

①

Q1. Solve the following loops & Calculate Time Complexity:

① Void Function (int n) {
 int i, j, k, Count = 0;
 for (i = $n/2$; i \leq n; i++)
 for (j = 1; j + $n/2$ \leq n; j++)
 for (k = 1; k \leq n; k* = 2)
 Count++
}

② Void Function (int n) {
 if (n == 1) return;
 for (int i = 1; i \leq n; i++) {
 for (int j = 1; j \leq n; j++) {
 Printf(" *");
 break;
 }
 }
}

③ Void Function (int n) {
 int i, Count = 0;
 for (i = 1; i * i \leq n; i++)
 Count++;
}

④ function (int n)

```
{  
    for (int i = 1; i <= n; i++)  
        for (int j = 1; j <= n; j *= 2)  
            printf("%*");  
}
```

y

⑤ function (int n) {

```
    for (int i = 1; i <= n/3; i++)  
        for (int j = 1; j <= n; j += 4)  
            printf("%*");  
}
```

y

⑥ function (int n)

```
{    int sum = 0;
```

```
    for (int i = 0; i < n; i++)
```

```
        if (i > j)
```

```
            sum += 1;
```

```
        else {
```

```
            for (int k = 0; k < n; k++)
```

```
                sum = sum - 1;
```

```
        }
```

```
}
```

Q Calculate Order of Complexity using Master Theorem:

① $T(n) = 3T(n/2) + n^2$

② $T(n) = 4T(n/2) + n^2$

③ $T(n) = T(n/2) + n^2$

④ $T(n) = 16T(n/4) + n$

⑤ $T(n) = 2T(n/2) + n \log n$

⑥ $T(n) = 2T(n/2) + n/\log n$

⑦ $T(n) = 2T(n/4) + n^{0.51}$

⑧ $T(n) = 8T(n/3) + n^2 \log n$

⑨ $T(n) = 7T(n/3) + n^2$

⑩ $T(n) = 4T(n/2) + \log n$

⑪ $T(n) = 16T(n/4) + n!$

⑫ $T(n) = 3T(n/2) + n$

⑬ $T(n) = 3T(n/4) + n \log n$

⑭ $T(n) = 3T(n/3) + n/2$

⑮ $T(n) = 2T(n/2) + n$

Q3. Compare the fun^{ns} and determine which is greater

$$\textcircled{1} \begin{array}{l} f(n): n \log n \\ g(n): n^{\log n} \end{array}$$

$$\textcircled{2} \begin{array}{l} f(n): 2^{\log n} \\ g(n): n^{\sqrt{n}} \end{array}$$

$$\textcircled{3} \begin{array}{l} f(n) = 2^n \\ g(n) = 2^{2n} \end{array}$$

Q4. Determine Time complexity for:

$$\textcircled{1} T(n) = \begin{cases} T(n-1) + n & ; n > 0 \\ 1 & ; n = 0 \end{cases}$$

$\textcircled{2}$

```
void Test (int n)
{
    if (n > 0)
    {
        Test (n-1);
        Test (n-1);
    }
}
```

③ ^{void} Test (int n)
{

if (n > 1)
{

Printf ("ICT");
Test (n/2);

y y
y

$$\textcircled{4} T(n) = \begin{cases} 2T(n/2) + n & ; n > 1 \\ 1 & ; n = 1. \end{cases}$$