Design and Analysis of Computer Algorithms

Homework #5: Due 11:59 PM, 22 May 2022 (Sunday).

Problem #1 (40 points). Programming

Shortest path with Dijkstra algorithm

Given a weighted undirected graph in which the vertices are indexed from 1 to n. You are required to use Dijkstra algorithm to find the shortest path between vertex 1 and vertex n.

Input:

Input is read from file **input.txt**

- The first line contains two integers n and m, where n is the number of vertices and m is the number of edges.
- The next m lines contain one edge each in form a_i, b_i and w_i, where a_i, b_i are edge endpoints and w_i is the length of the edge.

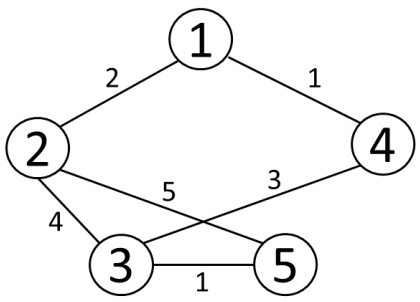
Output:

Output is written to file **output.txt**

- Write the path from vertex 1 to vertex n. If there are multiple paths, write any of them. If there is no path, write -1.

Example:

input.txt	output.txt
5 6 1 2 2 2 5 5 2 3 4 1 4 1 4 3 3 3 5 1	1 4 3 5



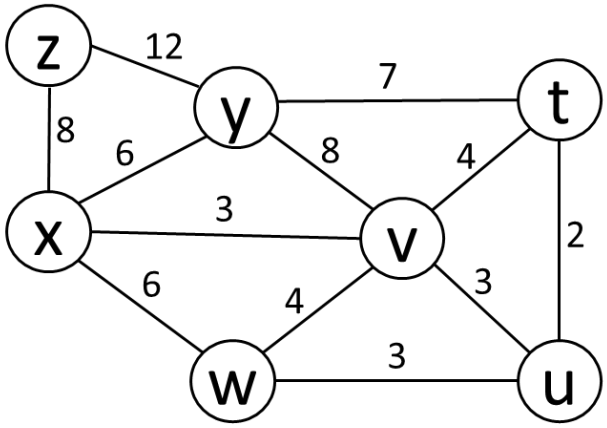
Limitation:

- Processing time < 1 second
- $2 \le n \le 10^5, 0 \le m \le 10^5$
- $1 \le a_i, b_i \le n, 1 \le w_i \le 10^6$

Notice:

- Using C, C++, Java, or Python
- The name of the source code file is **shortestpath.xxx** (The extension **xxx** depends on the used programming language)

Problem #2 (10 points). Run Dijkstra's algorithm on the following graph, use x as the source.

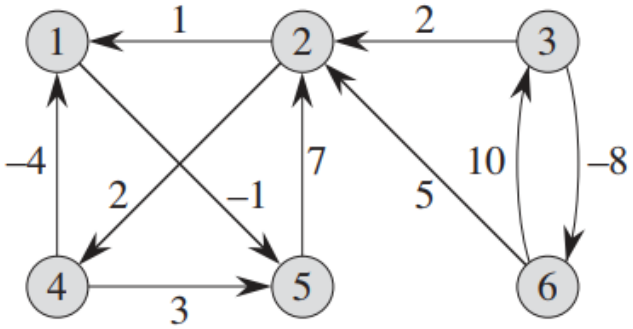


Problem #3 (15 points).

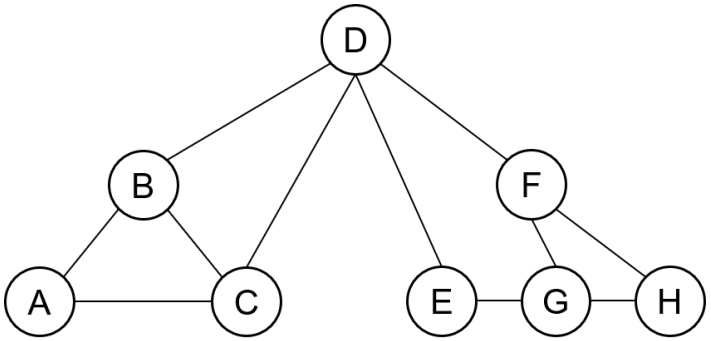
We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Write pseudocode of an algorithm which finds the most reliable path between two given vertices.

Problem #4 (15 points).

Run the Floyd-Warshall algorithm on the following graph. Show matrix $D^{(k)}$ after each iteration.

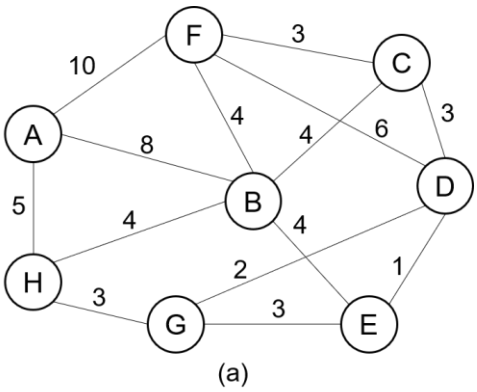


Problem #5. (10 points) Given the following graph. For the following questions, assume that the graph is represented using adjacency lists, and that all adjacency lists are sorted, i.e., the vertices in an adjacency list are always **sorted alphabetically**.

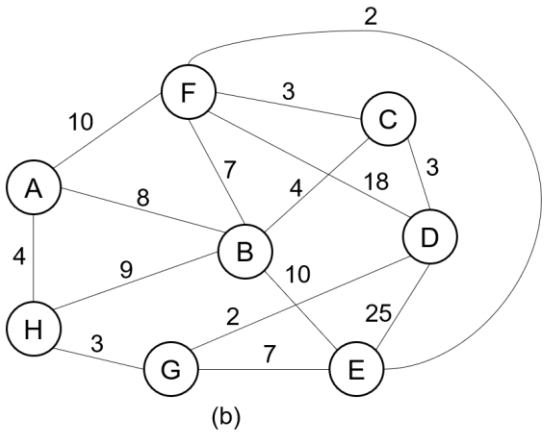


a) Suppose that you want to find a path from A to H. If you use breadth-first search, write down the resulting path as a sequence of vertices.
b) If you use depth-first search to find a path from A to H, write down the resulting path as a sequence of vertices.

Problem #6. (10 points) Given the following graphs.



(a)



(b)

a) Use Kruskal's algorithm to find the minimum spanning tree in graph (a). Draw the final tree with weights of all edges. What is the total weight of the final tree?
b) Use Prim's algorithm to find the minimum spanning tree in graph (b) with starting node D. Draw the final tree with weights of all edges. What is the total weight of the final tree?

What you have to submit:

1) Your source programs
2) Your input and output files
3) Documentation file (**HW5.DOCX**)
   - Solution of the assigned problems
   - Write the explanation about your implementation

◆ Submit your compressed file named as HW5_ID_NAME.zip (ex. HW5_2013711123_홍길동.zip) to iCampus.

NOTICE:

- ✓ BOTH ORIGINAL AND COPY WILL GET –30 POINTS EACH INSTEAD OF 0S.
- ✓ ANY SOURCE CODE WITH COMPILE OR RUNTIME ERROR WILL GIVE YOU 0 POINTS.
- ✓ THERE WILL BE POINTS OFF FOR INAPPROPRIATE SUBMISSION STYLE.
- ✓ ALL THE HOMEWORK MATERIALS (INCLUDING EMAIL CONTENTS AND DOCUMENTATION) SHOULD BE MADE IN **ENGLISH**.

Good luck!