CPSC320 Assignment 4

Evan Louie
72210099
m6d7

1.

a.

- Use Dijkstra’s algorithm as we normally would; starting at either of the vertices.
- Continue through the algorithm until the other vertex is considered ‘visited’ and removed from the unvisited set
- As the algorithm states; when a vertex has been removed from the ‘unvisited’ set, the shortest path to that vertex has been found.

b.

- If the graph is undirected:
    - Replace all undirected edges with a pair of directed edges (going both ways)
        - ie: {a,b} = (a,b) and (b,a)
- Assign a weight of 0 to all edges of the graph
- Replace all vertices v with a pair of vertices v_a and v_b.
- All edges going to v will now go to v_a and all edges going out of v will go out of v_b
- For all pairs v_a and v_b, create an edge between the two, the weight of which will be non-negative numbers originally assigned to the vertices.
- Execute Dijkstra’s algorithm starting at some arbitrary vertex n_a. As every path from some vertex i to some vertex j (in original graph) will have the same cost as the path from i_a to j_b (in modified graph); we can see that the cost of any given vertex v_b at the end of Dijkstra's algorithm is the shortest path length from s_i to v_b and hence the shortest path from s to v.

2.

a. **TRUE**

- Given node s of G, let e be an edge of s with the smallest weight
- Start Dijkstra's algorithm starting at s.
  - e will be the first edge considered 'visited' and added to the tree
- Start Kruskal's algorithm
  - Since e is an edge with the smallest weight joining to trees, it will be added to the minimum spanning tree
- Hence the tree of shortest paths and minimum spanning tree will share at least one edge

  QED

b. **TRUE**

- Execute Kruskal's algorithm on G, let G_i be the subgraph at i^{th} iteration of the while loop.
- When i=0
  - $G_i \cap H$ only contains nodes of H and no edges. Therefore $G_i \cap H \subseteq$ Minimum Spanning Tree of H
- Without loss of generality, we consider any i>0 and let e be the edge being evaluated at the i^{th} iteration of the while loop.
  - If e is an edge in H and Kruskal's algorithm doesn't add e to the minimum spanning tree, then $G_i = G_{i-1}$. Therefore, $G_i \cap H \subseteq$ Minimum Spanning Tree of H
  - if e is not an edge in H, $G_i \cap H = G_{i-1} \cap H$, therefore $G_i \cap H \subseteq$ Minimum Spanning Tree of H
  - If e is an edge in H and Kruskal's algorithm does add e to the minimum spanning tree, then e connects 2 trees which are not currently connected in G_{i-1}, therefore connecting 2 tree which are not currently connected in $G_{i-1} \cap H$. Because e is the minimum weighted edge which connects the 2 trees:
    - $(G_{i-1} \cap H) \cup \{e\} \subseteq$ Minimum Spanning Tree of H

  QED