

Name : Snadha Kedia

Date of Examination : 16 December, 2021

Time of Examination : 9:00 am to 1:00 pm

Examination Roll no. : 20234757053

Semester : III

Unique Paper Code : 223401301

Title of paper : Design and Analysis of Algorithm

Email-Id : 200083@cs.du.ac.in

Mobile no. of student : 200083@cs.du.ac.in 9899519848

Question no. : 1

No. of pages : 3

Name of the program : MCA

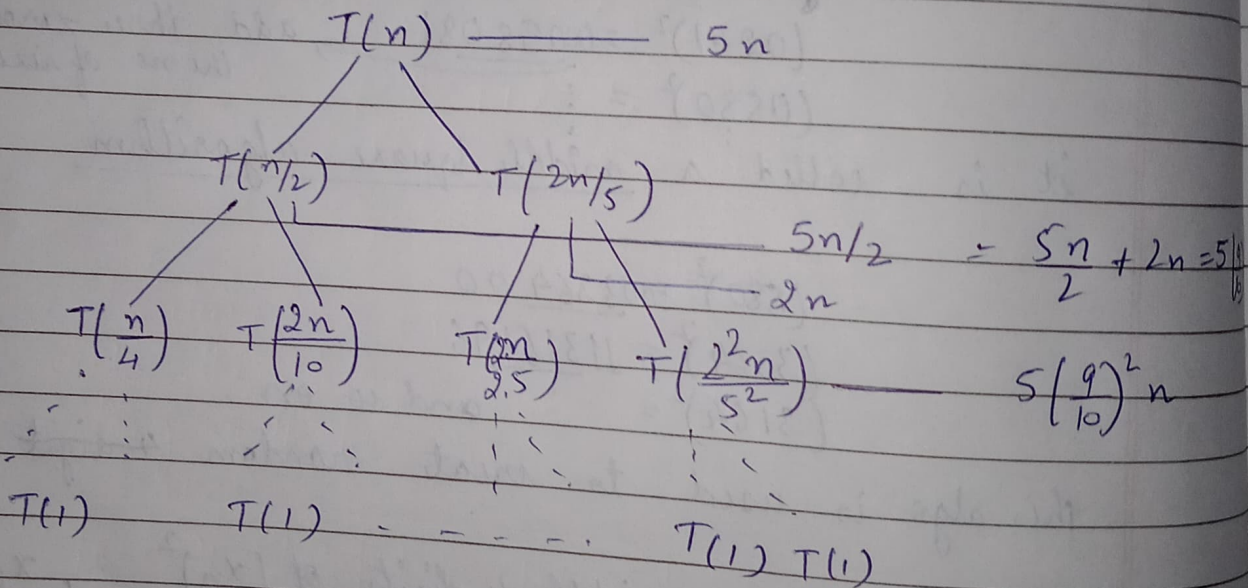
Name of the Department : DVCS

Ans 6(a) a) True, because a NP complete problem S is polynomial time reducible to R .

b) True, because no NP complete problem can be solved in polynomial time. Because, if one NP complete problem can be solved in polynomial time then all NP problem can be solved within that time. If that is the case, then NP and P set become same which contradicts the given condition.

(6) $T(n) = T(n/2) + T(2n/5) + 5n$

Recursion tree:



height of the tree depends on left most part of tree
 $\therefore O(\text{height of tree}) = \log_2 n - 1$

$$5n + 5\left(\frac{n}{10}\right) + 5\left(\frac{n}{10}\right)^2 \dots \log_2 n \text{ times}$$

$$O(n \left(1 + \frac{9}{10} + \left(\frac{9}{10}\right)^2 + \dots \right) \log n \text{ times})$$

$$\therefore S_n \left(\frac{1 - \left(\frac{9}{10}\right)^{\log_2 n}}{1 - \frac{9}{10}} \right)$$

$$= S_n \left(1 - \left(\frac{9}{10}\right)^{\log_2 n} \right) \times 10$$

$$\text{for } n \rightarrow \infty, \left(\frac{9}{10}\right)^{\infty} \rightarrow 1$$

$$\therefore O(T(n)) = \theta(n)$$

6 (c)

pattern: abefabefgabefg

assuming int i = 1, int j = 0

int[] lps array = array of length of string

```
while (i < s.length) {
    if (ch of s at i == ch of s at j) {
        j++;
        lpsArr[i] = j;
        i++;
    } else if (j > 0) {
        j = lpsArr[j-1];
    } else {
        lpsArr[i] = 0;
        i++;
    }
}
```

return lps Array;

with the following algo the lps Array for the given string will be

0	0	0	0	1	2	3	4	0	1	2	3	4	0
a	b	e	f	a	b	e	f	g	a	b	e	f	g