Roll No.: _ __ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Amrita Vishwa Vidyapeetham

B.Tech. First Assessment – January 2019

Fourth Semester

(Computer Science and Engineering)

**15CSE211 Design and Analysis of Algorithms**

Time: Two hours Maximum: 50 Marks

**Answer all questions**

**Course Outcomes**

| **CO** | **Description** |
|---|---|
| CO1 | Understand the correctness and analyze complexity of algorithms |
| CO2 | Understand various algorithmic design techniques and solve classical problems |
| CO3 | Solve real world problems by identifying and applying appropriate design techniques |
| CO4 | Analyze and map a given real world problem to classical problems and find solution |
| CO5 | Analyze the impact of various implementation choices on the algorithm complexity |

1. Answer the following

   (a) Explain how you would put a deck of cards in order by suit (in the order spades, hearts, clubs, diamonds) and by rank within each suit, with the restriction that the cards must be laid out face down in a row, and the only allowed operations are to check the values of two cards and to exchange two cards (keeping them face down). (CO4) [5]

   (b) Analyze the below code fragment and illustrate the time complexity calculation on it. (CO1) [3]

```
int count = 0;
for (int i = N; i > 0; i /= 2)
{
for (int j = 0; j < i; j++)
{
count++;
}
}
```

2. Consider the following problem. The input consists of n skiers with heights p1,...,pn, and n skies with heights $s_1, \ldots, s_n$. The problem is to assign each skier a ski to to minimize the average difference between the height of a skier and his/her assigned ski. That is, if the ith skier is given the α(i)th ski, then you want to minimize: (CO3)

$$\frac{1}{n}\sum_{i=1}^{n}\left|p_i - s_{a(i)}\right|$$

   (a) Consider the following greedy algorithm. Find the skier and ski whose height difference is minimized. Assign this skier this ski. Repeat the process until every skier has a ski. Is this algorithm optimal? If yes prove formally or give a counter example. [4]

   (b) Consider the following greedy algorithm. Give the shortest skier the shortest ski, give the second shortest skier the second shortest ski, give the third shortest skier the third shortest ski, etc. Prove or disprove that this algorithm is correct. [4]

3. Answer the following

   (a) Professor F. Lake suggests the following algorithm for finding the shortest path from node s to node t in a directed graph with some negative edges: add a large constant to each edge weight so that all the weights become positive, then run Dijkstra's algorithm starting at node s, and return the shortest path found to node t. Is this a valid method? Either prove that it works correctly, or give a counterexample. (CO2) [4]

(b) Suppose that instead of a linked list, each array entry Adj[u] is a hash table containing the vertices for which (u,v) in E. If all edge lookups are equally likely, what is the expected time to determine whether an edge is in the graph? What disadvantages does this scheme have? Suggest an alternate data structure for each edge list that solves these problems. Does your alternative have disadvantages compared to the hash table? Explain. (CO5)[5]

3. For each of the following pairs of functions f(n) and g(n), either f(n)=O(g(n)) or g(n)=O(f(n)), but not both. Determine which is the case. (CO1)[8]

(a) $f(n) = (n^2 - n)/2$, $g(n) = 6n$

(b) $f(n) = n + n\log n$, $g(n) = n\sqrt{n}$

(c) $f(n)= 2(\log n)^2$, $g(n) = \log n + 1$

(d) $f(n)= n^2 + 3n + 4$, $g(n) = n^3$

4. Answer the following: (CO2)

(a) Perform Huffman coding on the following data: [5]

| **Character** | A | B | C | D | _ |
|---|---|---|---|---|---|
| **Probability** | 0.4 | 0.1 | 0.2 | 0.15 | 0.15 |

i. Encode the text ABACABAD using the Huffman tree generated from the data.

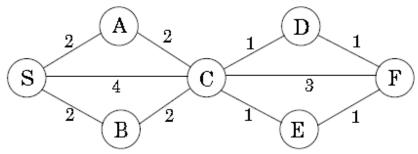
ii. Decode the text whose encoding is 100010111001010.

(b) Consider a "reversed" Kruskal's algorithm for computing a MST. Initialize T to be the set of all edges in the graph. Now, consider edges from largest to smallest cost. For each edge, delete it from T if that edge belongs to a cycle in T. Assuming all the edge costs are distinct, does this new algorithm correctly compute a MST? Prove using proof of contradiction or give a counter example. [4]

5. In cases where there are several different shortest paths between two nodes (and edges have varying lengths), the most convenient of these paths is often *the one with fewest edges*. For instance, if nodes represent cities and edge lengths represent costs of flying between cities, there might be many ways to get from city s to city t which all have the same cost. The most convenient of these alternatives is the one which involves the fewest stopovers. Accordingly, for a specific starting node s, define

best[u] = minimum number of edges in a shortest path from s to u:

In the example below, the best values for nodes S; A;B;C;D;E; F are 0; 1; 1; 1; 2; 2; 3, respectively.




*Input:* Graph G = (V;E); positive edge lengths le; starting node sεV .

*Output:* The values of best[u] should be set for *all* nodes uεV .

Give an efficient algorithm for the following problem. Explain your algorithm using examples, show its correctness, and derives its running time. (CO3) [8]