Stefanie Shidoosh

CS180: HW #8

1. Suppose $2n$ teams play in a round-robin tournament. Over a period of $2n - 1$ days, every team plays every other team exactly once. There are no ties. Show that for each day we can select a winning team, without selecting the same team twice. (Hint: Consider a bipartite graph with a set of team vertices and a set of day vertices. Take any set of days W and assume not all teams won in some day in W. Let *tw* be a team that did not win in any day in W. Consider the implication on the number of teams that won at least once in some day in W. Invoke Hall's Theorem.)

Let the set of team vertices be one half of the bipartite graph, V_t, and set of day vertices be the other half, V_d.

Proof by Contradiction:

Suppose that we cannot select a winning team each day, without selecting the same team twice

Hall's Marriage Theorem states there has to be a set of n days, such that there are less than n winners in n days

Suppose there exists a team that lost every game in n days

It then follows that this team lost to n *different* teams, making these teams winning teams

This is a contradiction as it was stated that there are less than n winners

Therefore, we can select a winning team each day, without selecting the same team twice.

Stefanie Shidoosh

2. Given an undirected graph $G = (V, E)$ and an integer k. A clique of G is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E. The **Clique** problem asks whether G contains a clique of size at least k. An independent set of G is a subset $V' \subseteq V$ of vertices such that each edge in E is incident on at most one vertex in V'. The **Independent-Set** problem asks whether G' contains an independent set of size at least k'. We proved in class that the Clique problem is NP-complete. Show that the independent set problem is same by reduction from Clique problem.

To prove that the Independent-Set problem is NP-complete, we need to show that it is both NP and NP-hard.

It is NP:

Consider the unweighted graph G, and the certificate is a set of vertices. If there is a witness algorithm that takes an input, G, verifies that the certificate, V, is a yes instance of the particular inputted problem, then the problem is NP.

Consider the following algorithm:

Checks that the all vertices in the certificate are vertices in the graph.

Checks that there are no edges between any two vertices in the certificate.

This algorithm runs in $O(V + E)$, a polynomial time complexity.

Therefore, the Independent-Set problem is NP.

It is NP-hard:

We can show this by reducing the Independent-Set problem from the Clique problem:

An instance of Clique is a graph G and an integer k. Take k and the complement of G, G', as the input to the Independent-Set.

A yes instance of Clique maps to a yes instance of Independent-Det and a yes instance of Independent-Set maps to a yes instance of Clique.

Suppose G is a yes instance of Clique

If this is true, then for all e in $E(C)$, where C is the Clique, are all also in $E(G)$. It then follows that for all e in $E(C)$, there is no e that is in $E(G')$. Thus, $V(C)$ is an independent set in G', and so G' is an instance of Independent-Set.

Suppose G' is a yes instance of Independent-Set

If this is true, then for all e in $E(I)$, where I is independent-set, there is no e in $E(G')$. It then follows that all e in $E(I)$, are in $E(G)$. Thus, $V(I)$ is a clique in G, and so G is an instance of Clique.

Therefore, because the Independent-Set problem is both NP and NP-hard, we can conclude that it is NP-complete given G and k.

Stefanie Shidoosh

3. Show that the following three problems are polynomial time reducible to each other.
• **Set-Cover**: Given a collection of sets, and a number k, the Set-Cover problem asks if there are at most k sets from the collection of sets such that their union contains every element in the union of all sets.
• **Hitting-Set**: Given a collection of sets, and a number k, the Hitting-Set problem asks if are there at most k elements of the sets such that there is at least one element from each set?
• **Dominating-Set**: Given an undirected graph G, and a number k, the Dominating-Set problem asks if there is a subset of vertices of size $\leq k$ such that every vertex in the graph is either in the subset or has a neighbor that is in the subset.
Prove Set-Cover, Hitting-Set and Dominating-Set are polynomial-time reducible to each other. (Hint: One strategy is to show Set-Cover $\leq p$ Hitting-Set, Hitting-Set $\leq p$ Dominating-Set and Dominating-Set $\leq p$ Set-Cover. In class, we have seen Vertex-Cover reduced poly to Dominating-Set)

Set-Cover $\leq p$ Hitting-Set:

Given a set $S_1,\ldots, S_n$ and a ground set W = $\{w_1,\ldots,w_m\}$ of elements to cover, let $a_1,\ldots, a_n$ represents the sets and let $E_1,\ldots, E_m$ represent the elements. a_i hits B_j *iff* S_i contains w_j.

It then follows that there is a hitting set size of at most k iff there is a set cover size of at most k, which is a polynomial reduction.

Hitting-Set $\leq p$ Dominating-Set:

In class, we have seen Vertex-Cover polynomial reduction to Dominating-Set, so showing that Hitting-Set $\leq p$ Vertex-Cover implies Hitting-Set $\leq p$ Dominating-Set
Given an undirected graph G and k, let N be the set of nodes in G, B_i to be the set of the two vertices that compose each edge e_i.

G has a vertex cover *iff* A has a hitting set size k for each element in B, which is a polynomial reduction.

Dominating-Set $\leq p$ Set-Cover:

Given a set $S_1,\ldots, S_n$ and a ground set W = $\{w_1,\ldots,w_m\}$ of elements to cover

Suppose there is a graph G which contains a vertex v_i, for each S_i and a vertex u_i, for each element w_j

There is an edge (v_i, w_j) *iff* w_j is in S_i

Add an edge (v_i, v_i) for each of of sets S_i and S_j

There is a dominating set size of at most k in G *iff* there is a set cover size of at most k, which is a polynomial reduction.

Stefanie Shidoosh

4. Given a directed graph $G = (V, E)$ and a pair of vertices s, t in G, the **Hamiltonian Path** problem asks whether there is a simple path from s to t that visits every vertex of G exactly once. The **Hamiltonian Cycle** problem asks if there is a cycle in a directed graph G that visits every vertex exactly once. Show that Hamiltonian Path and Hamiltonian Cycle problems are polynomial-time reducible to each other.

> Given a graph G of which we need to find Hamiltonian Cycle, for a single edge e=(u,v) add new vertices u' and v' such that u' is connected only to u and v' is connected only to v, constructing a new graph G'.
> G' has a Hamiltonian path if and only if G has a Hamiltonian cycle with the e.
> Run the Hamiltonian path algorithm on each G' for each e in E(G). If all graphs have no Hamiltonian path, then G has no Hamiltonian cycle. If at least one G' has a Hamiltonian path, then G has a Hamiltonian cycle which contains the edge e.