Shreya Shah
60004210043
A1 Comps
AOA

Assignment - 1

Q1] → Asymptonic Notations are a mathematical tool to find time or space complexity of an algorithm without implementing it in a programming language
It is a way of describing a major component of the cost of the entire algorithm. There are 3 notations:
(a) Big oh (O)
(b) Big Omego (Ω)
(c) Big theta (Θ)

→ The order of growth refers to rate at which the resources required by an algo increase as the size of input increases. It allows us to compare performance of different ~~got~~ algorithms and to make predictions about how the algo will scale as size of input increases

→ Efficiency classes are a way of describing the performance or time complexity of an algo or in terms of growth rate of its resource usage. Some examples are
(a) $O(\log n)$: logarithmic time. Its running time grows logarithmically with its input size
(b) $O(n)$: Linear Running time grows linearly with input size
(c) $O(n^2)$: quadratic Running time grows quadratically with input size.

Q2] ~~Ia~~ Karatsuba algorithm is a fast multiplication algo for long integers. It is a divide & conquer algo
The naive algo has a running time of $\Theta(n^2)$ while this has running time of $\Theta(n^{\log_2 3}) \approx \Theta(n^{1.585})$
Karatsuba stated that if we have to multiply 2 n-digit numbers x and y:
~~assume B is the base of m and m<n, first both.~~

P.T.O

Both numbers can be represented as ~~x_1, x_2 and y_1, y_2 i.e.~~

~~$x = x_1 B^m$ etc.~~ $x = a * B + b$

~~where (a, b, c, d are~~ $y = c * D + d$

where a, b, c, d are similar and B & D are powers of 10

such that B ≤ no. of digits in $x/2$

& D is largest power 10 such that D ≤ no of digits in $y/2$

$xy = (a * B + b) * c\,(c * D + d)$

$= a * c * B * D + (a * d + b * c) * B + b * d$

Now instead of separating:

$z_0 = b * d;$

$z_1 = (a+b) * (c+d); \; z_2 = a * c$

Thus $x * y = z_2 + B * D + (z_1 - z_2 - z_0) * B + z_0$

Eg: $x = 1234$ $\quad y = 5678$

$x = 12 * 100 + 34$ $\quad y = 56 * 100 + 78$

$z_0 = 34 * 78 = 2652$ $\quad z_1 = (12 + 34) * (56 + 78) = 4992$

$z_2 = 12 * 56 = 672$

$\therefore x * y = 672 * 10000 + (4992 - 672 - 2652) * 100 + 2652$

$= 7000052$

Q3]

Greedy	Dynamic
→ Choosing best option for best profit	→ Optimizing recursive backtracking solution
→ Time complexity is polynomial	→ Time complexity is polynomial
→ More efficient	→ less efficient
→ no guarantee of optimal solution	→ Always gives optimal solution
→ eg: Fractional Knapsack	→ Eg: 0/1 Knapsack

Shreya Shah
60004210045

Q4] All pair shortest path problem involves finding shortest path between all pairs of vertices in a weighted graph. One such algo is Floyd Warshall. Algo computes shortest path between all pair in $O(v^2)$ times $O(V^2)$ space complexity

Eg: $D^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 4 & \infty & 8 \\ 2 & \infty & 0 & 12 & 5 \\ 3 & 5 & \infty & 0 & \infty \\ 4 & \infty & \infty & 7 & 0 \end{array}$

$D^1 = \begin{bmatrix} 0 & 4 & \infty & 8 \\ \infty & 0 & 12 & 5 \\ 5 & 9 & 0 & 13 \\ \infty & \infty & 7 & 0 \end{bmatrix}$

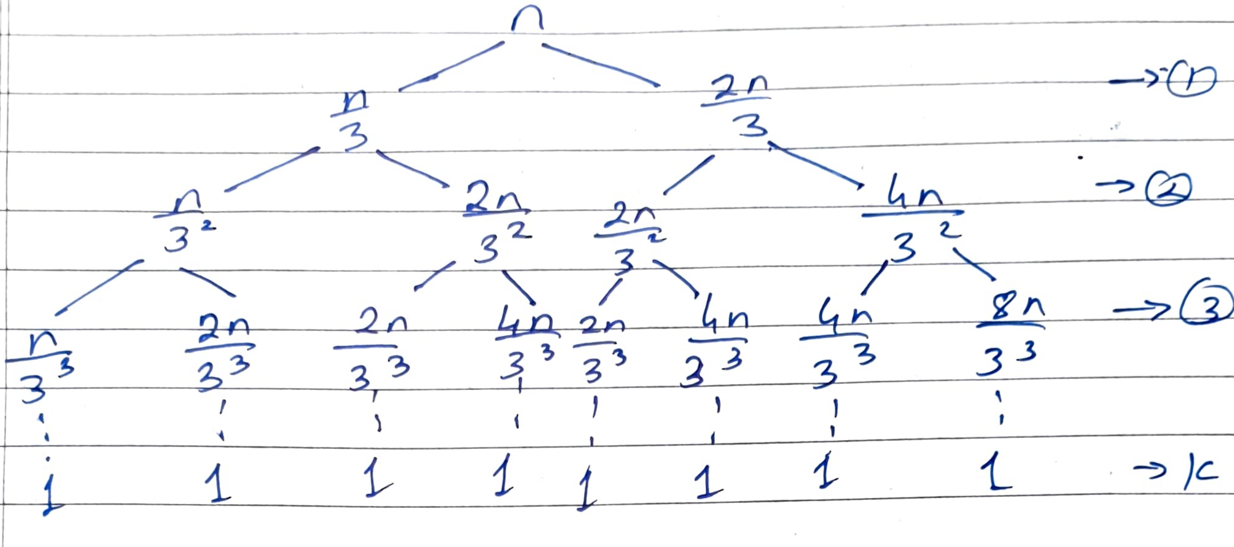
$D^2 = \begin{bmatrix} 0 & 4 & 16 & 8 \\ \infty & 0 & 12 & 5 \\ 5 & 9 & 0 & 13 \\ \infty & \infty & 7 & 0 \end{bmatrix}$

$D^3 = \begin{bmatrix} 0 & 4 & 16 & 8 \\ 17 & 0 & 12 & 5 \\ 5 & 9 & 0 & 13 \\ 12 & 16 & 7 & 0 \end{bmatrix}$

$D^4 = \begin{bmatrix} 0 & 4 & 15 & 8 \\ 17 & 0 & 12 & 5 \\ 5 & 9 & 0 & 13 \\ 12 & 16 & 7 & 0 \end{bmatrix}$

PTO

Q5] $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n^2$



Cost of each corresponding division of tree is n^2

Let height of tree $= K$

Value of not at level $1 = 2n/3$

$2 = 4n/3^2$.

$K = \left(\frac{2}{3}\right)^K n$

$\therefore \left(\frac{2}{3}\right)^K \cdot n = 1 \qquad \therefore K \log \frac{2}{3} + \log n = 0$

$\therefore K = \frac{\log n}{\log 3/2} = \log_{3/2} n$

$\therefore$ Total cost $= Kn^2 = \log_{3/2} n \cdot n^2 = n^2 \log_{3/2} n$

$\therefore$ Total time complexity $T(n) = O(n^2 \log_{3/2} n)$