CMPSCI 611 FALL '17: HOMEWORK 3
DUE 8PM, FRIDAY OCTOBER 27TH

- Homework may be completed in group of size with at most four students. You're not allowed to use material from the web or talk about the homework with anybody outside your collaboration group (aside from the lecturer or TA.)
- Solutions should be typed and uploaded as a pdf to `gradescope.com`.
- To get full marks, answers must be sufficiently detailed, supported with rigorous proofs (of both correctness and running time analysis). Faster algorithms will typically get more marks than slower algorithms.

Question 1. Suppose there are n questions in a homework that take times $t_1, t_2, \ldots, t_n$ and there are three people in a homework group. Design a dynamic program to determine whether it is possible to split the questions up amongst the team members such that each person takes an equal amount of time. The algorithm should run in time that is polynomial in n and $T = \sum_i t_i$. You may assume all times are integers.

Question 2. Consider a weighted graph on n nodes where all weights are positive and the graph is complete, i.e., there is an edge between every pair of nodes in the graph. We say these weights satisfy the triangle inequality if, for each edge (u, v) the weight of that edge $w(u, v)$ is less or equal to the length, $\delta(u, v)$, of the shortest path between u and v. Design a polynomial time algorithm that takes a complete graph with weights that don't satisfy the triangle inequality and reduces some of these weights such that the resulting weights satisfy the triangle inequality; your algorithm should minimize the *number* of edges whose weights are reduced and it doesn't matter by how much they are reduced. **Hint:** Note that reducing one edge weight can potentially cause other edge weights to violate the triangle inequality.

Question 3. Consider a directed graph $G = (V, E)$ where each edge e has weight w_e and a special node $t \in V$. The weights are all positive. A friend has computed values $d[v]$ for each $v \in V$ and claims that $d[v] = \delta(v, t)$, i.e., that the length of the shortest path from v to t is $d[v]$. All the $d[v]$ values are finite. Give an $O(|E|)$ time algorithm that determines whether your friend has correctly computed the values. Note that your algorithm just needs to verify the claimed values, i.e., if your friend is wrong, you just need to identify this fact rather than finding the correct values.

Question 4. Let G be a directed graph with n nodes including one labelled s and one labelled t. Let k be an integer between 1 and n. Suppose every edge has unit capacity. Design an efficient algorithm that finds a set of k edges whose deletion reduces the maximum s-t flow by as much as possible.

Question 5. Read Section 5.5 of the notes. A vertex cover of an undirected graph $G = (V, E)$ is a subset $S \subseteq V$ such that for each edge $\{u, v\} \in E$, one or both of u, v are in S. Design an algorithm to find the vertex cover in a bipartite graph that has the smallest number of nodes.

Question 6. Consider a bipartite graph $G = (L \cup R, E)$ where $|L| = |R| = n$. For any $S \subseteq L$, let $\Gamma(S) = \{v \in R : (u, v) \in E \text{ for some } u \in S\}$. Use the max-flow min-cut theorem to prove that there is a matching of size n iff $|\Gamma(S)| \geq |S|$ for all $S \subseteq L$.