Assignment 5 - Graphs

April 10, 2024

1 Instructions

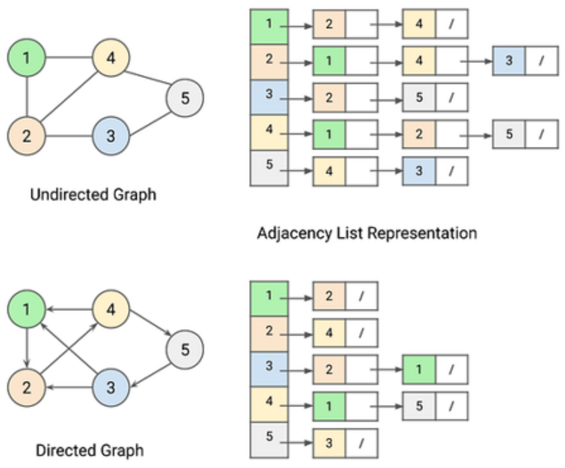
- You can code in either C or C++.
- The assignment needs to be submitted on OJ. This assignment comprises of two questions.
- The deadline for this assignment is 23^{rd} April. Extension requests will not be entertained, so please start early.
- Resorting to any sort of plagiarism (as mentioned in the policy shared) will lead to strict penalisation at the least

2 Introduction

A graph data structure is a collection of nodes connected by edges. A graph can be implemented in variety of ways, the most common are adjacency matrix and adjacency list. In this assignment, you will be required to implement the graph using an adjacency list.

2.1 Adjacency List

An adjacency list stores the neighbours of a node in the form of a linked list. An adjacency list for a directed and undirected graph is shown below.



As you can see in the above figure, in the case of **undirected graph**, if node A is connected to node B then node B is also connected to node A. So while adding an edge between A and B, you have to add B to the neighbours list of A and A to the neighbours list of B. For example, in the above case there is an edge between node 1 and 4. So, in the adjacency list, 4 is present in the neighbours list of 1 and also 1 is present in the neighbours list of 4.

But in the case of a **directed graph** each edge has a direction. If node A is connected to node B does not mean node B is connected to node A. So while adding an edge between A and B, you have to take care of the direction as well.

For example, in the above case in case of directed graph, there is an edge pointed from 4 to 1. This means that 1 is a neighbour of 4 but 4 is not a neighbour of 1. So you have to add only 1 to the neighbours list of 4.
If we want to access all the edges corresponding to node 1 then we have to traverse through the neighbours list corresponding to node 1. So, in the undirected graph, the neighbours of 1 are 2 and 4. So there is edge between 1-2 and 1-4.

3 Question 1: IIIT Hyderabad - Vindhya Maze Expansion Project

3.1 Introduction

The sprawling IIIT Hyderabad campus is known for its iconic Vindhya buildings, arranged in a maze-like fashion. To enhance the campus's unique character and encourage exploration, we're embarking on the "Vindhya Maze Expansion Project"! As a bright student of IIIT Hyderabad, you're tasked with helping us plan this exciting project. There are many Vindhya buildings and there exists a maze of pathways interconnecting them. However, they need not be completely interconnected. We want to create a network of new pathways so that anyone can travel from any building to any other building within the Vindhya complex.

3.2 Problem Statement

3.2.1 The Problem

You're provided with information about the current state of the Vindhya complex:

- The total number of buildings - n
- The number of existing pathways connecting these buildings - m

Each existing pathway connects two distinct buildings bidirectionally, and there can be at most one pathway between any two buildings.
Your objective is to determine the minimum number of new pathways needed to ensure anyone can travel between any two buildings in the Vindhya complex. Additionally, you need to specify which specific new pathways should be constructed.

3.2.2 Input Format

The first line of the input file will contain two space-separated integers:

- n: The total number of buildings in the Vindhya complex
- m: The number of existing pathways connecting these buildings

The following m lines will describe the existing pathways:

- Each line will contain two space-separated integers a and b, representing an existing pathway connecting building a and building b ($1 \leq a, b \leq n$).

3.2.3 Output Format

On the first line, print an integer k, representing the minimum number of new pathways needed to connect all buildings.

Following that, print k lines, each specifying a new pathway to be constructed. Each line should contain two space-separated integers, representing the two buildings that the new pathway will connect. You can print any valid solution (i.e., any set of new pathways that achieves full connectivity).

3.2.4 Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$

3.2.5 Example 1

Input:
4 2
1 2
3 4
Output:
1
2 3
Explanation:
In this example, there are four buildings and two existing pathways, as illustrated below on the left. The output indicates that only one new pathway needs to be constructed, connecting building 2 and building 3, as shown below on the right. This will ensure that anyone can travel between any two buildings in the Vindhya complex.

