Programming Assignment 1: Kruskal’s Algorithm

Problem Statement

The purpose of this assignment is to give you practical experience implementing graphs and finding minimum-weight spanning trees. In this assignment, you will take the role of an ISP bringing fiber optics to a new neighborhood. You want to minimize the cost of installation. The neighborhood can be modeled as a graph with buildings as vertices and possible connections between these as weighted edges based on the cost of installing that connection. Assume that fiber can only be laid along roads. For simplicity, we will assume the length of fiber used is the only cost, so the cost of connecting two homes is the Manhattan distance* between them. Also assume that every home can potentially be connected. Use Kruskal’s algorithm to find an installation plan that connections all buildings in a neighborhood with minimal cost.

Program Specifics

• Your program should read the graph from standard input in the following format:
  <# of buildings>
  <1st coordinate> //ex: 2,10
  <2nd coordinate>
  ...
  <Last coordinate>
• Your program must use Kruskal’s algorithm to compute the optimal network structure.
• Your program should print the following to standard output:
  <1st edge>, <weight> //ex: (1,1)--(4,5),5
  <2nd edge>, <weight>
  ...
  <last edge>, <weight>
  Total Cost: <Total cost>
• See program1_ex.in and program1_ex.out for example input and output.
• Don’t cheat! Many languages have pre-built graphs and graph algorithms. You need to build these yourself!
• All code must be written in either Java or Python.

Due Monday, April 22

---

* The Manhattan distance between two points is just the sum of the x-distance and the y-distance. For points \( p_1 = (x_1, y_1) \), \( p_2 = (x_2, y_2) \), the Manhattan distance \( (M) \) between the two would be \( M(p_1, p_2) = |x_2 - x_1| + |y_2 - y_1| \).