Exercise Sheet: Dijkstra

Data Structures and Algorithms (X_400614)

1. Execute Dijkstra's algorithm on the graph pictured in Figure 1, starting at the vertex A. In case of ties, the vertex with the lower letter is handled first.
   a. List the vertices in the order in which they are deleted from the priority queue and for each the shortest distance from A to the vertex.
   b. Draw the shortest path tree that results.

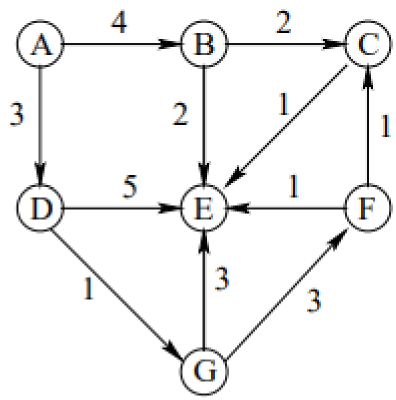

Figure 1.

2. Consider the graph in Figure 2.
   a. Execute Dijkstra's algorithm on this graph, starting from *s*. After each step, note down:
      i. The upper bounds *d[u]* (which denote what is currently believed to be the shortest distance to each node), for $u \in V$, between *s* and each node *u* computed so far,
      ii. the set *M* of all nodes for which the minimal distance has been correctly computed so far,
      iii. and the predecessor *p(u)* for each node in *M*.
   b. Replace the weight of edge (x,y) with -1 and redo the algorithm. Does the algorithm still compute correctly? If not, where does it break?
   c. Now, additionally change the weight of edge (v,y) to -10. Show that in this case the algorithm doesn't work correctly. That is, show that there is a vertex $u \in V$ such that *d[u]* is not equal to minimum distance from *s* to *u* after the execution of the algorithm.

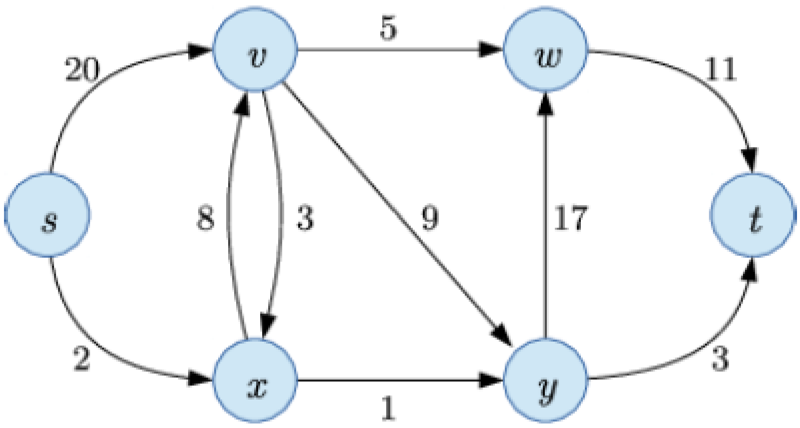

Figure 2.

3. A robot moves in the environment in Figure 3. It starts from the node labeled *Start* and needs to reach the node labeled *End*. The environment is continuous and the scale is supplied on the figure.
    a. Considering the robot as a point, what is the shortest path from *Start* to *End?*
    b. Can you find the same path faster using another algorithm? If so, which algorithm is better suited for this problem, and why?

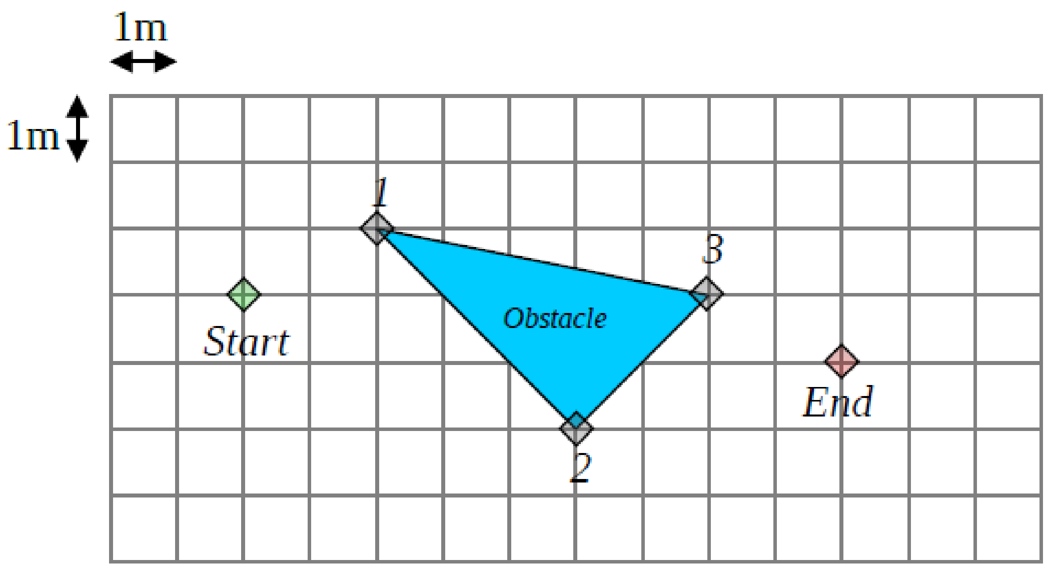

Figure 3.