KGiSL Institute of Technology

(An Autonomous Institution)

Affiliated to Anna University, Approved by AICTE, Recognized by UGC,
Accredited by NAAC & NBA (B.E-CSE,B.E-ECE, B.Tech-IT),
365, KGiSL Campus, Thudiyalur Road, Saravanampatti, Coimbatore – 641035.

Department of IT / AIML / ECE / CSE(CYS)

Data Structures and Algorithms

IA – 2 Important Questions

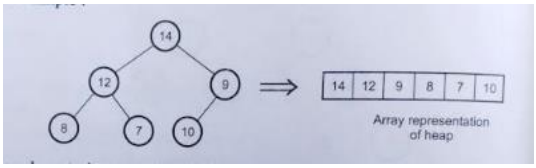
PART – A

1. Mention some problems for which heaps are more applicable.
2. Write a short note on multiway search trees.
3. What is the balance factor in an AVL tree and why is it important?
4. Differentiate between a binary search tree and a multiway search tree in terms of node structure.
5. Distinguish binary search tree and AVL tree.
6. State the principle of optimality in dynamic programming.
7. Define a Directed Acyclic Graph (DAG) and give one application.
8. What is a Minimum Spanning Tree (MST) and name one algorithm used to find it?
9. Mention some problems for which heaps are more applicable.
10. What is significance of articulation points with example?
11. Differentiate between weakly connected graph and strongly connected graph.
12. State the running time for topological sort.
13. Analyze the time complexity of merge sort.
14. Recall the concept of dynamic programming
15. Enumerate graph ADT
16. Explain minimum spanning tree
17. Infer the meaning of DAG
18. Perform one pass of bubble sort on the array [5, 3, 8, 4, 2]. What is the resulting array?
19. State the best-case and worst-case time complexities of insertion sort.
20. How does a high load factor affect the performance of a hash table?
21. Compute the hash value of key 23 using the hash function h(k) = k mod 10.
22. Enumerate the types of sorting.
23. Identify the best-case and worst-case time complexities of Linear Search.
24. Applications of linear and binary search?
25. List out the collision handling technique?
26. Relate between internal and external sorting.
27. List the limitations of linear probing.
28. Identify the steps involved in selection sort.
29. Why hashing is needed?

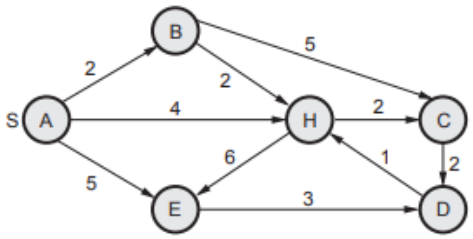
PART – B

1. Write a routine for AVL tree insertion. Insert the following elements in the empty tree, how do you balance the tree after each element insertion. Elements: 2, 5, 4, 6, 7, 9, 8, 3,1, 10
2. Insert the sequence of keys [40, 20, 30, 35, 25, 80, 32, 100] into a Max-Heap. Show step-by-step heapification after each insertion.
3. Explain AVL tree with examples of Rotations with examples.
4. Discuss the heapify process with an example by converting [10, 5, 20, 2, 4, 8] into a min-heap.

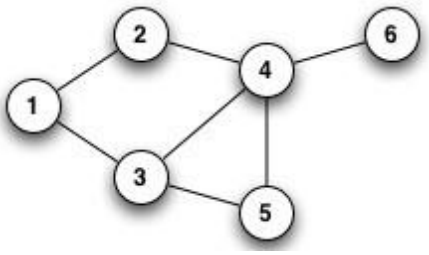
5. Write an algorithm to insert an element in the heap.and perform the following opeartions. (a) insert 18 to the heap.



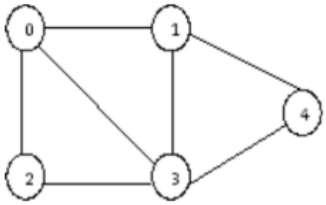
6. Using Dijikstra's Algorithm find the shortest path from the source node A.



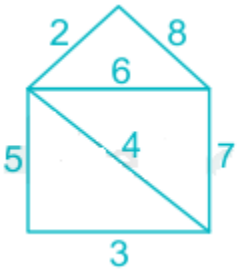
7. Develop the following graph using BFS and DFS.



8. Relate the following graph using BFS and DFS.

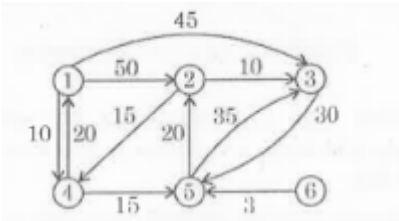


9. Discuss Prim's algorithm for computing the minimal spanning tree weighted undirected graph with example.



10. (i) Explain representation of graph with example.
    (ii) Write a python code for traversing a graph using depth first search traversal.

11. Describe topological sorting with a pseudo-code
12. State Dijkstra's Algorithm and identify the minimum cost and shortest path for the following graph.



13. Use Linear Search to find key 15 in the array [3, 7, 11, 15, 18, 21, 25, 30, 33, 40]. Show step-by-step comparisons and analyze best, worst, and average cases.

14. Using linear probing, insert keys 12, 22, 32, 42 into a hash table of size 10 using h(k) = k mod 10. Show the table after each insertion and explain how collisions are resolved.
15. Using quicksort Let's sort the array [10, 80, 30, 90, 40, 50, 70], with the last element as the pivot.
16. Write a python program to perform searching operations using linear and binary search.
17. Consider a hash table with 9 slots. The hash function is h(k)=k mod 9. The following keys are inserted in the order 5, 28, 19, 15, 20, 33, 12, 17, 10. Draw the contents of the hash table when the collisions are resolved by (i) chaining (ii)Linear Probing (iii) Double hashing
18. i)Examine the selection sort algorithm and show how it arranges the following array of ten elements in sorted order using python implementation. Analyze the complexity of the algorithm. A[1:10]=(31,28,17,65,35,42,86,25,45,52)
    ii)Sort the following set of number using Bubble sort with python implementation. A=(432, 8, 530, 90, 88, 231, 11, 45, 677, 199)