Figure 1: Illustration of buildings and pathways. Left: Initial configuration. Right: Final configuration.

3.2.6 Example 2

Input:

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

Output:

```
2
1 2
6 7
```

Explanation:

In this example, there are seven buildings and five existing pathways, as illustrated below on the left. The output indicates that two new pathways need to be constructed, connecting building 1 and building 2, building 6 and building 7; as shown below on the right. This will ensure that anyone can travel between any two buildings in the Vindhya complex.

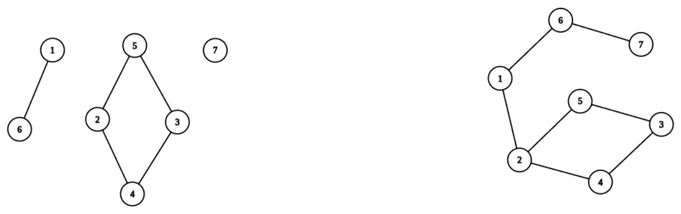


Figure 2: Illustration of buildings and pathways. Left: Initial configuration. Right: Final configuration.

4 Question 2: Need For Speed

4.1 Introduction

Lukarp has started his own tech company. He received a lot of funding from Igen with which he opened many offices around the world. Each office needs to communicate with one other, for which they're using high speed connections between the offices. Office number 1 is Lukarp's HQ. Some offices are important and hence need faster connections to the HQ for which Lukarp has use special fiber connections. Lukarp has already planned the connections but feels some fiber connections are redundant. You have been hired by Lukarp to remove those fiber connections which don't cause faster connections.

4.2 Problem Statement

4.2.1 The Problem

The offices and (bi-directional) connections (both normal and fiber) are given to you. **HQ is numbered as 1**. The i^{th} normal connection connects any two offices a_i and b_i. Normal connections have latency l_i. The i^{th} fiber connection connects the HQ with the office c_i. Fiber connections also come with a latency p_i. The total latency of a path is the sum of latencies on the connections. You are to output the **maximum number of fiber connections** that can be removed, such that the **latency of the smallest latency path** between the HQ and any other node remains the same as before.

- There are n offices with m normal connections and k high-speed fiber connections.
- The i^{th} normal connection connects offices a_i and b_i (bi-directionally) with latency l_i.
- The i^{th} fiber connection connects offices 1 and c_i (bi-directionally) with latency p_i.

4.2.2 Input Format

The first line of the input file will contain three space-separated integers n, m and k, the number of offices, the number of normal connections and the number of fiber connections.
There will be m lines after this, the i^{th} line signifying the i^{th} normal connection, each containing three space-separated integers a_i, b_i and l_i the two offices that are connected and the latency of the connection respectively.
There will be k lines after this, the i^{th} line signifying the i^{th} fiber connection, each containing three space-separated integers c_i and p_i, the office connected to the HQ and the latency of the fiber connection respectively.

4.2.3 Output Format

Output only one integer m - the **maximum number of fiber** connections that can be removed without changing the latency of **smallest latency path** from office 1 to any other office.

4.2.4 Constraints

- $2 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$
- $1 \leq k \leq 10^5$
- $1 \leq a_i, b_i, c_i \leq n$
- $1 \leq l_i, p_i \leq 10^9$

4.2.5 Example

Input:

```
4 5 2
1 2 2
1 4 9
1 3 3
2 4 4
3 4 5
3 4
4 5
```

Output:

```
1
```

Explanation:

In this example, there are five normal connections as shown in the figure below. The fiber connection going from 1 to 3 can be removed because the normal connection (3) is faster than the fiber connection (4). However, the fiber connection with 4 cannot be removed. Hence the maximum number of fiber connections that can be removed is 1.

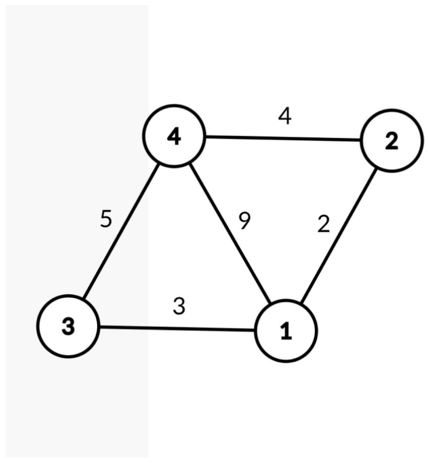


Figure 3: Normal Connections and their latencies