Name: Akashdeep Soni
Roll No: 101641003 6
Subject: Design & Analysis of Algorithms
Branch & Section: CS 5 B

Date 22/09/20
Page

Assignment → 1

Answer → 1 (i)

Trichotomy: For any two real numbers a and b, exactly one of the following must hold: $a < b$, $a = b$, $a > b$. Although any two real numbers can be compared, not all functions are asymptotically comparable. That is, for two functions $f(n)$ and $g(n)$, it may be the case that neither $f(n) = O(g(n))$ nor $g(n) = O(f(n))$ holds.

Answer → 1 (ii)

Divide and Conquer Approach

Divide: the problem into a number of sub problems.
Conquer the sub problems by solving them recursively
- Base Case: If the sub problems are small enough, just solve them by brute force.

Combine the sub problems solution to give a solution to the orginal problem

Answer → 1 (iii)

As per the definition of Big Oh

$f(n) \leq cg(n)$

Hence

$2^{n+1} \leq 2^n \cdot 2$

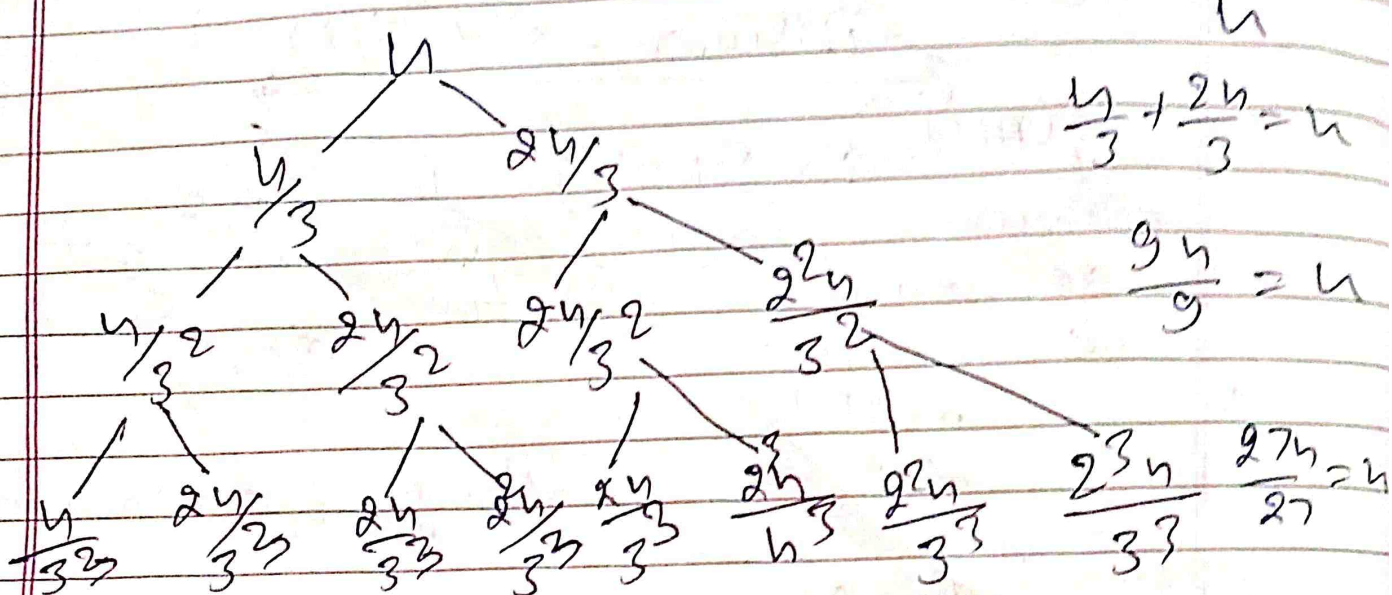
$2^{n+1} \leq 2 \cdot 2^n$ for all $n \geq 1$ & $c > 0$

Hence,

$f(n) \in O(g(n))$

Answer → 2 (i)

Q $T(n) = T(n/3) + T(2n/3) + n$



Left side = $\log_3 n$

Right side = $\log_{3/2} n$

Cost = n

Size of sub-problem = $\frac{n}{3}$ and $\frac{2n}{3}$

We can see n appearing n times

sum of all n = $n \log n$

Hence

$T(n) = O(n \log n)$

Answer → 2 (ii)

