CSC 373 Sample Solutions for Week 3 Tutorial Winter 2019

1. Hint: Use idea from exchange argument for MST correctness.

   Proof:

   For a contradiction, suppose T is a MST that does not contain e.

   Then T contains some path between the endpoints of e.

   Pick some edge e’ on this path.
   Then T’ = T u {e} - {e’} is a spanning tree (as in the proof).

   But c(T’) = c(T) + c(e) - c(e’) < c(T) because c(e’) > c(e)
   (by assumption, c(e) is minimum and unique).

   This contradicts the fact that T is a MST.

2. Hint: Think about when it is "safe" to remove edges without
   disconnecting a graph.

   Counter-example:

```
G = a --1-- b --2-- c --4-- d
     \______3______/
```

   Additional condition:

   If e belongs to some cycle C in G, then e belongs to no MST.

   Proof:
   Hint: You need an exchange argument different from the one in class
   (adding an edge to a tree and removing another edge from the cycle that
   results). Instead, you want to remove an edge from a tree then add
   another edge to connect the two components that result.

   For a contradiction, suppose e belongs to a MST T.

   Consider T - {e}. This is made up of two connected components.
   Because e belongs to some cycle C, there is a way to get from one
   endpoint of e to the other along this cycle. So there is at least
   one edge e’ of C that connects both components.

   Then T’ = T u {e’} - {e} is a spanning tree, and
   c(T’) = c(T) + c(e’) - c(e) < c(T) because c(e’) < c(e)
   (by assumption, c(e) is maximum and unique).

   This contradicts the fact that T is a MST.

3. Hint 1: Try to use same proof structure as before.
   Hint 2 (Reminder): Beware! You need to define "promising" and "extends"
   carefully to work with the algorithm!
   Hint 3: Exchange lemma will be different from the one in class, to match
   structure of algorithm. This will require the idea from above (working
   with cycles).

```
Def: MST T* "extends" T_i iff T* (_ T_i (_ T* u {e_{i+1},...,e_m}.
Def: T_i "promising" iff some T* extends T_i.
Proof: T_i promising for all i.
BC: T* (_ T_0 (_ T* u {e_1,...,e_m} for all MSTs T*.
I.H.: For some i >= 0, suppose T* extends T_i.
I.S.: Either T_{i+1} = T_i or T_{i+1} = T_i - {e_{i+1}}.
  - Case 1: T_{i+1} = T_i when T_i - {e_{i+1}} is disconnected, so
    e_{i+1} must belong to T*, i.e., T* extends T_{i+1}.
  - Case 2: T_{i+1} = T_i - {e_{i+1}}. Either e_{i+1} in T* or
    e_{i+1} not in T*.
      . Subcase 2.1: e_{i+1} not in T* implies T* extends T_{i+1}.
      . Subcase 2.2: e_{i+1} in T*.
        Consider T* - {e_{i+1}}: two components A, B (subsets of
        vertices).
        But endpoints of e_{i+1} still connected in T_{i+1}: some edge
        e_j in T_{i+1} crosses between A, B.
        Because T_i (_ T* u {e_{i+1},...,e_m}, must have j > i+1 so
        w(e_j) <= w(e_{i+1}). Let T** = T* - {e_{i+1}} u {e_j}.
        Then w(T**) <= w(T*) so T** is a MST and extends T_{i+1}.
Conclusion: every T_i promising => T_m promising => T_m = T* for some
MST => T_m is a MST.
```