CSE 6140 / CX 4140 Assignment 3
due Oct 16, 2020 at 11:59pm on Canvas

1 Dominating set [12 pts]

You're configuring a large network of workstations, which we'll model as an undirected graph G; the nodes of G represent individual workstations and the edges represent direct communication links. The workstations all need access to a common core database, which contains data necessary for basic operating system functions.

You could replicate this database on each workstation; this would make look-ups very fast from any workstation, but you'd have to manage a huge number of copies. Alternately, you could keep a single copy of the database on one workstation and have the remaining workstations issue requests for data over the network G; but this could result in large delays for a workstation that's many hops away from the site of the database.

So you decide to look for the following compromise: You want to maintain a small number of copies, but place them so that any workstation either has a copy of the database or is connected by a direct link to a workstation that has a copy of the database. In graph terminology, such a set of locations is called a *dominating set*.

Thus we phrase the *Dominating Set Problem* as follows. Given the network G, and a number k, is there a way to place k copies of the database at k different nodes so that every node either has a copy of the database or is connected by a direct link to a node that has a copy of the database?

Show that Dominating Set is NP-complete. Follow all steps we have outlined in class for a complete proof. *Hint*: consider the Vertex Cover problem.

Solution:

- Step 1: Show that *Dominating Set Problem* is in NP.
  A potential solution would be $L_k = [v_1, v_2, ..., v_k]$, which is a list of k vertices in the graph G that was placed a copy of the database. To check if L_k is a correct solution, we can loop through all the vertices in the L_k, store their neighbors in a hashset, and then check if the hashset has a length equal to $|V|$, the number of vertices in G. If we use a hashset to store L_k, then the worst runtime for checking a potential solution is $O(k|E|)$, where E is the number of edges in G. Therefore, *Dominating Set Problem* is in NP.

- Step 2: Choose an NP-complete problem X.
  Set Cover: Given a set U of elements, a collection $S_1, S_2, ..., S_m$ of subsets of U, and an integer k, dose there exist a collection of $\leq k$ of these sets whose union is equal to U?
  We know the *Set Cover* problem is NP-complete.

- Step 3: Prove that $X \leq_p$ Y, where X is the *Vertex Cover* problem and Y is the *Dominating Set Problem.*
  - loop through a vertex cover and records all it's neighbors. When their neighbors contains all the vertices, we can stop and add all the visited vortices into our solution set S. After that, add all the isolated vertices in G to S. The resulting S would be a solution of Y for $k = \hat{k} + I$ where I is the number of isolated vertices in G.

2 Frenemies [12 pts]

Assume you are planning a dinner party and going to invite a set of friends. However, among them, there are some pairs of persons who are enemies. You need to create a seating plan and you are wondering if it is possible to arrange this set of n friends of yours around a round table such that none of the two enemies will seat next to each other. Given the set of the n friends and the set of the pairs of enemies, prove that this problem is NP-Complete. Remember to follow the steps from lecture to prove NP-completeness.

You can use the fact that `Hamiltonian Cycle (HC)` is NP-complete.

Solution:

3 Let's go hiking [26 pts]

Alex and Baine go hiking together. They bring a bag of items and want to divide them up. For the following scenarios, decide whether the problem can be solved in polynomial time. If yes, give a polynomial-time algorithm; otherwise prove the problem is NP-complete.

- (8 pts) The bag contains n items of two weights: 1lb and 2lb. Alex and Baine want to divide the items evenly so that they carry the same amount of weight.

  Solution:

- (9 pts) The bag contains n items of different weights. Again they want to divide the items evenly.

  Solution:

- (9 pts) The bag contains n items of different weights. They want to divide the items such that the weight difference of items they carry is less than 10lbs.

Solution:

Hint: Recall Subset Sum problem: given a set X of integers and a target number t, find a subset $Y \subset X$ such that the members of Y add up to exactly t.