Enroll no. - 201B172

Jaypee University of Engineering and Technology

18B11CI311 – Data Structures
B.Tech -3rd Semester
Tutorial – 1 (Time Complexity)

1.Find the exact step counts (growth function) and time complexity for following algorithms -

i.
```
1.Input m and n
2.If m == n
3.      Print "Same"
4.      Sum=m+n
5.      Print Sum
6.Else  Print "Not same"
```

ii.
```
1. Initialize i =0
2. For i=0 to n-1
3.      Read an integer in Arr[i]
4. For i=0 to n-1
5.      Print Arr[i]
```

iii.
```
Algorithm Add(a,b,c,m,n)
1. for i :=1to m do
2.      for i :=1to n do
3.           c[i,j]:=a[i, j]+b[i,j];
```

iv.
```
1.int i, j, k = 0;
2.for (i  =1; i <= n*n; i++)
3.      for (j = 1; j <= n; j = j +1)
4.           k = k + n/2;
```

v.
```
Algorithm Sum (a,n)
1 s:=0.0;
2       for i :=1to n do
3               s :=s+a[i];
4 return s;
```

vi.
```
1. sum = 0
2. for i = 1 to n:
3.      for j = i to n:
4.           sum += a[i][j]
5. print(sum)
```

vii.
```
1. Set i =0
2. Repeat step 3-6 till i<=n
3.      Set J=n
4.      While J>0
5.           J = J/2
6.      i = i+1
```

viii.
```
1. Set I =n
2. While  I > 1
3.      J =1
4.      While J < n
5.           K=1
6.           While K < n
7.                K += 2
8.           J *= 2
9.      I /= 2
```

ix.
```
  void fun(int n, int arr[])
1. {
2.    int i = 0, j = 0;
3.    for(; i < n; ++i)
4.      while(j < n && arr[i] < arr[j])
5.         j++;
6. }
```

2. Show that following statements are correct:

i. $4n+100=O(n)$
ii. $500n^3+6n+6=O(n^3)$
iii. $n^3 \neq O(n^2)$
iv. $5n^2-6n= O(n^2)$
v. $n! = O(n^n)$
vi. $2n^2 2^n + n\log n = O(n^2 2^n)$
vii. $3n^3 +4n^2 = \Omega(n^2)$
viii. $=\theta(n^3)$

3. Compare the two functions n^2 and $2^n/4$ for various values of n. Determine when the second becomes larger than the first.

4. Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?

f1(n) = 2^n, f2(n) = n^(3/2), f3(n) = nLogn, f4(n) = n^(Logn)

Enroll no. - 2018172

Date 03/09/2021
Page 01

1.

(i.) Here we have if else conditional statement.
so depending upon the value of m and n, either
if condition or else condition will be executed.
So the exact step counts is 5 or 3
The time complexity is "O(1)".

(ii) Here we have two for loop, each of which run
for n times so that exact number of steps is 2n+1
The time complexity is "O(n)".

(iii) Here two for loops, first will execute 'm' times and
second will run for 'n' times.
So total steps is m*n
The time complexity is "O(m*n)"

(iv) Here we have nested for-loops and every
iteration takes 'n' times.
So the total number steps is n*n*n.
The time complexity is "O(n3)"

(v) Here the loop starts from i=1 and goes to 'n+1' times
Thus, the exact number of steps involves = 1+(n+1)+n = 2n+2
The time complexity is "O(n)".

(vi) Here the first for loop starts from i=1 and goes to 'n+1' times
Similarly loop no. second starts with j=1 and runs 'n+1' times
Thus each iteration of the first loop second loop iterates
(n-i+1) times.

Date___/___/_____
Page_________

Thus, the exact number of steps $= 1 + (n+1) + n^2 + n + n^2$
$= 2n^2 + 2n + 2$

The time complexity is "$O(n^2)$".

(vii) Here our first iteration work 'n' times and second work 'Log n' times
The time complexity for the above code is "$O(n\log(n))$"

(viii) Here our loop variables i and j we divided and multiplied respectively thus have $O(\log(n))$ complexity and remaining condition will work 'n' no. of time.
Thus the time complexity: "$O(\log(n) * \log(n) * n)$".

