Coding Assignment 5
ESO207 2024-25-I

September 29, 2024

1 Introduction

In this homework, you will write a C program to implement Disjoint Set Union data structure and finding Minimum Spanning Tree using Kruskal's Algorithm.

The full marks will be 60. The deadline of this homework is Saturday, October 5, 11 PM IST. Your submissions will not be accepted after this time. Submit a single C file titled 'hw.c' to the Gradescope active assignment titled 'DSU'.

2 Problem Statement

There will be 2 types of problems, the first line of each testcase will contain an integer 0 or 1 denoting its type, please carefully note the differences in input format for both types of problems.

2.1 Disjoint Set Union

You are given n items (represented by integers 1 to n) where each item is initially in its own set.
Then q queries are given, where each query can be of 2 types:

- merge(i,j): It means to combine (take union) of the sets containing items i and j, if i and j belong to the same set then nothing is required. This query will not generate any output.
- report(i,j): You need to report whether the two items i and j belong to the same set or not.

2.2 Minimum Spanning Tree

You are given a weighted undirected graph with n nodes. Your task is to compute the minimum possible weight of any spanning tree. You can use the Disjoint set union data structure from Part 1 to implement **Kruskal's algorithm**.

Input

The first line contains an integer indicating a flag (0 or 1) indicating DSU and MST problems respectively. The input after the first line depends on flag as follows.

1 **flag = 0 (DSU problem)**

   - The second line contains two integers n and q representing the number of items and number of queries.
   - Next q lines contain 3 space-separated integers *query*, i and j, where query can be 0 or 1 and denotes merge and report queries. i, j are from 1 to n.

2 **flag = 1 (MST problem)**

  - The second line contains an integer n representing the number of nodes.
  - Next n lines each will have the weighted adjacency list for the nodes. The list for i-th node will have each child j of i followed by the weight of the edge (i, j) separated by a space. Each adjacency list will end with a -1. Thus if there are no children, the list will only have a -1.

Output

- If the flag is 0, output 0 or 1 for each report query (query type 1) indicating whether the corresponding items i and j are in same set (1 if they are in same set, 0 otherwise). All outputs will come in a single space separated line.
- If the flag is 1, output a single line containing the minimum weight of MST.

Consider the sample inputs and outputs given below:
Example 1:

```
0
6 5
0 3 5
1 1 3
0 1 3
1 1 5
1 1 2
```

So, for the above input, the output will be (corresponding to query 2, 4 and 5:

```
0 1 0
```

Example 2:

```
1
5
2 2 3 3 5 5 −1
1 2 4 2 −1
1 3 5 1 −1
2 2 −1
1 5 3 1 −1
```

So, for the above input, the output will be:

```
8
```

Note that the output ends with a space(there is no newline after).

3 Test Cases

The above test cases will carry 20 marks. There will be 8 hidden test cases each of 5 marks. Thus the total will be 60.

4 Submission Instructions

- You must submit a C program titled 'hw.c'. Other programming languages such as C++, Python, etc. are not allowed. Your code must take the input from "stdin" and write the output to "stdout".
- Your code will be automatically graded in Gradescope on some test cases as above which will be hidden from you. Therefore, you must make sure that you understand and precisely follow the expected input-output behavior.
- Please write a single C code and name it as 'hw.c'. This is extremely important. If you violate this, your code will not pass the automatic test cases even if your code runs correctly on your local machine. Common examples of failures include:
  - if you write a C++ program that has the correct input-output behavior
  - if you write two or more different C codes or .h header codes and link them
  - or write a single correct code but name it as 'test.c'

  In any of the above cases, your code will fail. Thus, while you are perfectly allowed to develop your code in your local machine and it works correctly, your code may run into problems in Gradescope until and unless you follow the above instructions.
- Submit your code on *Gradescope* active assignment titled 'DSU'. Otherwise your code will not be graded. In particular, do NOT submit on hello IITK or over email. Email to us (instructor or the TAs), meet us, or start a discussion in helloIITK if you run into any issues.