

Hand in your solutions electronically using LearnUs.

This assignment is a completion points assignment: any reasonable attempt will receive *full* points, *even if it is incorrect*.

This is a programming assignment. Submit your source code(s), zipped as ***yourStudentID.zip***. For example, if your student ID is 2023000000, then you must zip all your source code(s) into ***2023000000.zip*** and submit this file. Each class should have its own **.java** file, of which the filename is the same as the class name. Do *not* include your student ID as part of the class names. You are not allowed to use any data structure libraries, including the linked lists provided by JDK.

Try to solve this problem without referring to the lecture notes. The problem description starts on the next page.

(1) (10 points) Write a program that finds a minimum spanning tree of a given weighted graph. Your program must internally use adjacency lists to represent graphs.

Your program must read its input from `input.txt` in the current working directory. The first line of the input file contains the number of vertices n and the number of edges m , separated by a space. Each vertex is numbered from 0 to $n - 1$. Each of the following m lines of the input file contains (the numeric IDs of) the two endpoints and the weight of each edge, in that order, separated by spaces. The input therefore consists of $m + 1$ lines in total.

Your program must output the result to `output.txt` in the current working directory. If the input graph does not have a minimum spanning tree, the output file contains a single line containing `none`. Otherwise, the output file consists of $n - 1$ lines, each of which contains (the numeric IDs of) the two endpoints of each edge in the minimum spanning tree found by the program.

The entry point of your program must be `CPA3.main()`.

Example 1

`input.txt`

```
4 5
0 1 -2
1 2 5
2 3 10
3 0 7
0 2 0
```

`output.txt`

```
0 1
3 0
0 2
```

Example 2

input.txt

```
4 3
0 1 10
1 2 -5
2 0 7
```

output.txt

none