# Assignment overview

Today we will help the little-known country of Solozol to build a new and inexpensive highway system.

Please answer the questions below and submit this Word document back to Learning Hub by the due date.

*There is NO CODING REQUIRED for this lab.*

# Submission information

**Due date**: As shown on Learning Hub. Late assignments will not be graded.

What to submit:

* This Word document (questions/answers below)

# Grading

This lab is worth 10 points.

# Questions

Study the programming assignment at the end of this handout and answer the questions below.

1. (1 point) What type of problem is this (i.e.: shortest path, topological sort, spanning tree, union find, etc.)?

Minimum Spanning Tree

1. (2 points) Which algorithm (that we have studied) can be used (with enhancements) to solve this problem? If more than one algorithm can be used, explain why you selected the one that you did.

Kuruskal / Prim algorithm can be used to solve this problem.

I would use Prim algorithm because it does not require to do pre-sort.

1. (2 points) Explain how you can model the problem as a graph. What is represented the vertices and by the edges? Draw the graph that corresponds to the sample input that is provided with the problem.

A picture containing text, dog, map

Description automatically generated

Vertices = number of town

Edges = highways

the weight of the edge = Cartesian distance

1. (3 points) Describe any modifications/enhancements/changes required to make the algorithm you selected work for this problem. Describe your enhancements in plain English. Pseudocode is NOT required, but you may include some if you feel it will help explain what you are doing.

1. Calculate the Cartesian distance between towns

2. Add Edges(towns) to the minimum spanning tree

1. (2 points) Apply your algorithm to the sample input provided with the problem. Show all your work so the instructor is convinced that your algorithm works and that you know what you are doing.

1. Connect all the towns and calculate the Cartesian distances

2. Add the highways already built to the graph

3. Apply Prim's algorithm

A close up of a map

Description automatically generated

# Solozol Highway Problem

(You do not need to write code for this problem. Just study the problem and answer the questions above in the handout.)

The independent nation of Solozol is perfectly flat. Unfortunately, Solozol has a very poor system of public highways. The Solozolian government is aware of this problem and has already constructed a number of highways connecting some of the most important towns. However, there are still some towns that you can't reach via a highway. It is necessary to build more highways so that it will be possible to drive between any pair of towns without leaving the highway system.

Solozolian towns are numbered from 1 to N. Town number i has a position given by the Cartesian coordinates (xi, yi). Each highway connects exactly two towns. All highways (both the existing ones and the ones that are to be built) follow straight lines, and thus their length is equal to Cartesian distance between towns. All highways can be used in both directions.

Highways can freely cross each other, but a driver can only switch between highways at a town that is located at the end of both highways. The Solozolian government wants to minimize the cost of building new highways. However, they want to guarantee that every town is highway-reachable from every other town. Since Solozol is so flat, the cost of a highway is always proportional to its length. Thus, the least expensive highway system will be the one that minimizes the total length of all highways.

### Input

The input consists of two parts. The first part lists all the towns in Solozol, and the second part describes all of the highways that have already been built.

The first line of the input contains a single integer N (1 ≤ N ≤ 750), representing the number of towns. The next N lines each contain two integers, xi and yi, separated by a space. These values give the coordinates of town number i (for i from 1 to N). Coordinates will have an absolute value no greater than 10000. Every town has a unique location.

The next line contains a single integer M (0 ≤ M ≤ 1000), representing the number of existing highways. The next M lines each contain a pair of integers separated by a space. These two integers give a pair of town numbers which are already connected by a highway. Each pair of towns is connected by at most one highway.

### Output

Write to the output file a single line for each new highway that should be built in order to connect all towns with minimum total length of new highways. Each highway should be presented by printing the town numbers that this highway connects, separated by a space.

If no new highways need to be built (all towns are already connected), then the output file should contain a line with the sentence "No new highways needed".

### Sample input

9

1 5

0 0

3 2

4 5

5 1

0 4

5 2

1 2

5 3

3

1 3

9 7

1 2

### Sample output for the sample input

1 6

3 7

4 9

5 7

8 3