It means $T(n) = 7T(\frac{n}{2}) + n^2$ if $n \geq 1$

$T(n) = 1$ when $n = 1$

Put $n = n/2$ in eqn (1) we get

$T(\frac{n}{2}) = 7T(\frac{n}{4}) + (\frac{n}{2})^2$

Put the value of $T(\frac{n}{2})$ in eqn (i), we set

$$T(n) = 7\left[7T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)^2\right] + n^2$$

$$T(n) = 7^2 T\left(\frac{n}{4}\right) + \frac{7n^2}{4} + n^2 \quad \text{---(ii)}$$

Put $n = \frac{n}{4}$ in eqn (i) we set

$$T\left(\frac{n}{4}\right) = 7T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2$$

Put the value of $T(\frac{n}{4})$ in eqn (ii) we, set

$$T(n) = 7^2\left[7T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2\right] + \frac{7n^2}{4} + n^2$$

$$T(n) = 7^3 T\left(\frac{n}{8}\right) + \frac{7^2 n^2}{4^2} + \frac{7n^2}{4} + n^2$$

$$\text{---} \quad \text{(iii)}$$

$$T(n) = 7^k T\left(\frac{n}{2^k}\right) + \frac{7^{k-1} n^2}{4^{k-1}} + \cdots + \frac{7^2 n^2}{4^2} + \frac{7n^2}{4} + n^2$$

$$T(n) = 7^k T\left(\frac{n}{2^k}\right) + n^2\left[\frac{7^{k-1}}{4^{k-1}} + \frac{7^{k-2}}{4^{k-2}} + \cdots + \frac{7^2}{4^k} + \frac{7}{4} + 1\right]$$

$$T(n) = 7^k T\left(\frac{n}{2^k}\right) + n^2\left[\sum_{i=0}^{k-1}\left(\frac{7}{4}\right)^i\right] \quad \text{---(iv)}$$

As we know that sum of finite geometric series is

$$a + ar + ar^2 + \cdots + ar^n = \sum_{i=0}^{n} ar^i = \frac{a(r^{n+1}-1)}{(r-1)}$$

Here, n (Total number of terms) $= \log_2 n$, $a = 1$ and $r = 2$

Hence equation 4 can be written as follows.

$$T(n) = 7^{\log_2 n} + n^2 \left(\left(\frac{7}{4}\right)^{\log_2 n} - 1 \right)$$

$$T(n) = n^{\log_2 7} + n^2 \left(\frac{n^{\log (7/4)} - 1}{0.75} \right)$$

$$T(n) = n^{\log_2 7} + n^2 \left(\frac{n^{\log_2 7 - \log_2 4} - 1}{0.75} \right)$$

$$T(n) = n^{2.8} + n^2 \left(\frac{n^{2.8-2} - 1}{0.75} \right)$$

$$T(n) = n^{2.8} + n^2 \left(\frac{n^{0.8} - 1}{0.75} \right)$$

Hence $T(n) = O(n^{2.8})$

Answer -> 3

In quicle sort the pivot is formed in $\log n$. So complexity is $O(n \log n)$ but if we choose median as pivot $O(n)$ so complexity of quicle sort becomes $O(n \cdot n) = O(n^2)$

Quicle Sort.

```
Partition (l, h)
{
    pivot = A[l]
    i = l
    j = h
  while (i < j)
    {
      do
      { i++;
      } while (A[i] <= pivot);
      do
      {
        j--;
      } while (A[j] > pivot);
    if (i < j)
      Swap (A[i], A[j]);
    return j;
}
```

```
Quick Sort (l, h)
{
  if (l < h)
  {
    j = partition (l, h);
    Quicksort (l, j);
    Quicksort (j+1, h);
  }
}
```

Analysis:

Best Case: partition start from middle $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$

$T(n) = O(n \log n)$

Height of tree

$\frac{n/2}{n/2} = 1$

$\vdots$

$\frac{n}{2^k} = 1$

$n = 2^k$

$k = \log_2 n$

$T(n) = O(n \log(n))$

- It is not stable algo
- It is comparison sort
- It is not inplace algo