DATE ___/___/___

# Assignment - 1

Q1. Asymptotic Notation is used to describe the running time of an algorithm how much time an algorithm takes with a given input. There are 5 different notations : big O, big Ω and big Ω, small ω, small θ.

Big O Notation → $f(n) = O(g(n))$
$g(n)$ is tight upper bound of $f(n)$.

Big Ω Notation → $f(n) = \Omega(g(n))$
$g(n)$ is tight lower bound of $f(n)$

Big Ω Notation → $f(n) = \Omega(g(n))$
$g(n)$ is tight upper as well as tight lower bound of $g(n)$.

Small ω Notation → $f(n) = \omega(g(n))$
$g(n)$ is lower bound of $f(n)$

Small θ Notation → $f(n) = \theta(g(n))$
$g(n)$ is upper bound of $f(n)$.

DATE __/__/____

Examples →

Big O → O(logn) represents Binary search algorithm's upper bound.

Big Ω → a Bubble sort algorithm has a running time of Ω(N) because in the best case scenario, the list is already sorted & bubble sort will terminate after first iteration.

Big Θ → if for some algorithm the time complexity is represented by Expression $10n^2 + 50n$ then we can say its time complexity is $O(n^2)$ as it will ignore the lower order or constant part of the expression.

Q2.

```
for (i = 1 to n)
    { i = i * 2;
    }
```

$O(\log_2 n)$

DATE ___/___/_____

Q3. $T(n) = 3T(n-1)$ — ① if $n > 0$

$T(1) = 1$

putting $n = (n-1)$

$T(n-1) = 3T(n-2)$ — ②

putting value of $T(n-1)$ in ①

$T(n) = 9T(n-2)$ — ③

put $n = n-2$ in ①

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

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

put in ③

$T(n) = 9[3T(n-3)]$

$T(n) = 27\,T(n-3)$ — ④

$T(n) = 3^k\,T(n-k)$

$n-k = 1$

$n-1 = k$

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

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

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

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

DATE ___/___/___

4. $T(n) = 2T(n-1) - 1$ ——① if $n > 0$

put $n = n-1$

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

$T(n-1) = 2T(n-2) - 1$ ——②

put ② in ①

$T(n) = 2[2T(n-2) - 1] - 1$

$T(n) = 4T(n-2) - 2 - 1$ ——③

put $n = n-2$

$T(n-2) = 4T(n-2-2) - 3$

$T(n-2) = 4T(n-4) - 3$ ——④

put ④ in ③

$T(n) = 4[4T(n-4-2) - 3] - 3$

$T(n) = 16T(n-6) - 15$

$= 2^{2k} T(n-$

$T(n-2) = 2T(n-3) - 1$ ——④

put ④ in ③

$T(n) = 4[2T(n-3) - 1] - 2 - 1$

$= 8T(n-3) - 4 - 2 - 1$ ——⑤

DATE ___/___/___

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

put $n=(n-3)$

$T(n-3) = 2T(n-4) - 1 \quad -(6)$

put (6) in (5)

$= 8(2T(n-4)-1) - 4-2-1$

$= 16\,T(n-4) - 8-4-2-1$

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

$n-k = 1$

$n-1 = k$

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

$= 2^{n-1} - 2^{n-1} + 1$

$= 1 \Rightarrow O(1)$

---

5. int i=1, s=1 —(1)

while (s<=n) {

i++;

s=s+i;

printf("#");

}

$s = 1+2+3+\ldots k+1 + k(\frac{k+1}{2})+1$

$\frac{k(k+1)}{2} + 1 > n$

$k(k+1) + 2 > 2n$

$k^2 + k > 2n-2$

$k(k+1)$

DATE ___/___/___

$k^2 + k > 2n$ ÷2

$k^2 + k - (2n-2) > 0$

$k = \frac{-1 \pm \sqrt{1 - 4(2n-2)}}{2}$

$= \frac{-1 \pm \sqrt{1 - 8n - 8}}{2} = \frac{-1 + \sqrt{-8n-7}}{2}$

removing all constants

Ans = $O(\sqrt{n})$.

Q6. void function (int n)
{int i, j count = 0;
for (i = 1; i*i <= n; i++) → $\sqrt{n}$
for (j = 0 1, j <= n; j++) ~~2~~
count++;
}

$1^2 + 2^2 + 3^2 \ldots\ldots n$

$= \frac{n(n+1)(2n+1)}{6}$

$i^2 = n$

$i = \sqrt{n}$

$O(\sqrt{n})$

DATE ___/___/___

Q7.
```
void function(int n){
  int i, j, k, count=0;
  for(i=n/2; i<=n; i++)          — n/2
    for(j=1; j<=n; j=j*2)        log n
      for(k=1; k<=n; k=k*2)      — log n
        count++;
}
```

$\frac{n}{2} \log n \times \frac{n}{2} \log n$

$\Rightarrow \frac{n^2}{4} (\log n)^2$

$O(n^2 \log^2 n)$

Q8. $O(n^4)$

Q9. $O(\log n)$

Q10. Case 1: $n^k \leq c^n \Rightarrow k \log n \leq n \log c$

$\frac{\log n}{n} \leq \frac{\log c}{k} \Rightarrow O(c^n)$

Case 2: $c^n \leq n^k \Rightarrow n \log c \leq k \log n$

$\frac{\log c}{\log k} \leq \frac{\log n}{n} \Rightarrow O(n^k)$

Case 3: $n^k \rightarrow O(c^n)$ OR $c^n \rightarrow O(n^k)$.