Programming Assignment 2: Prim’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 Prim’s algorithm to find an installation plan that connects all buildings in a neighborhood with minimal cost.

Program Specifics

• Your program should perform I/O in the same fashion as in Assignment 1. See program2_ex.in and program2_ex.out for example input and output.
  o Note: My solution is unlikely to be unique. Often, multiple MSTs exist for a given graph. Additionally, the choice of a starting vertex and how your priority queue breaks ties will definitely effect the order in which you add edges to your MST.
• Your program must use Prim’s algorithm to compute the optimal network structure.
• Don’t cheat! Many languages have packages with pre-built graphs and graph algorithms. You need to build these yourself!
  o Exception: you don’t have to build your own priority queue. However, most pre-build priority queues lack a delete and decreaseKey method, which means you may have to get clever…
• All code must be written in Java, Python or C/C++. If you use Python, please specify which version you used (i.e. 2.X or 3.X) when you turn it in.

Due Wednesday, May 1 @ 11:59 PM

* 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 \( M(p_1, p_2) = |x_2 - x_1| + |y_2 - y_1| \).