Homework 8

Due date: 3/10/2021 right before class

Problem 1

The problem of finding a Clique of size k in a graph $G = (V, E)$ requires finding a set of vertices of size k such that every pair of vertices in the set is adjacent. The problem of finding an Independent-Set of size k in a graph $G = (V, E)$ requires finding a set of vertices of size k such that no pair of vertices in the set is adjacent.

Given that the Clique problem is NP-Complete, Show that the Independent-Set problem is NP-Complete as well.

Solution 1

In the discussion slides.

Problem 2

A Hamiltonian Cycle in an undirected graph G is a simple cycle that goes through all nodes. A Hamiltonian Path in G is a simple Path that goes through all nodes.

Given that the Hamiltonian path problem is NP-Complete, show that the Hamiltonian cycle problem is NP-Complete as well.

Solution 2

a. HC is in NP. Given a certificate (a proposed solution) we can iterate the graph in the order it states in polynomial time to verify.

b. Given a graph $G = (V, E)$ we reduce the HP problem to the HC problem by adding a special vertex v^* and connect all vertices in the original graph to it. Now if the original graph had a HP, in the new graph we will be able to find a HC. The reduction is of course polynomial since we can create a new vertex and connect all vertices to it in linear time.

Problem 3. Page 425 Q19

You've periodically helped the medical consulting firm Doctors Without Weekends on various hospital scheduling issues, and they've just come to you with a new problem. For each of the next n days, the hospital has determined the number of doctors they want on hand; thus, on day i, they have a requirement that exactly p_i doctors be present. There are k doctors, and each is asked to provide a list of days on which he or she is willing to work. Thus doctor j provides a set L_j of days on which he or she is willing to work. The system produced by the consulting firm should take these lists and try to return to each doctor j a list L'_j with the following properties.
(A) L'_j is a subset of L_j, so that doctor j only works on days he or she finds acceptable.
(B) If we consider the whole set of lists $L'_1, ..., L'_k$, it causes exactly p_i doctors to be present on day i, for $i = 1, 2, ..., n$.

a. Describe a polynomial-time algorithm that implements this system. Specifically, give a polynomial-time algorithm that takes the numbers $p_1, p_2, ..., p_n$, and the lists $L_1, ..., L_k$, and does one of the following two things.
   – Return lists $L'_1, L'_2, ..., L'_k$ satisfying properties (A) and (B); or
   – Report (correctly) that there is no set of lists $L'_1, L'_2, ..., L'_k$ that satisfies both properties (A) and (B).

b. The hospital finds that the doctors tend to submit lists that are much too restrictive, and so it often happens that the system reports (correctly, but unfortunately) that no acceptable set of lists $L'_1, L'_2, ..., L'_k$ exists. Thus the hospital relaxes the requirements as follows. They add a new parameter $c > 0$, and the system now should try to return to each doctor j a list L'_j with the following properties.
   (A*) L'_j contains at most c days that do not appear on the list L_j.
   (B) (Same as before) If we consider the whole set of lists $L'_1, ..., L'_k$, it causes exactly p_i doctors to be present on day i, for $i = 1, 2, ..., n$. Describe a polynomial-time algorithm that implements this revised system. It should take the numbers $p_1, p_2, ..., p_n$, the lists $L_1, ..., L_k$, and the parameter $c > 0$, and do one of the following two things.
   - Return lists $L'_1, L'_2, ..., L'_k$ satisfying properties (A*) and (B); or
   - Report (correctly) that there is no set of lists $L'_1, L'_2, ..., L'_k$ that satisfies both properties (A*) and (B).

Solution 3

In the end.

Problem 4. Page 435 Q33

Let $G = (V, E)$ be a directed graph, and suppose that for each node v, the number of edges into v is equal to the number of edges out of v. That is, for all v,

$|(u, v) : (u, v) \in E| = |(v, w) : (v, w) \in E|.$

Let x, y be two nodes of G, and suppose that there exist k mutually edge-disjoint paths from x to y. Under these conditions, does it follow that there exist k mutually edge-disjoint paths from y to x? Give a proof or a counterexample with explanation.

Solution 4

In the end.

We construct a flow network as follows. There is a node u_j for each doctor j and a node v_i for each day i. There is an edge (u_j, v_i) of capacity 1 if doctor j can work on day i, and no edge otherwise. There is a source s, and an edge (s, u_j) of capacity $|L_j|$ for each j. There is a sink t, and an edge (v_i, t) of capacity p_i for each i.

Now we ask: is there an s-t flow of value $\sum_{i=1}^{n} p_i$ in this flow network? If there is, then there is an integer-valued flow, and we can produce a set of lists $\{L'_i\}$ from this as follows: assign doctor j to day i if there is a unit of flow on the edge (u_j, v_i). In this way, each doctor only works on days he or she finds acceptable, and each day i has p_i working on it. Conversely, if there is a valid set of lists for the doctors, then there will be a flow of value $\sum_{i=1}^{n} p_i$ in the network: we simply raise one unit of flow on each path s-u_j-v_i-t, where doctor j works on day i.

The total running time for this algorithm is dominated by the time for a flow computation on a graph with $O(kn)$ edges, where the total capacity out of the sink is $O(kn)$; thus, the total running time is $O(k^2n^2)$.

(b) We take the previous flow network and add some nodes to it, modeling the requirements. For each doctor j, we add a "spill-over" node u'_j. There is an edge (u'_j, v_i) of capacity 1 for each day i such that doctor j *doesn't* want to work on i. There is an edge (s, u'_j) of capacity c for each j.

Again we ask: is there an s-t flow of capacity $\sum_{i=1}^{n} p_i$ in this flow network? If there is, then there is an integer-valued flow, and we can produce a set of lists $\{L'_i\}$ from this as follows: assign doctor j to day i if there is a unit of flow on the edge (u_j, v_i) *or* if there is a unit of flow on the edge (u'_j, v_i).

[1] ex329.573.267

If we put a capacity of 1 on each edge, then by the integrality theorem for maximum flows, there exist k edge-disjoint x-y paths if and only if there exists a flow of value k. By the max-flow min-cut theorem, this latter condition holds if and only if there is no x-y cut (A, B) of capacity less than k.

Now suppose there were a y-x cut (B', A') of capacity strictly less than k, and consider the x-y cut (A', B'). We claim that the capacity of (A', B') is equal to the capacity of (B', A'). For if we let $\delta^-(v)$ and $\delta^+(v)$ denote the number of edges into and out of a node v respectively, then we have

$$\begin{aligned} c(A', B') - c(B', A') &= |\{(u, v) : u \in A', v \in B'\}| - |\{(u, v) : u \in B', v \in A'\}| \\ &= |\{(u, v) : u \in A', v \in B'\}| + |\{(u, v) : u \in A', v \in A'\}| - \\ &\quad |\{(u, v) : u \in A', v \in A'\}| - |\{(u, v) : u \in B', v \in A'\}| \\ &= \sum_{v \in A} \delta^+(v) - \sum_{v \in A} \delta^-(v) \\ &= 0. \end{aligned}$$

It follows that $c(A', B') < k$ as well, contradicting our observation in the first paragraph. Thus, every y-x cut has capacity at least k, and so there exist k edge-disjoint y-x paths.

[1] ex52.743.508