CIT 596 Online **Spring 2020**

Module 11–12

Name: NAME HERE

Questions requiring written answers.

1. Suppose you have a flow network G with integer capacities, and an integer maximum flow f . Suppose that, for some edge e, we increase the capacity of e by one. Describe an $O(|E|)$ time algorithm to find a maximum flow in the modified graph.

2. (Problem 9,Chapter 7, Page 419 of text) Network flow issues come up in dealing with natural disasters and other crises, since major unexpected events often require the movement and evacuation of large numbers of people in a short amount of time.
   Consider the following scenario. Due to large-scale flooding in a region, paramedics have identified a set of n injured people distributed across the region who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals, depending on where they are right now).
   At the same time, one doesn't want to overload any one of the hospitals by sending too many patients its way. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the injured people in such a way that the load on the hospitals is *balanced*: Each hospital receives at most $\lceil n/k \rceil$ people.
   Give a polynomial-time algorithm that takes the given information about the people's locations and determines whether this is possible.

3. In a flow network with n nodes and m edges, given a flow f from s to t, prove that f can be decomposed into flows on at most m paths. A stronger claim would be that the flow can be decomposed into a flow on at most n paths. Is this claim true? Prove or give a counter-example.

4. A graph is called *d-regular* if all vertices in the graph have degree d. Prove that a d-regular bipartite graph (for $d \geq 1$) has a perfect matching. Furthermore, show that a d-regular bipartite graph is the disjoint union of d perfect matchings. **Hint:** The min-cut in an appropriate flow network can be useful in answering this question.

5. Show the following problem is in NP:

   Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, are they *isomorphic*?

   Recall that two graphs are said to be isomorphic if there exists a bijection f from V_1 to V_2 such that for any two vertices $u, v \in G_1, \;\; (u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$.

6. We have seen how the independent set (decision) problem is NP complete. Suppose we are given a subroutine that will solve this problem. Show how you can use the subroutine to find a maximum independent set in a graph.

7. Consider the following two decision problems: In the SINGLE-SOURCE-DISTANT-VERTEX problem you are given a weighted, directed graph G, a source vertex s and a number L, and asked if there is a vertex u whose distance from s is at least L. In the ALL-PAIRS-DISTANT-VERTEX problem, you are just given a weighted, directed graph G and a number L, and asked if there are two vertices u and v such that the distance from u to v is at least L.

   Given an algorithm that solves SINGLE-SOURCE-DISTANT-VERTEX, show how you can solve ALL-PAIRS-DISTANT-VERTEX with at most n calls to the algorithm.

How about the other direction? Given an algorithm for solving ALL-PAIRS-DISTANT-VERTEX can you see a way of using it to solve SINGLE-SOURCE-DISTANT-VERTEX efficiently? If you are not able to, that's okay. An explanation of your attempt and the difficulty you ran into will suffice.