

## Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Lab Final, Course Code: CSE-2201

All questions have equal marks

Name: -----

Roll no: -----

1. Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of  $n$  (i.e.,  $n$  should be  $>1000$ ) which the number of elements in the list to be sorted and plot a graph of the time taken versus  $n$ . The number of element  $n$  to be sorted should be read from a file "quick.txt" and generate the lists of numbers randomly.
2. Solve the experiment defined at Q1 using Merge Sort algorithm. Compare the required time of merge sort with the Quick sort approach.
3. Implement 0/1 and fractional Knapsack problem using Dynamic and Greedy approach Programming. Consider the following dataset for the above problems: A knapsack of size 11kg, there are 5 items need to be chosen for the given task and their weight and values are as follows: weights ( $w_1, w_2, w_3, w_4, w_5$ )=(1, 2, 5, 6, 7) and values ( $v_1, v_2, v_3, v_4, v_5$ )=(1, 6, 18, 22, 28). Using the 0/1 and fractional Knapsack approach fill up the knapsack to maximize the values.
4. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm. I appreciate to use the file **graph.txt** for reading the following input.

Sample Input:

```
Enter the number of nodes:5

Enter the cost matrix:
0      5      12      17      999
999    0      999      8       7
999    999    0       9      999
999    999    999     0      999
999    999    999    999     0

Enter the source matrix:1
```

Sample Output:

```
Shortest path:
1->2,cost=5
1->3,cost=12
1->4,cost=13
1->5,cost=12
```

5. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm. I appreciate to use the file **kgraph.txt** for reading the following input.

Sample input:

```
Implementation of Kruskal's algorithm

Enter the no. of vertices
4

Enter the cost adjacency matrix
0      20      10      50
20      0      60      999
10      60      0      40
50      999      40      0
```

Sample output:

```
The edges of Minimum Cost Spanning Tree are

1 edge (1,3) =10
2 edge (1,2) =20
3 edge (3,4) =40

Minimum cost = 70
```

6. Print all the nodes reachable from a given starting node in a digraph using BFS method.

Sample input:

```
Enter the number of vertices:4

Enter graph data in matrix form:
0      1      1      1
0      0      0      1
0      0      0      0
0      0      1      0
```

Sample output:

```
Enter the starting vertex:1

The node which are reachable are:
2      3      4      _
```

7. Implement N Queen's problem using Back Tracking. N should be read from a file **queen.txt** and N should not be <4. Determine at least two solutions and print the number of iteration needed for each solution.

Sample input: 4

Sample Output:

```
Solution #1:
*      Q      *      *
*      *      *      Q
Q      *      *      *
*      *      Q      *

Solution #2:
*      *      Q      *
Q      *      *      *
*      *      *      Q
*      Q      *      *
```

- Special Instructions:**
- i. Save your solutions to a .zip file and with name Roll\_name.
  - ii. Inside the .zip file create folder for each of the question (i.e., 1, 2, etc.) and put the code and input file inside the same folder. So that it can be run from here for test without any modification.
  - iii. If your code and algorithm is far different from others you will receive a bonus number of 10. If it matches with others you will lose that bonus number and also be penalized accordingly.

Submit by: Tomorrow 10:30 PM