Name :- Abhay Rawat

Course :- B. Tech (CSE) Sec :- B

Sem :- 5 Roll No :- 01

University Roll No :- 1961002

Design & Analysis of Algorithms

Tutorial Sheet 1

Ques What is the complexity of the following piece of code:

1 What is the time, space complexity of follow -ing code :-

```
int a=0, b=0;
for(i=0; i<N; i++) — N
{  a = a + rand()
 for(j=0; j<M; j++) — M
 {
    b = b + rand();
 }
```

Ans O(N+M) time O(1) space

2.
```
int sum=0; i;
for(i=0; i<n; i=i+2)
  {
   sum+=i;
  }
```

Ans $i = 0, 2, 4, 6, 8, \ldots\ldots n$

k step

$T(n) = O(k)$

$a = 0$, $d = 2$

k th term

$$a_k = a + (k-1)\,d$$
$$a_k = 0 + (k-1)\,2$$
$$n = (k-1)\,2$$
$$\boxed{\frac{n}{2} + 1 = k}$$

$$T(n) = O\left(\frac{n}{2} + 1\right) = O(n)$$

$$\boxed{T(n) = O(n)}$$

3. int sum=0, i;

```
for (i=1; i<n; i=i*2)
{
  sum += i;
}
```

Ans $\underbrace{i = 1, 2, 4, 8, 16, \ldots\ldots, n}_{k \text{ iteration}}$

$a = 1$ $\qquad r = 2$ $\qquad k^{th}$ term $= n$

$$G_k = a r^{n-1}$$
$$n = 1\,(2)^{k-1}$$
$$\log_2 n = (k-1)\log_2 2$$
$$\boxed{k = \log n + 1}$$

$$T(n) = O(\log n + 1) = O(\log n)$$

$$\boxed{T(n) = O(\log n)}$$

Scanned by TapScanner

4.
```
int sum=0, i;
for(i=0; i*i<n; i++)
{
    sum+=i;
}
```
Ans $O(\sqrt{n})$

5.
```
int j=1, i=0;
while (i<=n)
{
    i=i+j;
    j++;
}
```
Ans

No. of step(k)	i	j
0	0	1
1	1	2
2	3	3
3	6	4
4	10	5
5	15	6
6	21	7
⋮	⋮	⋮
k steps	n	⋮

$T(n) = O(k)$

$i = 0, 1, 3, 6, 10, \ldots\ldots n$

~~$S_n = i = 1, 2, 3, 4, \ldots$~~

~~$S_n =$~~

$$S_n = 1 + 3 + 6 + 10 + 15 + \dots\dots + n$$
$$S_n = \quad 1 + 3 + 6 + 10 + \dots\dots + (n-1) + n$$
$$(-) \quad (-) \quad (-) \quad (-) \quad - \; - \; - \; - \; - \; -$$

$$0 = 1 + 2 + 3 + 4 + 5 + \dots\dots \cancel{+n} - n$$
$$n = 1 + 2 + 3 + 4 + 5 + \dots\dots \text{ k th step}$$
$$n = \frac{k}{2}(2(1) + (k-1)1)$$
$$2n = k(2 + k - 1) \Rightarrow 2n = k(k+1)$$
$$2n = k^2 + k$$
$$2n = (k + 1/2)^2 - (1/2)^2$$
$$(k + 1/2)^2 = 2n + (1/2)^2$$
$$k + 1/2 = \sqrt{2n + (1/2)^2}$$
$$\boxed{k = \sqrt{2n + (1/2)^2} - 1/2}$$

$$T(n) = O(k)$$
$$T(n) = O(\sqrt{2n + (1/2)^2} - 1/2)$$
$$\boxed{T(n) = O(\sqrt{n})}$$

6. void recursion (int n)
```
{
   if (n==1) return;
   recursion (n-1);  — T(n-1)
   print (n);
   recursion (n-1);  — T(n-1)
}
```

