

# Coding + Theory Assignment 4: CS2233

3th November, 2025

**Max Marks 130**

---

**Instructions:** Kindly adhere to the following instructions.

- Please write a C/C++ program corresponding to each problem. Your code should be well commented on, and variable names should be appropriately chosen.
  - Create a folder and put all the code files and name all the files as the question number (i.e. 1.cpp/1.c), add the scanned copy of the theory assignment solution to it, give a name to the folder as “yourRollNo”, zip the folder and submit it to the Google Classroom portal. Also, submit the hard copy of the theory assignment to the class.
  - Strictly follow the input and output format for each problem.
  - Any code that does not follow the input-output criteria won't be evaluated and will get **ZERO**.
  - Your code will also be checked against plagiarism (both from web and peer).
  - Any form of plagiarism (web/chatGPT/with peers) will be severely penalised and will result in an F grade.
  - For theory questions, the answer should consist of the following three parts: (i) algorithm/pseudocode of the problem, (ii) a theorem statement stating that the algorithm mentioned outputs the desired result, and a comment about the time/space complexity of the algorithm, (iii) a formal proof of the theorem.
  - The submission (strict) timeline is 17th November, Monday, 11:30 PM.
-

# 1 Coding Problems:

In the following problems, suppose the graph is given as input in the adjacency list format.

## 1. Graph Traversal:

- (a) Suppose  $G = (V, E)$  is an undirected graph, and a vertex  $s \in V$  is given as input. Implement the **non-recursive DFS traversal algorithm**. For every vertex  $v \in V$ , you need to output both the **length of the path** from  $s$ , and the **list of vertices** in the path.

[10 Marks]

### Input Format:

- The first line contains an integer  $n$  denoting the number of vertices.
- The second line contains an integer  $m$  denoting the number of edges.
- The next  $m$  lines contain two integers  $u$  and  $v$  representing an undirected edge between  $u$  and  $v$ .
- The last line contains an integer  $s$  denoting the source vertex.

### Output Format:

- For each vertex  $v \in V$ , print:
  - The length of the path from  $s$  to  $v$ .
  - The list of vertices in the path from  $s$  to  $v$ .

### Sample Input:

```
6
7
1 2
1 3
2 4
2 5
3 6
4 5
5 6
1
```

### Sample Output:

```
Vertex 1: Length = 0, Path = 1
Vertex 2: Length = 1, Path = 1 2
Vertex 3: Length = 1, Path = 1 3
Vertex 4: Length = 2, Path = 1 2 4
Vertex 5: Length = 2, Path = 1 2 5
Vertex 6: Length = 2, Path = 1 3 6
```

## 2. Minimum Spanning Tree (MST):

- (a) Suppose  $G = (V, E)$  is an undirected weighted graph given as input. Compute the MST of this graph using **Prim's algorithm**. You need to output the list of edges that comprise the MST. [10 Marks]
- (b) Suppose  $G = (V, E)$  an undirected weighted graph is given as input. Compute the MST of this graph using **Kruskal's algorithm**. You need to output the list of edges that comprises MST. [10 Marks]

### Input Format:

- The first line contains an integer  $n$  denoting the number of vertices.
- The second line contains an integer  $m$  denoting the number of edges.
- The next  $m$  lines contain three integers  $u, v, w$  representing an undirected edge between  $u$  and  $v$  with weight  $w$ .

### Output Format:

- Output the list of edges included in the MST in any order.
- Also print the total weight of the MST.

### Sample Input:

```
5
7
1 2 2
1 3 3
2 3 1
2 4 4
3 5 5
4 5 7
3 4 6
```

### Sample Output:

```
Edges in MST:
(2, 3) weight = 1
(1, 2) weight = 2
(2, 4) weight = 4
(3, 5) weight = 5
Total Weight = 12
```

## 3. Single Source Shortest Path:

- (a) Suppose  $G = (V, E)$  is a directed weighted graph with non-negative edge weights, and a vertex  $s \in V$  is given as input. Implement **Dijkstra's algorithm** for computing the shortest path from  $s$  to every vertex  $v \in V \setminus \{s\}$ . For each  $v$ , output both the **length of the shortest path** from  $s$  to  $v$ , and the **list of vertices** in the path.

