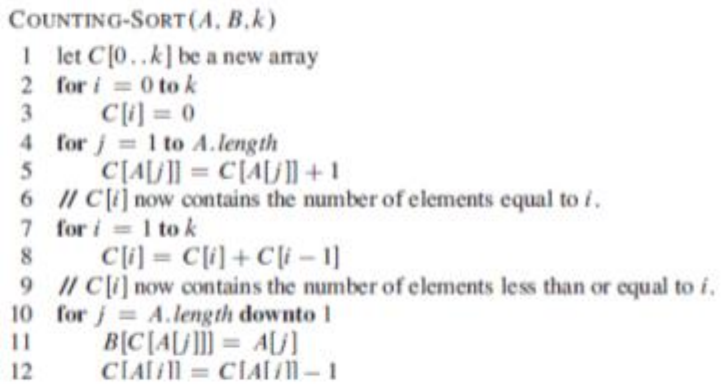
Amrita Vishwa Vidyapeetham
Amrita School of Computing
Department of Computer Science and Engineering
19CSE302 – Design and Analysis of Algorithms

Lab Assignment - 2 **Topic: Sorting & Graphs** **Date: 24th August, 2023**

Question 1: Suppose that we were to rewrite the for loop header in line 10 of the COUNTINGSORT as 10 for j = 1 to A.length. Write a program, so that the algorithm still works properly.



```
COUNTING-SORT(A, B, k)
1   let C[0..k] be a new array
2   for i = 0 to k
3       C[i] = 0
4   for j = 1 to A.length
5       C[A[j]] = C[A[j]] + 1
6   // C[i] now contains the number of elements equal to i.
7   for i = 1 to k
8       C[i] = C[i] + C[i - 1]
9   // C[i] now contains the number of elements less than or equal to i.
10  for j = A.length downto 1
11      B[C[A[j]]] = A[j]
12      C[A[j]] = C[A[j]] - 1
```

Compare the results of the algorithm given and modified algorithm as told above. Plot the input vs time graph for large values of n

Question 2: Problem statement: John has to attend some conferences. There are N cities numbered from 1 to N and conferences can be held in any city. John lives in city1 and he will attend the conference as per schedule.
Design and implement an algorithm with minimum time complexity that will find the shortest path from John's location to any conference's location. Consider all cities are connected. Graph is a simple graph, no parallel edges or self loop. It is not mandatory that the graph should be complete.

Input Format:

First line: Two space-separated integers denoting N (The number of cities) and M(number of possible routes between cities).

Next M lines: Each line contains three space-separated integers x, y & t. t defines the distance between city A and city B.

Output Format:

First line: Two space-separated integers denoting N (The number of cities)and M(number of routes in final graph).

Next M lines: Each line of the subsequent lines contains the values of x, y and c. x is city1. c defines the distance between city 1 and city y.

Constraint

1 <= N <= 1000
1 <= M <= 2000
1≤x,y≤N

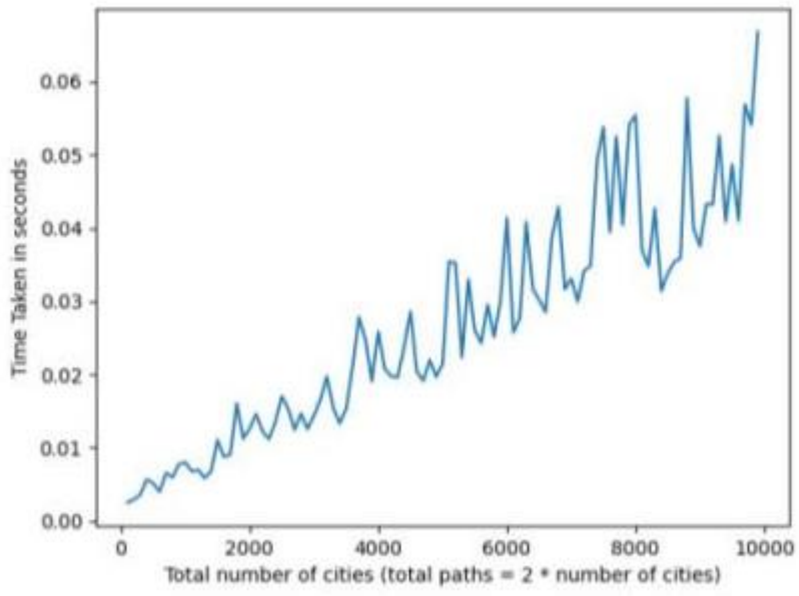
Example:
input
6 8
1 2 1
1 4 5
2 3 2
2 5 1
2 4 2
3 6 2
3 5 3
5 6 2

output

```
6 5
1 2 1
1 3 3
1 4 3
1 5 2
1 6 4
```

You can use MS excel/matplotlib to plot the graph. Consider relation between edges and vertices and then plot the graph between time taken and no. of vertices. Submit the document including algorithm, time complexity, graph and screenshots of output.

Please test your solution for Adjacency Matrix representation and Priority Queue representation. Plot the time complexity as shown below in the sample



Sample Graphs for testing:
https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html
You can generate synthetic graphs using networkx package of python as follows

```python
import networkx as nx
G = nx.Graph()
G.add_edge("a", "b", weight=0.6)
G.add_edge("a", "c", weight=0.2)
G.add_edge("c", "d", weight=0.1)
G.add_edge("c", "e", weight=0.7)
G.add_edge("c", "f", weight=0.9)
G.add_edge("a", "d", weight=0.3)
```

```
print(G)
```

```
Graph with 6 nodes and 6 edges
```

```
print(G.nodes())
```

```
['a', 'b', 'c', 'd', 'e', 'f']
```

```
print(G.edges())
```

```
[('a', 'b'), ('a', 'c'), ('a', 'd'), ('c', 'd'), ('c', 'e'), ('c', 'f')]
```

Question 3: Drunken Donuts, a new wine-and-donuts restaurant chain, wants to build restaurants on many street corners with the goal of maximizing their total profit. The street network is described as an undirected graph G = (V, E), where the potential restaurant sites are the vertices of the graph. Each vertex u has a nonnegative integer value pu, which describes the potential profit of site u. Two restaurants cannot be built on adjacent vertices (to avoid selfcompetition). You are supposed to design an algorithm that outputs the chosen set U ⊆ V of sites that maximizes the total profit u∈U pu. First, for parts (a)–(c), suppose that the street network G is acyclic, i.e., a tree.

(a) Consider the following "greedy" restaurant-placement algorithm: Choose the highest- profit vertex u0 in the tree (breaking ties according to some order on vertex names) and put it into U. Remove u0 from further consideration, along with all of its neighbors in G. Repeat until no further vertices remain. Give a counterexample to show that this algorithm does not always give a restaurant placement with the maximum profit.

(b) Suppose that, in the absence of good market research, DD decides that all sites are equally good, so the goal is simply to design a restaurant placement with the largest number of locations. Give a simple greedy algorithm for this case, and prove its correctness.