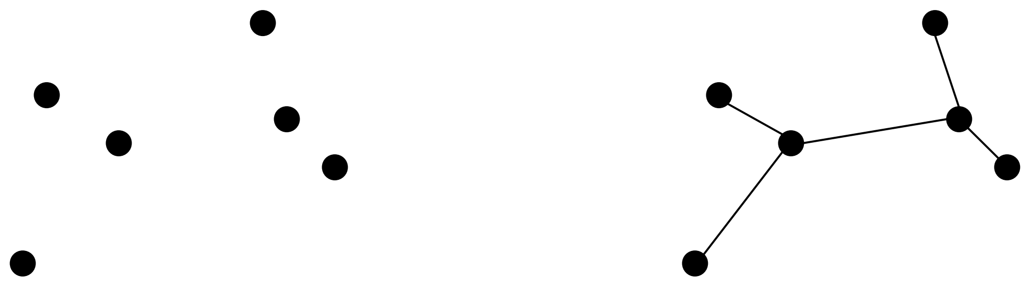
Names and Student IDs (2-4): SAMPLE SOLUTION

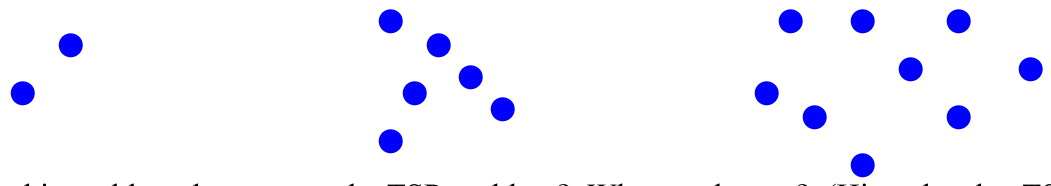
CPSC 101/WMST 201 EXERCISE: Algorithms

You are hired by a company that has invented teleportation. Each teleporter connects **one** location to **one** other location. The cost of a teleporter is proportional to the distance between the two ends of the connection. So, when you build a teleport network, you want to connect all cities (so there's a route to teleport along between any pair of cities), using the smallest, cheapest collection of teleporters. For example, the network on the right below is the cheapest one for the six cities on the left.



(a) Is this problem specifically about the diagram above? If not, sketch at least three other very different instances of the same problem.

No, the diagram above is just one example of an instance of this problem. Other examples might have **more or fewer cities** and **different locations for cities**. Here are three examples of other problem instances:



(b) Is this problem the same as the TSP problem? Why or why not? (Hint: sketch a TSP solution on the figure on the left. Is it the same as what the problem explicitly states is the optimal solution on the right? Why or why not?)

No, this is not the same as TSP. Although the inputs are very similar (but for the lack of a "start city") and the outputs are also somewhat similar (both a list of edges, in some sense), the constraints are radically different. This problem does not require the edges to form a single cycle, nor does it require any "return path" back to a start city (indeed, any extra path at all would be a waste of an edge in this problem!).

(c) What is the difference between an *algorithm* and *characterizing a problem algorithmically* (i.e., clearly describing the inputs of the problem, the outputs of the problem, and the constraints on the inputs, outputs, and their relationship)?

Characterizing a problem algorithmically means clearly defining the inputs, outputs, and constraints (on the inputs and the outputs and their relationship) that together define what it means to solve the problem. An algorithm, on the other hand, is a list of steps that, when executed, actually solves the problem!

(d) Characterize this problem algorithmically.

INPUTS:
A list of city locations. (Note: it wouldn't be correct to say the inputs are: "costs in $ between cities and time between cities and distance between cities". These are three different examples of how we might measure cost between pairs of cities, but for any one problem, we'd ONLY list one of these!)

OUTPUTS:
A list of pairs of cities, i.e., a list of teleporter links that we will make.

CONSTRAINTS:
(1) The cities used in the output are drawn from the cities given in the input.
(2) Every city in the input is reachable from every other city in the input by following some series of links listed in the output (i.e., the output is a connected point-to-point network of the input cities).
(3) (Ideally) The total of the distances between each pair of cities listed in the output is as small as it can be and still conform to the other constraints (i.e., we found the "best" solution).

(e) Propose a way to solve this problem **by transforming inputs to this problem into inputs to a TSP problem and transforming the TSP solution back into a solution for this problem**. Hint: be careful not to build more links than you need to build!

Note: you should **not** propose a solution to TSP. You need only say how you would transform input to this problem into input to TSP and how you would transform output of TSP to output of this problem (as with DNA sequencing and puzzles!).

As mentioned above, this problem and TSP are already quite similar. We need only:

INPUT: keep the list of locations as is and select some start city (arbitrarily)

OUTPUT: transform the list of cities that is the tour (c1, c2, c3, …, cn, c1) into a list of edges (c1-c2, c2-c3, …, cn-c1).

As the hint notes, we end up with one more edge than needed. So, eliminate one edge. To get as good a solution as we can, eliminate the longest edge.

(f) Explain why using TSP as in (e) would not usually produce the best solution to this problem.

TSP constrains its solution to have exactly two edges connecting to each city, but that will often (as in the example problem!) not be the best solution.