S.E. Sem 4 Comp. Analysis of Algorithms. 19/12/08

Con. 5699-08. **RC-6110**

(3 Hours) **[Total Marks : 100**

N.B.: (1) Question No. **1** is **compulsory**.
(2) Attempt any **four** questions from the remaining **six** questions.
(3) Assumption should be **highlighted** and **justified**.

1. (a) Compare the time complexities of the following algorithms giving their complexities in terms of big- 0, Ω, θ. **5**
   (i) Quick sort (ii) Heap sort (iii) Shell sort (iv) Insertion sort.
   (b) Define the following terms with respect to spanning trees. Give example. **5**
   (i) Tree edge (ii) Cross edge (iii) Forward edge (iv) Backward edge.
   (c) Write a routine to delete a word from a tries. **5**
   (d) State the applications of graph theory. **5**

2. (a) To prove that complexity of heap sort is $0(n \log_2 n)$. Also write a program for heap sort. **10**
   (b) To implement the binary search, prove that the complexity of binary search is $0(\log_2 n)$. **10**

3. (a) The following values are to be stored in a hash table **10**
   25, 42, 96, 101, 102, 162, 197.
   Describe how the values are hashed by using division method of hashing with a table size of 7. Use chaining as the method of collision resolution.
   (b) To find minimum cost Spanning tree for the given graph in **figure 3·1**, using Prim's and Kruskal's Algorithm. **10**

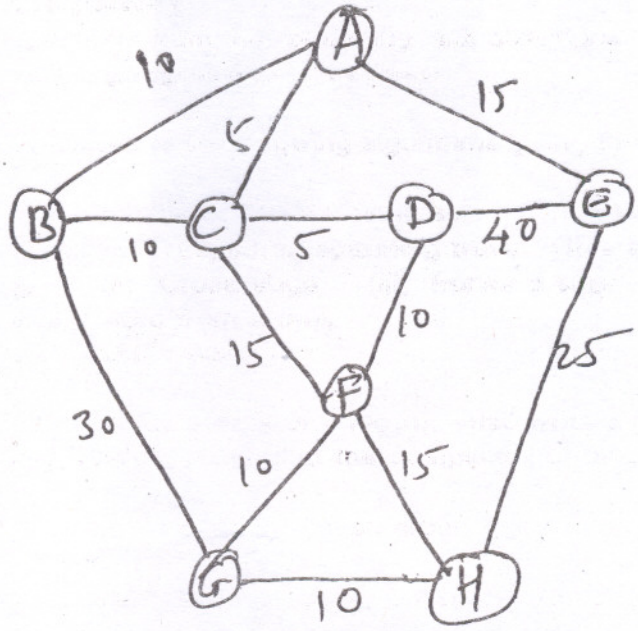

Figure 3.1

4. (a) What is AVL tree ? Explain four cases to balance AVL tree. To construct the AVL tree for the given numbers to be inserted one by one. **10**
   3, 5, 11, 8, 4, 1, 12, 17, 2, 6, 10.
   (b) To explain min/max heap and then implement heap sort. Also derive its time complexity. **10**

5. (a) Write a functions to implement DFS and BFS graph searching methods. **10**
   (b) Explain back tracking method. Develop an algorithm for finding solution to N-queen problem. **10**

5. (a) Write a functions to implement DFS and BFS graph searching methods. **10**
   (b) Explain back tracking method. Develop an algorithm for finding solution to N-queen problem. **10**

6. (a) Explain Greedy approach to solve knapsack problem. Find optimal solution to knapsack instance $n = 3$, $m = 20$, **10**
   $(P_1, P_2, P_3) = (25, 24, 15)$ and $(W_1, W_2, W_3) = (18, 15, 10)$.
   (b) Explain B+ tree with an example and show how insertions can be done in it. State its applications. **10**

7. (a) Show that any n-node Binary tree can be converted to any other n-node binary tree using O(n) rotations. **10**
   (b) Design a divide-and-conquer algorithm for finding the minimum and maximum element of n numbers using no more than $3^n/_2$ comparisons. **10**
