Todays Activity

Interesting TC based questions

Match following Complexities

1. ~~Linear~~	A : $N^{k+6} \longrightarrow O(N^{k+6})$	(4)
2. ~~Logarithmic~~	B : $5^{N*2} \longrightarrow (5^2)^N = 25^N$	(3)
3. ~~Exponential~~	C : $N/4 \log_2^N \rightarrow O(N\log_2^N)$	(5)
4. ~~Polynomial~~	D : $3^{20}N + 10^5 \rightarrow O(N)$	(1)
5. ~~Linear Logarithmic~~	E : $10N + 9N/100 + 340N^2 \rightarrow O(N^2)$	(6)
6. ~~Quadratic~~	F : $10^3 \log_2^{4N} \longrightarrow O(\log_2^N)$	(2)

$$\left[\log_2^a + \log_2^b = \log_2^{ab}\right]$$

$$\log_2^{4N} = \left\{\log_2^4 + \log_2^N = 2 + \log_2^N\right.$$

Note:

exponential: Number raised to power N.

Tricky Questions.

```
void fun(int N){
    int c=0;
    for(int i=N/2; i<=N; i++){
        int j=N;
        while(j>1){
            j=j/2;
        }
    }
}
```

i	j	
$N/2$	$j:[N\ N/2\ N/4\ \ldots]\ 1$	$\log_2 N + 1$
$N/2+1$	$j:[N\ N/2\ N/4\ \ldots]\ 1$	$\log_2 N + 1$
$N/2+2$	$j:[N\ N/2\ N/4\ \ldots]\ 1$	$\log_2 N + 1$
$\vdots$		$\vdots$
N	$j:[N\ N/2\ N/4\ \ldots]\ 1$	$\log_2 N + 1$
$N+1$		

Outer loop $= [N/2 \ldots N] = N - N/2 + 1 \Rightarrow N/2 + 1$

Inner loop $= [N/2 + 1] \log_2 N$

Total iterations:

$= N/2 + 1 + [N/2 + 1] \log_2 N$

$= N/2 + 1 + \frac{N}{2} \log_2 N + \log_2 N$

$= \boxed{O(N \log_2 N)}$

```
void fun(int N){
    int c=0;
    for(int i=N; i>0; i/=2){
        for(int j=0; j<i; j++){
            c+=1;
        }
    }
}
```

Table

i	j = [0..i-1]
N	j: [0..N-1] = N +
N/2	j: [0..N/2-1] = N/2 +
N/4	j: [0..N/4-1] = N/4 +
N/8	j: [0..N/8-1] = N/8 +
⋮	
1	j: [0..0] = 1

Outerloop = $\log_2^N$

Inner loop = $N + N/2 + N/4 + N/8 + \dots 1 = 2N-1$

$$S = N + N/2 + N/2^2 + N/2^3 + N/2^4 + \dots$$

$$2S = 2N + \cancel{N} + \cancel{N/2} + \cancel{N/2^2} + \cancel{N/2^3} + \cancel{N/2^4} + \dots \cancel{2}$$

$$S = \cancel{N} + \cancel{N/2} + \cancel{N/2^2} + \cancel{N/2^3} + \cancel{N/2^4} + \dots \cancel{2} + 1$$

$$S = 2N-1$$

Total = $\log_2^N + 2N-1 \Rightarrow O(N)$

```
void fun(int N){
    for(int i=0; i<N; i++){
        for(int j=N; j>i; j--){          ≡   j: i+1 > i :
        }                                    j: i > i : stop
    }
}
```

i	j:[N... i+1]
0	j:[N,.. 1] = N
1	j:[N...2] = N-1
2	j:[N...3] = N-2
⋮	
N-1	j:[N,,N] = 1
N: stop	

a b b-a+1

Outer loop : i = [0,,N-1] = N-1-0+1 = N

Inner loop : j = $\frac{(N)(N+1)}{2}$

Total iterations = $N + \frac{N^2+N}{2} = \frac{N^2}{2} + \frac{3N}{2} = O(N^2)$

Q3

```
int fun(int N, int k){
    int ans=0;
    for(int i=1; i<=N; i++){
        int p= pow(i,k) : i^k
        for( int j=1; j<=p or i^k; j++){
            ans = ans+p;
        }
    }
}
```

Table

i	j : [1.. i^k]
1	[1.. 1^k] = 1^k +
2	[1.. 2^k] = 2^k +
3	[1.. 3^k] = 3^k
⋮	
N	[1.. N^k] = N^k