Ans $T_0 = 1$

$T_n = T(n-1) + T(n-1) + k$

$$T(n) = 2T(n-1) + c \quad \text{if } n > 1$$
$$T(1) = 0 \quad \text{if } n = 1$$

$$T(n) = 2T(n-1) + C$$
$$T(n-1) = 2T(n-2) + c$$

$$T(n) = 2(2T(n-2) + c) + c$$
$$T(n) = 2^2\, T(n-2) + 2c + c$$
$$T(n-2) = 2T(n-3) + c$$

$$T(n) = 2^2(2T(n-3) + c) + 2c + c$$
$$T(n) = 2^3 T(n-3) + 2^2 c + 2c + C$$
$$\vdots$$

<u>General Equation</u>

$$T(n) = 2^i\, T(n-i) + (2^0 + 2^1 + 2^2 + \dots + 2^{i-1})c$$
$$T(n-i) = T(0)$$
$$n - i = \phi \Rightarrow \boxed{n-1 = i}$$

$$T(n) = 2^{n-1}\, T(n-(n-1)) + (2^0 + 2^1 + 2^2 + \dots + 2^{n-2})c$$
$$T(n) = 2^{n-1} + 2^0 + 2^1 + 2^2 + \dots + 2^{n-1}$$
$$T(n) = \frac{2^0(2^n - 1)}{2-1} = 2^n$$
$$\boxed{T(n) = O(2^n)}$$

7. int recursion (int what[], int thisone, int thatone, int x)

```
{
 if ( thatone >= thisone)
 {
   int something = thisone + (thatone - thisone)
        /2;
  if ( what[something] == x)
       return something;
  else if (what[something] > x)
      return recursion (what, thisone,
      something-1, x);
  return recursion( what, something+1,
     thatone, x);
  }
  return -1;
}
```

Ans

$$T(n) = T(n/2) + 1$$

$$T(1) = 1$$

$$T(n) = \begin{cases} T(n/2) + 1 & n > 1 \\ 1 & n = 1 \end{cases}$$

$$T(n) = T(n/2) + 1 .$$

$$T(n) = T(n/2) + 1$$

$$T(n/2) = T(n/2^2) + 1)$$

$$T(n) = T(n/2^2) + 2$$

$$T(n/2^2) = T(n/2^3) + 1$$

$T(n) = T(n/2^3) + 3$

$\vdots$

General form

$T(n) = T(n/2^i) + i$

$$\frac{n}{2^i} = 1 \Rightarrow n = 2^i \; \& \; \boxed{i = \log n}$$

$T(n) = T(1) + \log n$

$T(n) = 1 + \log n$

$\boxed{T(n) = O(\log n)}$

Ques 8 Solve the following recurrence relation T
$T(1) = 1$.

~~Ans~~

(I) $T(n) = T(n-1) + 1$

Ans

$T(n) = T(n-1) + 1$

$T(n-1) = T(n-2) + 1$

$T(n) = (T(n-2) + 1) + 1$

$T(n) = T(n-2) + 2$

$T(n-2) = T(n-3) + 1$

$T(n) = (T(n-3) + 1) + 2$

$T(n) = T(n-3) + 3$

$\vdots$

General Term

$T(n) = T(n-k) + k$ — (i)

$T(n-k) = T(1)$

$$n - k = 1$$
$$n = 1 + k \Rightarrow \boxed{n - 1 = k}$$

Put the value of k in Eq (i)

$$T(n) = T(n-(n-1)) + n - 1$$
$$T(n) = T(1) + n - 1$$
$$T(n) = \cancel{1} + n \cancel{-1}$$
$$T(n) = n$$
$$\boxed{T(n) = O(n)}$$

2. $T(n) = T(n-1) + n.$

Ans

$$T(n) = T(n-1) + n$$
$$T(n-1) = T(n-2) + (n-1)$$
$$T(n) = T(n-2) + (n-1) + n$$
$$T(n) = T(n-2) + 2n - 1$$
$$T(n-2) = T(n-3) + (n-2)$$
$$T(n) = T(n-3) + (n-2) + 2n - 1$$
$$T(n) = T(n-3) + 3n - 3$$

⋮

General Term.

