Sample Output:

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology

Lab Final, Course Code: CSE-2202, Date: 07-08-2021, Submission Date: 09-08-2021

All questions have equal marks

Name: ---------- Roll no: ----------, Section: ----------

1. Sort a given set of elements using the Quicksort method and determine the time required to sort the elements in millisecond. Repeat the experiment for 3 different values of ***n*** (i.e., ***n*** should be >1000 for each case) which the number of elements in the list to be sorted and create a bar graph of the time taken versus ***n***. The number of element ***n*** to be sorted should be read from a file "**quick1.txt, quick2.txt, quick3.txt**"""" and generate the lists of numbers randomly.
2. Solve the experiment defined at Q1 using Merge Sort algorithm. Create a bar graph for same problem and compare the required time of merge sort with the Quick sort algorithm.

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. Print out the optimal solution for each of the Knapsack problem and compare the results.

4. **Problem**: Sorting in linear time: Counting Sort
   i. Generate N random integers within range 0 to 10000 in a file named input.txt.
   ii. Implement bubble sort to sort the numbers (from input.txt) and count the time.
   iii. Implement counting sort to sort the numbers (from input.txt) and count the time.
   iv. Increase the value of N (up to 100000) and plot the performance
   v. Write your comments on the findings.

5. 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
```

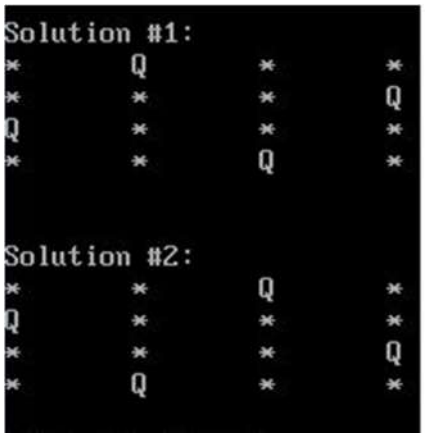
   Sample output:

```
Enter the starting vertex:1

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

6. 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 Output:
Sample input: 4



Special Instructions:

i. Save your solutions to a .zip file and name the file as roll_name_Section.pdf.

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 special reward in assessment. If it matches with others, marks will be incurred appropriately and also be penalized accordingly.

iv. All solutions/file should be hand written and report should contain the algorithm, code, sample input and output. Images can be used for the sample input and output. A short description of data is accepted if it is too large.

Submit by: 09-08-2021, 10:30 PM