Outer loop = N

Inner loop = $\underline{1^k + 2^k + 3^k + \dots N^k}$

Higher Order Term

if k==1 : $S = 1^1 + 2^1 + 3^1 + \dots N^1 = \frac{(N)(N+1)}{2}$ $= \left[\frac{N^2}{2}\right]$

if k==2 : $S = 1^2 + 2^2 + 3^2 + \dots N^2 = \frac{(N)(N+1)(2N+1)}{6} = \frac{2N^3}{6} = \left[\frac{N^3}{3}\right]$

if k==3 : $S = 1^3 + 2^3 + 3^3 + \dots N^3 = \left[\frac{(N)(N+1)}{2}\right]^2 = \left[\frac{N^2+N}{2}\right]^2 = \left[\frac{N^4}{4}\right]$

if Input k $S = 1^k + 2^k + 3^k + \dots N^k \longrightarrow \frac{N^{k+1}}{k+1}$

Final = $O\left(\frac{N^{k+1}}{k+1}\right)$. We cannot neglect $\frac{1}{k+1}$ because it's not a constant

Q4

```
void fun(int ar[], int N){
    int j=0;
    for(int i=0; i<N; i++){
        while((j<N) && (ar[i] <= ar[j])){
            j++;
        }
    }
}
```

Ex: ar[N]; ($a_0 \quad a_1 \quad a_2 \ldots a_{b_1} \quad a_4 \quad a_5 \ldots\ldots a_{N-1}$

i=0 , j

i	j	iterations
0	$[0 \ldots b_1]$	$b_1 - 0 + 1 = b_1 + 1 = \cancel{b_1} + 1$
1	$[b_1+1 \ldots b_2]$	$b_2 - (b_1+1) + 1 = b_2 - b_1 - \cancel{1} + \cancel{1} = \cancel{b_2} - \cancel{b_1}$ +
2	$[b_2+1 \ldots b_3]$	$b_3 - (b_2+1) + 1 = b_3 - b_2 - \cancel{1} + \cancel{1} = \cancel{b_3} - \cancel{b_2}$ +
3	$[b_3+1 \ldots b_4]$	$b_4 - (b_3+1) + 1 = b_4 - b_3 - 1 + 1 = \cancel{b_4} - \cancel{b_3}$ +
⋮		
N-1	$[b_{N-1}+1 \ldots N-1]$	$= N-1 - (b_{N-1}+1) + 1 = N-1 - b_{N-1} - \cancel{1} + \cancel{1} = N - \cancel{b_{N-1}} - 1$

Total Outer loop = N

Total Inner loop = $N - \cancel{1} + \cancel{1} = N$

Total iterations = $N + N = 2N = O(N)$

i=0 ⟶ : N

Ex: ar[N]; $a_0 \quad a_1 \quad a_2 \ldots a_{b_1} \quad a_4 \quad a_5 \ldots\ldots a_{N-1}$

j=0 ⟶ : N

$= 2N = \underline{O(N)}$

```
void fun(int N){
    for(int i=1; i<=2^N; i++){
        for(int j=1; j<=i; j++){
            print("Hello")
        }
    }
}
```

Table

i	j : [1...i]
1	j : [1..1] = 1 +
2	j : [1.. 2] = 2 +
3	j : [1..3] = 3 +
2^N	j : [1..2N] = 2^N +

Outer loop = 2^N

Inner loop = $\underline{1 + 2 + 3 + \dots\ 2^N}$

Sum of first k natural numbers = $\frac{(k)(k+1)}{2}$

$k = 2^N \longrightarrow \frac{[2^N][2^N+1]}{2}$

$$\frac{[2^N][2^N+1]}{2} = \frac{2^N * 2^N + 2^N}{2} = \frac{2^{2N} + 2^N}{2}$$

Total iteration = $\frac{2^{2N} + 2^N}{2} + 2^N = \frac{2^{2N} + 3*2^N}{2} = \frac{2^{2N}}{\cancel{2}} + \frac{3*2^N}{2}$

Final BigO = $O(2^{2N}) = O((2^2)^N) = O(4^N)$

```
void fun(int N){
    for(int i=1; i<=N; i++){
        for(int j=1; j<=2^i; j++){
            print("Hello");
        }
    }
}
```

i	j : [1..2^i]
1	$j : [1..2^1] = 2^1$ +
2	$j : [1..2^2] = 2^2$ +
3	$j : [1..2^3] = 2^3$ +
⋮	
N	$j : [1..2^N] = 2^N$ +

Outer loop: N

Inner loop: $2^1 + 2^2 + 2^3 + \dots 2^N$

$$S = 2^1 + 2^2 + 2^3 + 2^4 + \dots 2^{N-1} + 2^N$$

$$2S = \cancel{2^2} + \cancel{2^3} + \cancel{2^4} + \cancel{2^5} + \dots \cancel{2^N} + 2^{N+1}$$

$$S = 2^1 + \cancel{2^2} + \cancel{2^3} + \cancel{2^4} + \dots \cancel{2^{N-1}} + \cancel{2^N}$$

$$S = 2^{N+1} - 2$$

$$S = 2^1 * 2^N - 2 \Rightarrow 2 * 2^N - 2$$

$$\text{Total} = N + \boxed{2 * 2^N} - 2; \quad = O(2^N)$$