COL351 Holi2023: Tutorial Problem Set 8

1. Two paths in a graph are said to be *edge-disjoint* if they don't share an edge. Design a polynomial-time algorithm which, given a directed graph $G = (V, E)$ and two vertices $s, t \in V$, outputs a maximum collection of pairwise edge-disjoint paths from s to t in G. (Note that you need to output a list of s-t paths, not just the maximum number of s-t paths.) How does your algorithm and proof of correctness change if the graph is undirected? Note that in case of undirected graphs, edge-disjoint paths are not allowed to use a common edge in opposite directions either.

2. Consider a hospital that has employed n nurses. You are given the job of assigning duties to them. The week is divided into m time slots, and you are required to assign a subset of time slots to each nurse. Each nurse i has specified a subset A_i of slots when he / she is available. The hospital has specified a number c_i for each nurse i, which is an upper bound on the number of slots that can be assigned to him / her. Finally, the hospital has also specified a number d_j for each slot, which is the number of nurses that are required to be present in slot j. From this input, you are required to output the set B_i of slots assigned to each nurse i, satisfying the following constraints.

   - Each nurse is only assigned slots in which he / she is available. That is, $B_i \subseteq A_i$ for each i.
   - Each nurse i is assigned at most c_i slots. That is, $|B_i| \leq c_i$ for all i.
   - In each slot j, at least d_j nurses are present. That is, for every j, at least d_j sets out of $A_1, \ldots, A_n$ contain slot j.

   Design an algorithm that outputs an assignment of slots to nurses satisfying the above constraints if such an assignment exists, and otherwise outputs "NO". Prove that your algorithm is correct. Your algorithm must run in time polynomial in n and m.

3. Let $\mathcal{N} = (G, s, t, C)$ be a network and let f^* be a maximum flow in it. Design a polynomial-time algorithm that, given $\mathcal{N}$ and f^*, decides whether f^* is the unique maxflow in $\mathcal{N}$.

4. [Kleinberg-Tardos Chapter 7 Exercise 24] Design a polynomial time algorithm that, given a network (G, s, t, C), decides whether the network has a unique min-cut (i.e., a cut of capacity strictly less than that of all other cuts).

5. An edge in a network is called downward-critical if for all $\varepsilon > 0$, decreasing the capacity of this edge by ε decreases the maximum flow in the network. An edge in a network is called upward-critical if for all $\varepsilon > 0$, increasing the capacity of this edge by ε increases the maximum flow in the network. Design a polynomial time algorithm which identifies all the downward-critical and upward-critical edges in an input network.