No. of Printed Pages : 4 **MMTE-002**

M. SC. (MATHEMATICS WITH APPLICATIONS IN COMPUTER SCIENCE) M. Sc. (MACS)

Term-End Examination

June, 2020

MMTE-002 : DESIGN AND ANALYSIS OF ALGORITHMS

Time : 2 Hours *Maximum Marks : 50*

***Note :** Question No. **6 is compulsory.** Answer any **four** questions from Question Nos. **1** to **5**. Calculators are not allowed.*

---

1. (a) Sort the following numbers using the merge sort algorithm, showing all the steps you use in the process : 5

15, 32, 88, 78, 66, 23, 79, 25, 42, 37

(b) Construct a 2-3-4 B-tree by inserting the following numbers in the order given. Show all the steps you have used in the process : 5

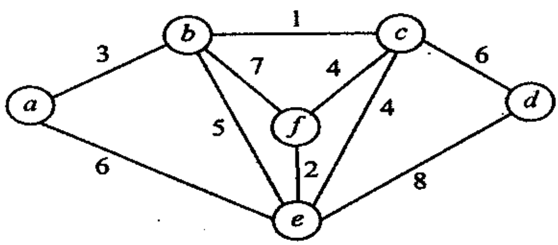
3, 1, 4, 2, 8, 7, 9, 6, 5, 11

2. (a) Sort the following numbers using the heap sort algorithm, showing all the steps involved : 5

25, 35, 11, 12, 89, 68, 23

(b) Determine an LCS of AABCBBDAAC and ACBDABBACA, using the dynamic programming approach, showing all the steps involved. 5

3. (a) Find the minimum spanning tree for the following graph using Kruskal's algorithm : 5



(b) Find all the solutions to the equation $15x \equiv 12 \pmod{39}$. Show all the steps you have used in the process. 5

4. (a) Express the following polynomials in point-value representation : 5

$$f(x) = x^2 - x + 1$$

$$g(x) = x^3 - x^2 + x + 2.$$

Also find the point-value representation of $f(x)g(x)$, and hence find the coefficient representation of $f(x)g(x)$.

(b) Solve the recurrence relation :

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + O(n),$$

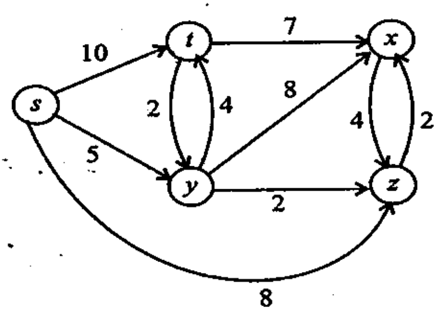
using the recursion tree method. 5

5. (a) Give an example, with justification of each of the following : 4

   (i) Optimal substructure

   (ii) Overlapping sub-problems

(b) Apply Dijkstra's algorithm for the following example with s as source vertex : 6

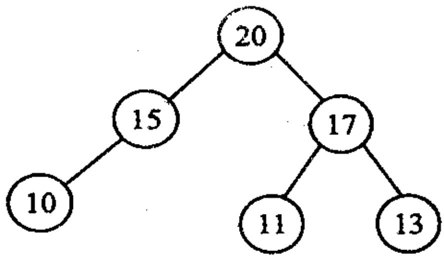


6. Which of the following statements are true and which are false ? Justify your answer with a short proof or a counter example : 10

(i) $2^n = O((2.5)^n)$.

(ii) Quick sort is always faster than counting sort when applied on an array of numbers.

(iii) The following is an example of max-heap :



(iv) In any binary search tree with n-nodes searching for a key can be done in O ($\log n$) time.

(v) The value of the Euler phi-function $\phi(n)$ is always even for $n > 2$.

560