Michael Nugent mpn1@pitt.edu

1 Prim's Minimal Spanning Tree (on a fully connected Euclidean graph)

This is the simplified version of Prim's algorithm for when the input is a graph that is full connected and each vertex corresponds to a point in Euclidean space, and distances between vertices are the distances between the points. In other words, for two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ the distance from p_1 to p_2 is $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

```
prim(list of vertices L)
  Choose start vertex u from L
  for each vertex v of L do
    v.priority = d(v, u)
    v.parent = u
  end for
  MST = new List
  for i = 1...V − 1 do
    min = ∞
    minVertex = NULL
    for each vertex v of L do                          ▷ Find the lowest priority vertex
      if v.priority > 0 and v.priority < min then
        min = v.priority
        minVertex = v
      end if
    end for
    minVertex.priority = 0
    MST.add(edge(minVertex, minVertex.parent))         ▷ Update the MST
    for each vertex v of L do                          ▷ Update the priority queue
      if v.priority > d(v, minVertex) then
        v.priority = d(v, minVertex)
        v.parent = minVertex
      end if
    end for
  end for
```

This algorithm runs in time $O(V^2)$, since the outer loop is done $V - 1$ times and there are two inner loops which are both done V times. Note that using a better priority queue and make the runtime of the algorithm on arbitrary graphs $O(E \log V)$, but that on a fully connected graph this is $O(V^2 \log V)$ which is actually slightly worse than this algorithm.

2 Turning MST into TSP tour

This is easy. Simply do a DFS search starting at some vertex of the tree and as you visit each vertex, add it to the tour. This gives a tour that is at most twice as long as the optimal Travelling

Salesperson tour. Why? Consider any tour of all the vertices, and remove one edge from that tour. What remains is a spanning tree. We know then that the total edge length of the minimal spanning tree is less than that of the optimal TSP minus one edge. By doing the DFS, we walk around the outside edge of the tree, going over each edge twice, and so the total length of the DFS is at most twice the length of the MST, and thus at most twice the optimal TSP. But didn't we skip some nodes when walking, so it's not quite the same distance? Yes, but by the triangle inequality, we know that skipping nodes cannot make the total distance longer.