$$T(n) = T(n-k) + \cancel{3k}\, kn - k \quad \text{— (i)}$$
$$T(n-k) = T(1)$$
$$n - k = 1$$
$$\boxed{k = n - 1}$$
$$T(n) = T(n-(n-1)) + (n-1)n - (n-1)$$
$$T(n) = T(1) + n^2 - n - n + 1.$$
$$T(n) = 1 + n^2 - 2n + 1$$

$$T(n) = n^2 - 2n - 2$$

$$\boxed{T(n) = O(n^2)}$$

3. $T(n) = T(n/2) + 1$

Ans $T(n) = T(n/2) + 1$

$T(n/2) = T(n/2^2) + 1$

$T(n) = T(n/2^2) + 1 + 1$

$T(n) = T(n/2^2) + 2$

$T(n/2^2) = T(n/2^3) + 1$

$T(n) = T(n/2^3) + 1 + 2$

$T(n) = T(n/2^3) + 3$. — (i)

⋮

General Term

$T(n) = T(n/2^k) + k$

$T(n/2^k) = T(1)$

$\frac{n}{2^k} = 1 \Rightarrow \boxed{n = 2^k}$ or $\boxed{\log n = k}$

Put the value of k in Eq (i)

$T(n) = T(1) + \log n$

$$\boxed{T(n) = O(\log n)}$$

4. $T(n) = 2T(n/2) + 1$

Ans $T(n) = 2T(n/2) + 1$

$T(n/2) = 2T(n/2^2) + 1$

$T(n) = 2(2T(n/2^2) + 1) + 1$

$T(n) = 2^2 T(n/2^2) + 2 + 1$

$T(n/2^2) = 2T(n/2^3) + 1$

$T(n) = 2^2(2T(n/2^3)+1) + 2 + 1$

$T(n) = 2^3\, T(n/2^3) + 2^2 + 2^1 + 2^0$

$\vdots$

General Term

$T(n) = 2^k\, T(n/2^k) + (2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^{k-1})$ — (i)

$T(n) = 2^k\, T(n/2^k) + 2^0\left(\frac{2^{k} - 1}{2 - 1}\right)$

$T(n) = 2^k\, T(n/2^k) + 2^k$ — (i)

$T(n/2^k) = T(1)$

$\frac{n}{2^k} = 1 \Rightarrow \boxed{n = 2^k}$

$T(n) = n\,T(1) + n \Rightarrow 2n$

$T(n) = 2n$

$\boxed{T(n) = O(n)}$

5. $T(n) = 2T(n-1) + 1$

Ans $T(n) = 2T(n-1) + 1$

$T(n-1) = 2T(n-2) + 1$

$T(n) = 2(2T(n-2)+1) + 1$

$T(n) = 2^2\, T(n-2) + 2 + 1$

$T(n-2) = 2T(n-3) + 1$

$T(n) = 2^2(2T(n-3) + 1) + 2 + 1$

$T(n) = 2^3\, T(n-3) + 2^2 + 2^1 + 2^0$

$\vdots$

General Term

$$T(n) = 2^k T(n-k) + (2^0 + 2^1 + 2^2 + \cdots + 2^{k-1})$$

$$T(n) = 2^k T(n-k) + \cancel{2^{k-1}}\, 2^0 \left(\frac{2^{k} - 1}{2 - 1} \right)$$

$$T(n) = 2^k T(n-k) + 2^k \quad \text{—— (i)}$$

$$T(n-k) = T(1)$$

$$n-k = 1 \Rightarrow \boxed{n-1 = k}$$

$$T(n) = 2^{n-1} T(n-(n-1)) + 2^{n-1}$$

$$T(n) = 2^{n-1} + 2^{n-1} = 2^n$$

$$\boxed{T(n) = O(2^n)}$$

6. $T(n) = 3T(n-1)$, $T(0) = 1$.

A. $T(n) = 3T(n-1)$

$$T(n-1) = 3T(n-2)$$

$$T(n) = 3 \times 3T(n-2)$$

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

$$T(n) = 3^3 T(n-3)$$

⋮

General Term

$$T(n) = 3^k T(n-k) \quad \text{—— (i)}$$

$$n-k = 0$$

$$\boxed{n = k}$$

Put in E (i)

$$T(n) = 3^n T(n-n) = 3^n (1)$$

$$\boxed{T(n) = O(3^n)}$$

Scanned by TapScanner

7. $T(n) = T(\sqrt{n}) + 1$

Ans $T(n) = T(n^{1/2}) + 1$

$\uparrow$ —— $T(n^{1/2}) = T(n^{1/2^2}) + 1$

$T(n) = T(n^{1/2^2}) + 1 + 1$

$T(n) = T(n^{1/2^2}) + 2$

$\uparrow$ —— $T(n^{1/2^2}) = T(n^{1/2^3}) + 1$

$T(n) = T(n^{1/2^3}) + 1 + 2$

~~$T(n) = T(n^{1/2^k}) + k$~~

⋮

General equation

$T(n) = T(n^{1/2^k}) + k$ —— (i)

