

Note :

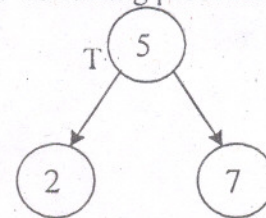
1. Q1 is compulsory and attempt any 4 from remaining questions.
2. Answer to the point.
3. Assumptions should be highlighted and justified.
4. Draw diagram to explain the theory wherever necessary.
5. Start major question on new page and maintain the order of questions.
6. For every question and its answer, title it with details e.g. Q3/a/2/i if there are subsections.
7. While stapling the supplements, take care that questions numbers and answers are not obscured.

1. (a) i) State how recursion is disadvantageous wrt time and space. Consider a tree T as shown in the diagram. State how many times the following procedure is invoked recursively and output of the program. 10

```

traversal(tree T)
{
    if (T!=NULL) {
        traversal(T->left);
        traversal(T->right);
        print(T->data);
    }
}

```



ii) Compare the time complexities of the following algorithms giving their complexities in terms of big-O,  $\Omega$ ,  $\theta$ . Quick sort, Heap sort, Shell's sort, Insertion sort.

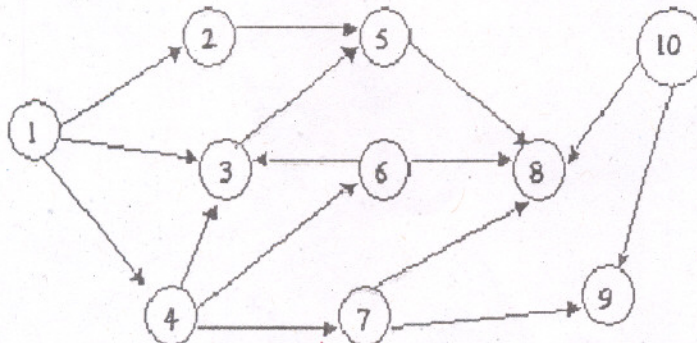
- (b) i) State applications of graph theory. 10  
 ii) Define the following terms with respect to spanning trees Give example.  
 (a) Tree edge (b) cross edge (c) forward edge (d) backward edge.

2. (a) Show how the Sorting of the given list is done using insertion sort by every pass 10

567 456 455 324 213 665 777 907 990 654

Give the time and space complexity for every case for insertion sort.

- (b) Give algorithm / pseudo code for Breadth First Search and Use Breadth First Search technique to give a spanning tree for the following. Consider the node 1 as source. 10



3. (a) Explain Topological Sorting 10  
 (b) What is Hashing? What are collisions? Explain at least 3 collision-handling techniques with example. 10

- techniques with example.
4. (a) Explain B+ trees with an example and show how insertions can be done in it. 10  
State its applications.
- (b) Give the possible solutions for improving the efficiency of sequential search. 10
5. (a) Explain greedy approach of solving problems. Give an example of any such 10  
algorithm, which follows greedy approach.
- (b) Explain backtracking with example and comment on complexity 10
6. (a) Explain Index Sequential searching method. Give its complexity and 10  
application.
- (b) Explain AVL RL and LR rotation with suitable example 10
7. (a) State 10 popular tree applications and Draw various trees types. 10
- (b) Suggest the data structure to store the information such that it will be faster to 10  
search than binary search.
-