

Project 2: Graph Algorithms

Singles-source shortest path and Minimum Spanning Tree (MST)

In this project, you will implement two graph algorithms mentioned below.

Note: You can work **alone or in a team of two max**. If you want to work in a team but have no project partner, let me know by next week.

Problem 1:

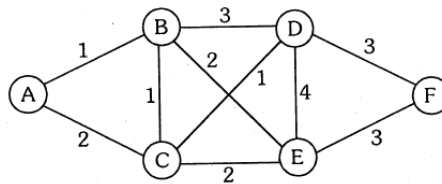
Find shortest paths in both directed and undirected weighted graphs for a given source vertex. Assume there is no negative edge in your graph. You will print each path and path cost for a given source.

Problem 2:

Given a connected, undirected, weighted graph, find a spanning tree using edges that minimizes the total weight $w(T) = \sum_{(u,v) \in T} w(u,v)$. Use Kruskal or Prim's algorithm to find Minimum Spanning Tree (MST). You will print out the edges of the tree and the total cost of your answer.

Input format:

For each problem, you will take input from a text file. Say you want to run algorithm on the following undirected graph. The corresponding file format would be:



```

6      10      U
A      B      1
A      C      2
B      C      1
B      D      3
B      E      2
C      D      1
C      E      2
D      E      4
D      F      3
E      F      3
A
```

Here, the first two numbers represent the number of vertices and edges. The letter U stands for undirected graph (D for directed). From the second line list all edges and its weight (e.g. $edge(A,B)$ and its weight is 1. The last line is optional. If given, it represents the source node.

Submission instructions:

- A well-formatted report covering short description of each algorithm, data structure used, runtime of your code, sample input/output, instruction to run your program easily.
- For each problem, run your program for three different graphs of your choice. Graph should be big enough. Use your judgement to define test graphs that you think interesting and reasonable. For example:
 - Undirected graph: at least 6 nodes and 10 edges

- Directed graph: at least 8 nodes and 15 edges
- Clean code for TA to execute.
- You can use any programming language (e.g. C/C++, Java, Python, etc.)
- If worked in a team, still both members are required to submit everything separately.
- **Hardcopy** of your report to me directly; one copy per team.

Grading scheme:

Shortest Path in directed and undirected Graphs	40
Minimum Spanning Tree	30
Input taken from file properly	5
Proper output display	5
Report according to the description	20