[10 Marks]

**Input Format:**

- The first line contains an integer  $n$  denoting the number of vertices.
- The second line contains an integer  $m$  denoting the number of edges.
- The next  $m$  lines contain three integers  $u, v, w$  representing a directed edge from  $u$  to  $v$  with weight  $w$ .
- The last line contains an integer  $s$  denoting the source vertex.

**Output Format:**

- For each vertex  $v \in V$ , print:
  - The shortest distance from  $s$  to  $v$ .
  - The path from  $s$  to  $v$ .

**Sample Input:**

```
6
9
1 2 4
1 3 2
2 3 5
2 4 10
3 5 3
5 4 4
4 6 11
5 6 5
3 6 8
1
```

**Sample Output:**

```
Vertex 1: Length = 0, Path = 1
Vertex 2: Length = 4, Path = 1 2
Vertex 3: Length = 2, Path = 1 3
Vertex 4: Length = 9, Path = 1 3 5 4
Vertex 5: Length = 5, Path = 1 3 5
Vertex 6: Length = 10, Path = 1 3 5 6
```

## 2 Theory Questions:

1. Suppose that a weighted, directed graph  $G = (V, E)$  has a negative-weight cycle. Give an efficient algorithm to list the vertices of one such cycle. Prove that your algorithm is correct. [10 Marks]
2. Given a graph  $G$  and a minimum spanning tree  $T$ , suppose that we decrease the weight of one of the edges not in  $T$ . Give an algorithm for finding the minimum spanning tree in the modified graph. [10 Marks]

3. Show that a graph has a unique minimum spanning tree if, for every cut of the graph, there is a unique light edge crossing the cut. Show that the converse is not true by giving a counterexample. [10 Marks]
4. Suppose that a graph  $G$  has a minimum spanning tree already computed. How quickly can we update the minimum spanning tree if we add a new vertex and incident edges to  $G$ ? [10 Marks]
5. Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers. Please elaborate on what points the proof of Dijkstra's algorithm breaks when negative-weight edges are allowed? [10 Marks]
6. Let  $G = (V, E)$  be a weighted, directed graph with nonnegative weight function  $w : E \mapsto \{0, 1, 2, \dots, W\}$  for some nonnegative integer  $W$ . Modify Dijkstra's algorithm to compute the shortest paths from a given source vertex  $s$  in  $O(VW + E)$  time. [10 Marks]
7. Consider an array  $A$  over  $n$  distinct elements. Prove that the best case running time for running HEAPSORT on  $A$  is  $\Omega(n \ln n)$ . [10 Marks]
8. Consider a directed graph  $G = (V, E)$  with no self loops. Consider the **incident Matrix** of the graph  $G$ , a matrix of size  $|V| \times |E|$  denoted as  $B$ , whose each entry  $b_{ij}$  is defined as follows:

$$b_{ij} = \begin{cases} -1 & \text{1 if edge } j \text{ leaves vertex } i \\ 1 & \text{1 if edge } j \text{ enters vertex } i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Describe what the entries of the matrix product  $BB^T$  represent, where  $B^T$  is the transpose of  $B$ . [10 Marks]

9. The square of a directed graph  $G = (V, E)$  is the graph  $G^2 = (V, E^2)$  such that an edge  $(u, v) \in E^2$  if and only if  $G$  contains a path with at most two edges between  $u$  and  $v$ . Describe efficient algorithms for computing  $G^2$  from  $G$  for both the adjacency-list and adjacency-matrix representations of  $G$ . Analyze the running times of your algorithms.

[10 Marks]