Evan Edelstein

1. Pseudocode:

Function in\_out\_degree(G,):

in\_degree = zero initialized array of length G.adj

out\_degree = zero initialized array of length G.adj

For i in G.adj:

d = 0

For j in G.adj[i]:

d++

in\_degree[j]

out\_degree[i] = d

for i in G.adj:

print(“node: ”, i)

print(“in-degree: “, in\_degree[i])

print(“out-degree: out\_degree[i])

* 1. Since the pipe is straight, we only need to consider the difference in longitudinal value between the wells. In this way we can think of all the wells as being on single vertical line and reframe the problem as searching for a point on this line that has the minimum distance to all the points. We can use the median of all the wells y axis value (or height on the imaginary line) to determine the horizonal line that minimizes the north/south pipeline length. This is because, by definition, the median is the point that is closest to all the other points.
  2. The Select algorithm from the textbook can be used to find the median in O(n) time. For an array with an odd number of wells (n), we would select for the (n+1)/2 order statistic and for even we select either the (n/2) or (n+1)/2 order statistic.