+2+5)n^2 = cn$

for $c = 15$ when $n \geq n_0 = 2$

$2^0 = 1$

ملحوظة

$$\boxed{\begin{array}{c} n = 1 \text{ عند } n \log n = 0 \\ \text{صفر} \quad \log_2 1 = 0 \end{array}}$$

عكس اختيار $n_0 = 1$ تجعل المسألة ...

$20n^3 + 10n\log n + 5$ is $O(n^3)$

الإثبات

$20n^3 + 10n\log n + 5 \leq 35n^3$

for $n_0 = 1$

$20 + 0 + 5 \leq 35$ ✓

$n_0 \geq 1 \quad C = 35 \qquad \log_2 1 = 0$

$2^0 = 1$

$3\log n + 2$ is $O(\log n)$

الإثبات

$3\log n + 2 \leq 5\log n$

for $n \geq n_0 = 2$, $C = 5$

$n_0 \geq 2$

$2^n + 2$ is $O(2^n)$

الإثبات

$2^n + 2 \leq 2^{n+2}$

$\leq 2^n * 2^2$

$\leq 4\,2^n$

$$\boxed{\begin{array}{l} 2^n+2 \leq 2^{n+1} \\ \quad \leq 2\,2^n \\ c = 2 \\ n_0 \geq 1 \end{array}}$$ ✓

$C = 4 \qquad n_0 = 1$

$$f(n) = n^2 + n + 5 \leq (1+1+5)n^2$$
$$\leq 7n^2$$

$\therefore O(n^2)$ $\quad c = 7$, $n_0 = 1$

$n^5 + n^3 + 2n + 1$ $\quad O(n^5)$

$n^5 + n^3 + 2n + 1 \leq 5n^5$

$c = 5 \quad n_0 \geq 1$

$\log(n^5+1) + (\log(n))^4$ $\quad O(n^4)$

$$\log(n^5+1) \leq \log(2n^5)$$
$$\leq \log 2 + \log n^5$$
$$\leq \log n + 5\log n \leq \underline{6} \log n$$

$\therefore O(\log n)$ $\quad c = 6$, $n_0 \geq 1$

$(\log n)^4 \leq n^4$ $\quad \therefore O(n^4)$

$c = 1$
$n_0 \geq 1$

$\therefore$

$6\log n + n^4 = \boxed{O(n^4) \quad c = 1 \quad n_0 \geq 1}$

$$\frac{n^4 + n^2 + 1}{n^4 + 1} \leq \frac{n^4 + n^2 + 1}{n^4} , n_0 \geq 1$$

$$\leq \frac{3 \cancel{n^4}}{\cancel{n^4}}$$

$$\leq 3$$

$O(1)$ $c = 3$ $n_0 \geq 1$

$n^2 - n$ $O(n^2)$

$n^2 - n \leq n^2$ for $c = 1$

$n_0 \geq 0$

الساالب لحل

ترتيب الدوال حسب النمو

1 $\log n$ n $n\log n$ n^2 n^3 2^n

$$\underbrace{(n^2+2)}_{\text{حلها}} \; \underbrace{\log(n^5+1)}_{\text{لحلها}}$$

$$n^2 + 2 \leq 3n^2$$

$c = 3$
$n_0 \geq 1$

$$O(n^2)$$

$$\log(n^5+1) \leq \log(2n^5)$$

$$\leq \log 2 + \log n^5$$

$$\leq \log 2 + 5\log n$$

$$\leq \log n + 5\log n$$

$$\leq 6\log n$$

$c = 6$ $\quad n_0 \geq 0$

$\cancel{6}$

$$(n^2+2)\log(n^5+1) \leq (n^2+2)\cdot 6\log(n$$

$$\leq 6n^2\log n + 12\log n$$

$$\leq 18n^2\log n$$

$c = 18$

$n_0 \geq 0$

$$n^5 + n^2 - 5 + \frac{n^2}{n+1}$$

$$n^5 + n^2 - 5 \leq \underline{2 \cdot n^5}$$

$$O(n^5) \qquad c = 2 \ , \ n_0 \geq 0$$

$$\frac{n^2}{n+1} \leq \frac{\cancel{n^2}^{\,n}}{\cancel{n}}$$

$$\leq n$$

$$O(n) \qquad c = 1 \qquad n_0 \geq 0$$

$$O(n^5) \qquad n_0 \geq 0$$