CS 330 – Summer 2019 – Lab 10 Solutions

Question 1 (Efficient Recruiting Problem - KT Ch8,Ex3)**.**
Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who's skilled at each of the n sports covered by the camp (baseball, volleyball, and so on). They have received job applications from m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is: For a given number $k < m$, is it possible to hire at most k of the counselors and have at least one counselor qualified in each of the n sports? We'll call this the Efficient Recruiting Problem.
Show that Efficient Recruiting is NP-complete.

Solution: 1. The problem is in NP: Given a set of k counselors, we can easily identify in linear time in the number of sports and k counselors, if for every sport there is at least one counselor skilled for it.
2. We will prove that the problem is NP-complete, by a reduction from the *Set Cover*. Recall, that in the *Set Cover* problem, given a set U of n elements, and a collection of m subsets of U- whose union is the set of elements- we ask whether there are at most k of these sets whose union is equal to all of U. Given such an instance of the *Set Cover*, we can construct an instance of *Efficient Recruiting* as follows: For each element of U, create a sport. For each of the m subsets, create a counselor and let this counselor be skilled at the sports that are items of this subset. Clearly, this reduction takes polynomial time, and
3. There are k counselors that all together are skilled in every sport, iff there are k subsets, whose union is U.
Hence, *Set Cover* $\leq_P$ *Efficient Recruiting*. Therefore, ER is an NP-Complete problem.

Question 2 (KT Ch8,Ex19)**.**
Suppose you're acting as a consultant for the port authority of a small Pacific Rim nation. They're currently doing a multi-billion-dollar business per year, and their revenue is constrained almost entirely by the rate at which they can unload ships that arrive in the port. Handling hazardous materials adds additional complexity to what is, for them, an already complicated task. Suppose a convoy of ships arrives in the morning and delivers a total of n cannisters, each containing a different kind of hazardous material. Standing on the dock is a set of m trucks, each of which can hold up to k containers.
Here are two related problems, which arise from different types of constraints that might be placed on the handling of hazardous materials. For each of the two problems, give one of the following two answers:
•*A polynomial-time algorithm to solve it; or*
•*A proof that it is NP-complete.*

(a) *For each cannister, there is a specified subset of the trucks in which it may be safely carried. Is there a way to load all n cannisters into the m trucks so that no truck is overloaded, and each container goes in a truck that is allowed to carry it?*

(b) *In this different version of the problem, any cannister can be placed in any truck; however, there are certain pairs of cannisters that cannot be placed together in the same truck. (The chemicals they contain may react explosively if brought into contact.) Is there a way to load all n cannisters into the m trucks so that no truck is overloaded, and no two cannisters are placed in the same truck when they are not supposed to be?*

Solution:

(a) We can reduce this problem to a network flow problem and solve it in polynomial time.
We have to create a node for each of the n cannisters c_i and one for each of the m trucks, r_j. There is also an edge of capacity one from the cannister-nodes to the truck-nodes, whenever cannister i can be carried by truck j. We also have a source node that we link to every cannister-node with capacity 1, and a sink node, that is connected from every truck-node with an edge of capacity k.
There is a feasible way to place all cannisters in trucks iff there is an $s-t$ flow of value n. If there is a feasible placement, we can send one unit of flow from s to t, for every cannister carried by a truck. Constraints are not violated, as each cannister goes to only one truch (by the capacity of the edges between cannisters and trucks), and a truck can not carry more than k cannisters. Conversely, if there is a flow of value n, then there is a placement of cannisters to trucks, where each cannister i, goes to truck j, if there is one unit of flow along the edge c_i, r_j.
Running time is polynomial in the number of trucks and cannisters.

(b)
1. First of all, this problem is in NP, as given the placement of cannisters to trucks, we can easily check in polynomial time, that each truck carries only cannisters that can be brought together.
2. We will prove that this problem is NP-Complete by reducing from the $Graph\ 3-coloring$ problem.
Consider an instance of the $m-coloring$ problem, $m \geq 3$. A graph G with n nodes. The question is whether we can color the nodes of the graph with m different colors, such that no incident nodes have the same color. We will reduce this instance to our problem as follows: Each node v_i in the graph is a cannister. There is an edge between two cannisters, if

these cannisters can not be in the same truck. Also, each one of m colors, denotes a truck with capacity $k = n$.
3. We also need to argue that there is a solution to the "cannisters into trucks problem" if and only if there is a solution to the $m - coloring$ problem, for some $m \geq 3 \Leftarrow$ If there is a solution to our problem, a way to assign cannisters to the m trucks, there is also a solution to the $m - coloring$ problem, as we can color the nodes of the graph with the color of the truck every cannister goes into. $\Rightarrow$ Conversely, if there is a solution to the $m - coloring$ problem, then we can assign all cannisters (represented by nodes in the graph G) of the same color to one truck. In that way, no two cannisters that can not be brought together will end up in the same truck.
Hence, this problem is at least as difficult to the $3 - coloring$ problem, therefore $NP - Complete$.