COL351 Holi2023: Tutorial Problem Set 6

1. Recall the Prim's algorithm discussed in class. Prove that it correctly finds a minimum weight spanning tree of the edge weighted graph given to it as input. Also, show how to implement the algorithm using the appropriate data structures so that it runs in time $O(m \log n)$.

2. Recall the reverse delete algorithm discussed in class. Prove that it correctly finds a minimum weight spanning tree of the edge weighted graph given to it as input.

3. Consider the following algorithm on an edge-weighted connected graph $G = (V, E)$, where w_e denotes the weight of edge $e \in E$.

   Order the edges arbitrarily, say $e_1, \ldots, e_m$.
   $T \leftarrow \emptyset$.
   for $i = 1$ to m **do**
       if $T \cup \{e_i\}$ is acyclic **then**
           $T \leftarrow T \cup \{e_i\}$.
       else
           $e' \leftarrow$ max weight edge in the unique cycle in $T \cup \{e_i\}$.
           $T \leftarrow (T \cup \{e_i\}) \setminus \{e'\}$.
       end if
   end for
   Return T.

   Prove that this algorithm correctly finds a minimum weight spanning tree of G.

4. Consider the following algorithm on an edge-weighted connected graph $G = (V, E)$, where w_e denotes the weight of edge $e \in E$.

   Order the edges arbitrarily, say $e_1, \ldots, e_m$.
   $T \leftarrow E$.
   for $i = 1$ to m **do**
       if $T \setminus \{e_i\}$ is connected **then**
           $T \leftarrow T \setminus \{e_i\}$.
       else
           $e' \leftarrow$ min weight edge in E whose endpoints are in different connected components of $T \setminus \{e_i\}$.
           $T \leftarrow (T \setminus \{e_i\}) \cup \{e'\}$.
       end if
   end for
   Return T.

   Prove that this algorithm correctly finds a minimum weight spanning tree of G.

5. Design a polynomial time algorithm that, given an edge weighted connected graph $G = (V, E)$ with weight w_e of edge $e \in E$, classifies the edges into the following three categories.

   (a) Edges that do not belong to any min-weight spanning tree of G.

   (b) Edges that belong to some but not all min-weight spanning trees of G.

   (c) Edges that belong to every min-weight spanning tree of G.