(ix) Here the two nested loops. First for loop will execute from 0 to n and second while loop will also execute n times.
The time complexity is "$O(n^2)$".

Date / /
Page

2.

(i) $4n + 100 = O(n)$

$f(n) = O(g(n))$

$f(n) \leq c(g(n))$

here c is constant

Put $c = 5$

$4n + 100 \leq 5n$

Hence it is true.

(ii.) $500n^3 + 6n + 6 = O(n^3)$

Taking highest degree

$500n^3 = O(n^3)$

$500n^3 \leq cn^3$

let $c = 600$

$500n^3 \leq 600n^3$

Hence it is true

(iii) $n^3 \neq O(n^2)$

$f(n) = O(g(n))$

$f(n) \leq c(g(n))$

here c is constant

Now c = putting any value want satisfy.

$n^3 \leq cn^2$

Thus $n^3 \neq O(n^2)$.

Date ___/___/___
Page ___

(iv.) $5n^2 - 6n = O(n^2)$

Taking higher degree.

$5n^2 = O(n^2)$

$5n^2 \leq c(n^2)$

let $c = 6$

$5n^2 \leq 6(n^2)$

Hence it is true.

v) $n! = O(n^n)$

Here $n! = n(n-1)(n-2)$ & $n^2 = n(n).(n)$

Both multiplications are n times.

$f(n) = O(g(n))$

Here f(n) have upper bound x^2 and f(x) is constant

n or n^2

$f(n) \leq c(n^2)$

$n! \leq cn^n$. $\forall$ c is constant.

(vi) $2n^2 2^n + n\log n = O(n^2 2^n)$

$f(n) = O(g(n))$.

$f(n) \leq c g(n)$ where c is constant

Here $2n^2 2^n > n\log n$

$\therefore 2n^2 2^n = 2^{n+1} n^2 \Rightarrow n^2 2^n$

Now, let $c = 2$.

$n^2 2^n \leq 2n^2 2^n$

It is true.

Date___/___/___
Page______

(vii) $3n^3 + 4n^2 = \Omega(n^2)$

It is in form $f(n) = \Omega g(n)$

If $f(n) = \Omega(g(n))$ then there must exists constants c and n, such that $f(n) \geq c g(n) \ \forall \ n \geq n_0$

i.e $3n^3 + 4n^2 \geq c \cdot n^2 \ \forall \ n >= n_0$.

Now dividing by n^2.

$3n + 4 \geq c \ \forall \ n >= n_0$

If we choose $n_0 = 1$ the we need a value of c such that

$3 + 4 \geq c$

We can set $c = 7$, we have

$3n^3 + 4n^2 \geq 7n^2 \ \forall \ n >= 1$.

$\therefore 3n^3 + 4n^2 = \Omega(n^2)$

Date __/__/__
Page __

3. function 1 : n^2
function 2 : $2^n/4$

n	function 1: (n^2)	function 2: ($2^n/4$)
1	1	1/2
2	4	1
3	9	2
4	16	4
5	25	8
6	36	16
7	49	32
8	64	64
9	81	128

when $n=9$ 2nd becomes larger then 1st.

4. Here $f3(n) = n\log n$ is linearlogirithmic, $f2(n) = n^{3/2}$ and $f4(n) = n^{\log n}$ are polynomial and $f1(n) = 2^n$ is exponential function.

We know that increasing order of asymptomatic complexity of linearlogirithmic, polynomial and exponential function is follows these functions

Now for both polynomial $f_2(n) = n^{3/2}$ and $f_4(n) = n^{\log n}$. base is same i.e n.

$\therefore \quad 3/2 < \log n$

$n^{3/2} < n^{\log n}$.

Date / /
Page

The required increasing order asymptomatic complexity of function is.

$f_3(n) = n\log n < f_2(n) = n^{3/2} < f_4(n) = n^{\log n} < f_1(n) = 2^n$

So, $f_3 < f_2 < f_4 < f_1$