11 Jan'21

1/24) INTRODUCTION - ANALYSIS OF ALGORITHMS (70 min)
[25 min]

Asymptotic analysis is a theortical approach where we measure the order of growth in terms of input size.

θ	Thetha	==
O	Big O	<=
Ω	Omega	>=

$\theta(n^2)$ = $100n^2 + 2n$
= $10n^2 + 3000$
= $n^2 + 10^8 n$

$O(n^2)$ = $100n^2 + 2n$
= $2n + 5$
= 3

$\Omega(n^2)$ = $100n^2 + 2n$
= $5n^3 + 24$
= $10000n^8 + 8n^4 + 3$

Linear Search

Time Complexity : $O(n)$
Worst Case Time Complexity : $\theta(n)$

Q1) TC = $\lceil n/c \rceil$ = $\theta(n)$

Q2) TC = $\lceil n/c \rceil$ = $\theta(n)$

Q3) TC = $1, c, c^2, \ldots\ldots, c^K$ = $\theta(\log_c n)$

Q4) TC = $n/c,\ n/c^2, \ldots$ = $\theta(\log_c n)$

Q5) TC = $\log_c[\log_c(n)]$

Q6) TC = $O(n)$
TC in best case = $\theta(1)$
TC in worst case = $\theta(n)$

Q7) TC = $C_1 . m + C_2 . n$ = $O(m+n)$

Q8) TC = $\frac{n(n+1)}{2}$ = $n^2/2 + n/2$ = $\theta(n^2)$

* [20 min] Analysis of Recursion

Recursion Tree Method says that we write non-recursive part as root of tree and recursive part as children. Then we keep expanding child until we see a pattern. Other methods of finding time complexity is master method.

Q1) TC = $O(n.\log n)$ ← Merge Sort

Q2) TC = $\theta(2^N)$

⇒ $AP = \frac{n}{2}\left[2a + (n-1)d\right]$

⇒ $GP = a\left[\frac{r^n - 1}{r - 1}\right]$

⇒ $y = x^{\log_x(y)}$

Q3) TC = $\theta(\log n)$

Q4) TC = $\theta(n)$

In some cases, the recursion tree method can not give exact answer but can still give us a good upper bound.

Q1) TC = $O(n)$

Q2) TC = $O(2^n)$ ← Factorial

Hence, in relationships when the tree is dividing in un-equal time. Just consider the equation that takes slowest time to fill. Hence take the worst case scenario.