DAA Tutorial 1

SK Sohail
F, 29

1. What do you understand by asymptotic notations. Define diff. Asymptotic notation with examples.

Asymptotic notations are the mathematical notations used to describe running time of an algorithm when the input tends towards a particular value.

Asymptotic Notations are mainly categorized into following 3 types:

(i) Big O Notation: It gives worst case complexity.

(ii) Omega Notation: It gives best case complexity.

(iii) Theta Notation: It gives avg case complexity.

Eg. Bubble sort algo. has $O(n)$ time complexity in best case & $O(n^2)$ time complexity in worst case & $O(n^2)$ in avg case.

2. What should be the time complexity of

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

$i = 1, 2, 4, 8, \ldots\ldots n \rightarrow GP$

$a_K = ar^{K-1}$ $\quad a=1, r=2$

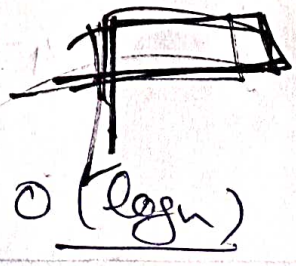
$a_K = 1.2^{K-1}$

$n = 2^{K-1}$

$\log_2 n = K-1$

$K = 1 + \log_2 n$

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



3. $T(n) = \{3T(n-1)$ if $n>0$, otherwise $1\}$

$T(0) = 1$

$T(n) = 3T(n-1)$ — (i)

put $n = n-1$ in eq. (i)

$T(n-1) = 3T(n-2)$ — (ii)

put (ii) in (i)

$T(n) = 3(3T(n-2)) = 3^2 T(n-2)$ — (iii)

put $n = n-2$ in eq. (i)

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

$T(n) = 3^2 \cdot 3T(n-3) = 3^3 T(n-3)$

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

let $n-k=0$

$T(n) = 3^n T(0) = T(n) = 3^n$

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

---

4. $T(n) = \{2T(n-1)-1$ if $n>0$ otherwise $1\}$

$T(n) = 2T(n-1)-1$ — (i)

$T(0) = 1$ — (ii)

put $n = n-1$

$T(n-1) = 2T(n-2)-1$ — (iii)

put (iii) in (i)

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

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

$= 2^2 T(n-2)-2-1$ — (iv)

put $n = n-2$ in (1)

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

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

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

$T(n) = 2^k T(n-k) - 2^{k-1} - 2^{k-2} - 2^{k-3} \ldots 2^0$

let $n-k=0$, $n=k$

$T(n) = 2^n T(n-n) - 2^{n-1} - 2^{n-2} - 2^{n-3} \ldots 2^0$

$T(n) = 2^n T(0) - 2^{n-1} - 2^{n-2} - 2^{n-3} \ldots 2^0$

$T(n) = 2^n - 2^{n-1} - 2^{n-2} \ldots 2^0$

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

$T(n) = 1$

$T(n) = O(1)$

5. What should be the time complexity of

```
int i=1, s=1;
while (s <= n){
  i++;  s=s+i;
  printf("#");
}
```

i=1	s=1	
i=2	s=3	s = 1+2
i=3	s=6	s = 1+2+3
i=4	s=10	s = 1+2+3+4

$s = 1+2+3+4+\ldots k = \frac{k(k+1)}{2} > n$

$\phi = \frac{k^2+k}{2} > 0$

$k > \sqrt{n}$

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

6. Time complexity of

```
void function (int n)
{
    int i, count = 0;
    for(i=1; i<=n; i++)
    count++;
}
```

$i = 1, 2, 3, \ldots n$

$i^2 = 1^2, 2^2, 3^2, \ldots n^2$

$i^2 \leq n$

$i \leq \sqrt{n}$

$a_k = a + (k-1)d$

$a = 1, d = 1$

$a_k \leq \sqrt{n}$

$\sqrt{n} = 1 + (k-1) \cdot 1$

$\sqrt{n} = k$

$\therefore T(n) = O(\sqrt{n})$