Assume $n^{1/2^k} = 2$

$\frac{1}{2^k} \log_2 n = 1$

$\log_2 n = 2^k$

$\boxed{\log_2(\log_2 n) = k}$

$T(n) = 1 + \log_2(\log_2 n)$

$\boxed{T(n) = O(\log(\log n))}$

8. $T(n) = T(n^{1/2}) + n$

Ans $T(n) = T(n^{1/2}) + n$

$\uparrow$ —— $T(n^{1/2}) = T(n^{1/4}) + n$

$T(n) = T(n^{1/2^2}) + n + n$

$\uparrow$ —— $T(n^{1/2^2}) = T(n^{1/2^3}) + n$

Scanned by TapScanner

$$T(n) = T(n^{1/2^3}) + 3n$$

$\vdots$

General Term

$$T(n) = T(n^{1/2^k}) + kn \quad \text{— (i)}$$

$$n^{1/2^k} = 2$$

$$\frac{1}{2^k} \log n = 1$$

$$\log n = 2^k \Rightarrow \boxed{\log(\log n) = k}$$

Put in Eq (i)

$$T(n) = T(1) + n \log(\log n)$$

$$\boxed{T(n) = O(n \log(\log n))}$$

Ques 9

```
int sum=0, i;
for(i=0; i<n; i++)  — n
{
    sum += i;
}
```

Ans $O(n)$

Ques 10 What is the time complexity of following code.

```
int a=0;
for(i=0; i<N; i++)
{
    for(j=N; j>i; j--)
    {
        a=a+i+j;
```

}

}

Ans for i it will n times

for ~~for~~ j it will linear so than also it will n times

Hence $\boxed{O(n^2) = T(n)}$

Ques 11 What is the time complexity of following code.

```
int i, j, k=0;
for(i=n/2; i<=n; i++)  — n
{
  for (j=2; j<=n; j=j*2) — log n
     k = k * n/2;
}
```

Ans $O(n \log n)$

Ques 12 What does it mean when we say that an algorithm X is asymptotically more efficent than Y?

Ans X will be a better choice for all input except possibly small inputs.

Ques 13 What is the time complexity of following code.

```
int a=0, i=N;
while(i>0)
{
```

Scanned by TapScanner

a+=i;
i/=2;
}

Ans $O(\log n)$

Ques 14 Solve the following recurrence relation?

$T(n) = 7T(n/2) + 3n^2 + 2$.

Ans $a = 7$ $b = 2$ $f(n) = 3n^2 + 2$

$\boxed{C = 2}$

It falls in master's theorem case 1:

$\log_b(a) = \log_2 7 = 2.81 > 2$

$O(n^{2.81})$

Ques 15 Sort the following function in the decreasing order of the asymptotic (big-O) complexity.

$f1(n) = n^{\sqrt{n}}$ $f2(n) = 2^n$ $f3(n) = (1.000001)^n$

$f4(n) = n^{10} * 2^{n/2}$

Ans $f2 > f4 > f3 > f1$

Ques 16 $f(n) = 2^{2n}$

What is the following correctly represents the above function.

Ans $\Omega(2^n)$

Ques 17 $T(n) = 2T(n/2) + n/2$. $T(n)$ will be

Ans $O(n \log n)$

Ques 18

```
int gcd (int n, int m)
{
  if (n%m==0) return m;
  if (n<m) swap (n, m);
  while (m>0)
  {
    n=n%m;
    swap (n,m);
  }
  return n;
}
```

Ans $O(\log n)$

Ques 19

```
int a=0, b=0;
for(i=0; i<N; i++)   — n
{
  for(j=0; j<N; j++)   — n
     a=a+j;
  for(k=0; k<N; k++)   — n
     b=b+k;
}
```

Ans $n(n+n)$

$= n(2n)$

$= 2n^2$

$\boxed{= O(n^2)}$