Name: Harshit Jain
Access ID: hmj5262
Recitation: 8

Problem 0

Points:

Acknowledgements

(a) I worked with Yug Jarodiya.

(b) I did not consult with anyone in my group members.

(c) I did not consult any non-class materials.

Problem 1

Points:

The minimum cut of a weighted graph is defined as the minimum sum of weights of edges that, when removed from the graph, divide the graph into two sets.

Algorithm 1: UniqueMinimumCut

Input : $G = (V, (E, \ell_e))$
Output: A unique cut is present or not

C = STCut(G)
$|C|$ = CapacityOfCut(C)
for $e_i \in C$ **do**
  capacity(e_i) $+= 1$
  $|C_i| =$ STCut(G)
  if $|C| == |C_i|$ *and* $C \neq C_i$ **then**
    return "Min-cut is not unique"
  end if
end for
return "Min-cut is unique"

Conversely, if there is a different minimum cut C' in the original graph, there will be some $e_i \in C$ that is not in C', so increasing the capacity of that edge will not change the volume of C', thus $|C| = |C_i|$. In conclusion, the graph has a unique minimum cut iff $|C| < |Ci|\ \forall\ i$. The algorithm takes at most $n+1$ computing of minimum cuts, and therefore runs in **polynomial time**.

Problem 2

Points:

The cut property: Let A be a subset of edges of some MST of $G = (V,E)$. Let $(S,V-S)$ be a cut that respects A. Let e be the lightest edge across the cut. Then $A \cup e$ is part of the MST.

Using the cut property, we can take edge e which is the lightest edge in the subgraph H, so we know that edge $e \in T \cap H$ because it is the lightest edge and will preserve among H. We can take edge e as the cut edge for H and so the cut set is $(H \cap S, H - S)$ which is defined in the cut property above. Because the edge $e \in T \cap H$, that same cut exists in an MST of H.

Problem 3

Points:

⇒ Given that $G = (V, E)$ and $V' \subset V$.

⇒ Edge (u, v) is present in T', so this the edge is the least weighted edge here.

⇒ Now, the MST has V' edges and contains no cycle. Now, add the edge (u, v) to the MST.

⇒ The cycle includes edge (u, v) and other edges which are part of T, you can notice that one cycle forms in T due to the addition of edge (u, v).

⇒ In this cycle, select an edge where weight is highest, it certainly will not be the edge (u, v) because edge (u, v) weights the least. Hence you will select an edge from your earlier minimum spanning tree T. Let the edge be E'.

⇒ Remove E' from T and add edge (u, v). Let this tree be called T'.

⇒ If we remove one edge from this cycle, the connectivity of new tree will still be maintained.

⇒ As there are V' edges, there are no cycles in T'. So the T' is the spanning tree.

⇒ In order to be minimum spanning tree, Weight(T') = Weight(T) − (E' - edge (u, v)).

⇒ As per the follows, we got to know that (E'- edge (u, v)) is the positive entity.

⇒ Hence Weight(T') < Weight(T), therefore T cannot be minimum spanning tree. (T') is the minimum spanning tree which has edge(u, v).

⇒ **Thus, T' is the minimum spanning tree of G'.**