Problem Set 6
Due: Jan 8, 2021, 8pm

Please answer each of the following problems.

Note: For all problems, if you include pseudocode in your solution, please also include a brief English description of what the pseudocode does.

1. **True or False? (10 points)** Decide whether the following statement is true or false. If it is true, give a short justification. If it is false, give a counterexample.

   > Let $G = (V, E)$ be an arbitrary connected weighted undirected graph with no self-loops (that is, no edges from a vertex to itself), so that the weight of an edge $e \in E$ is $w(e)$, and all the weights are distinct. Let $e^* \in E$ be the cheapest edge: that is, the edge so that $w(e^*) = \min_{e \in E} w(e)$. Then there is a minimum spanning tree T of G that contains e^*.

   [We are expecting: your true/false answer, and either a justification (a few sentences) or a counterexample (a graph).]

2. **Greedy algorithms don't always work. (10 points)** Let $G = (V, E)$ be an undirected unweighted graph. Say that a vertex v can "see" an edge if v is an endpoint of that edge. Say that a set $S \subseteq V$ is "all-seeing" if every edge $e \in E$ is seen by at least one vertex in S. In this problem, we will try to greedily construct the smallest all-seeing set possible.

   Consider the following greedy algorithm to find an all-seeing subset:

```
findAllSeeingSubset(G):
  S = {}
  while G contains edges:
    choose an edge e = (u,v) in G
    S.add(u)
    S.add(v)
    remove u and all of its adjacent edges from G
    remove v and all of its adjacent edges from G
  return S
```

   (a) (4 pt) Prove that `findAllSeeingSubset(G)` always returns an all-seeing subset.
   [We are expecting: A short but rigorous proof (a few sentences).]

   (b) (6 pts) Give an example of a graph on which `findAllSeeingSubset(G)` does not return a smallest all-seeing subset, for at least one way of choosing edges.
   [We are expecting: your example, and a brief justification that it is an example of this.]

   (c) (0 pts) **[BONUS: this question is NOT REQUIRED but it might be fun to think about. It will not affect your grade.]**

      i. Can you find a greedy algorithm that does find a smallest all-seeing subset?

      ii. `findAllSeeingSubset` has a very nice property. Let $OPT(G)$ denote the size of the smallest all-seeing subset in G. Prove that for all graphs G,

      $$\frac{|\texttt{findAllSeeingSubset}(G)|}{OPT(G)} \leq 2.$$

   [We are expecting: Nothing, this problem is optional.]

3. **Properties of Maximum Flow. (10 points)** Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. If f is a maximum s-t flow in G. Decide whether you think the following statements are true or false. If one is true, give a short explanation. If one is false, give a counterexample.

   (a) f saturates every edge out of s with flow (i.e., for any edge e out of s, we have $f(e) = c_e$)

   (b) Let E be the edges directing from s to t crossed by a minimum cut, then f saturates every edge $e \in E$ with flow (i.e., for any edge $e \in E$, we have $f(e) = c_e$)

4. **Enlarge Maximum Flow. (10 Points)** Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. Let (A, B) be a minimum s-t cut with respect to these capacities $c_e : e \in E$. Now suppose we add 1 to the capacity of every edge; then is (A, B) still a minimum s-t cut with respect to these new capacities $1 + c_e : e \in E$? If your answer is true, give a short explanation. If your answer is false, give a counterexample.