CSE 40113: Algorithms

Homework 8

You may complete this homework in groups of 3 or less students. Note that the integrity policy applies: your group should write up your own work, and cite any sources used (including other students). If you have any questions, please re-read both the homework guidelines and the academic integrity policy carefully, and then come discuss any questions or concerns with me.

Required Problems

1. A website is trying to analyze the behavior of its customers from recent sales. They store prior purchase information in a 2-dimensional array X, where the columns correspond to products and the rows to customers; the entry $P[i][j]$ then specifies the number of items of product i that customer j has bought in the past year.

   For example:

	Alice	Bob	Carol	David
stickers	0	6	0	3
tshirts	2	3	0	0
hats	5	0	1	3
mugs	0	3	2	7
cards	3	0	1	0

   Now the store wishes to identify a set of customers that respectively purchase different things, to identify a good subset of customers to poll for feedback on as many distinct products as possible. In other words, we want to find a large set of the customers C such that no two customers in C have purchased the same product.

   This leads us to the *customer poll* problem: given the matrix P and a target value k, is there a subset of at least k customers such that no two have purchased the same product? Show that this problem is NP-Complete.

2. Let's take a look at a more complex resource scheduling problem. Consider a group of n asynchronous processes. However, these processes might need access to some subset of the m shared resources. At any given timestep, each process has a set of resources it would like to use. Each resource might be requested by many of the processes, but it can only actually be used by one at a time. If a process can gain access to all of its requested resources, it is *active*; otherwise, it is *blocked.*

   Your job is to allocate the resources to processes as effectively as possible; here, this means we would like to allocate so that as many processes as possible can be active. This leads to a decision version of the question: given a set of n processes, m resources, and a list of requested resources for each of the processes (so a set of n lists, each of length $\leq m$), and a value k, is it possible to allocate resources to processes so that at least k are active?

   (a) Show that this problem is NP-Hard.

(b) Suppose now that there are only two types of resources, and each process requires exactly one resource of each type. (Example: say every resource is either a person or a piece of equipment, and each process requires one specific person and one specific piece of equipment from the list.) Show that this version of the problem can be solved in polynomial time (by giving an algorithm).

3. In my excitement over the planned Mandalorian movie, I have been hunting for algorithms problems while re-watching far too much Star Wars. To my delight, I have found several!

Consider the faced by the Rebel Alliance as they fly from the Death Star back to the secret base on Degobah. We can view the galaxy as an undirected graph $G = (V, E)$, where each node is a start system and each edge $\{u, v\}$ indicates you can travel between u and v. The Death Star is represented by a node s, the hidden Rebel base by a node t. Certain edges have longer distances than others; thus we will give each edge an integer length $l_e \geq 0$. Also, certain edges represent routes that are more heavily patrolled by evil Imperial spacecraft; so each edge e also gets an integer risk $r_e \leq 0$, indicating the expended amount of damage incurred from the special-effects-intensive space battles if you use this edge.

There is a tradeoff here: it would be safest to travel through the outer rim of the galaxy, from one quiet far away star system to another, but then the ship would likely run out of fuel long before getting to its destination. (After all, they are on the run, so stopping to refuel should be avoided!) Alternatively, it would be fasted to dive through the cosmopolitan core of the galaxy, but then there would be far too many Imperial spacecraft to deal with. In general, for any path from s to t, we get both a length (the sum of all the lengths of its edges), and a total risk (the sum of all the risks of its edges).

So Luke, Leia, and company are looking at a complex shortest path type problem in this graph: they want to get from s to t along a path whose total length and risk are *both* reasonably small. In concrete terms, we will phrase this as the *Galactic Shortest-Path Problem* as follows: Given a setup as above and integer bounds L and R, is there a path from s to t whose total length is at most L and whose total risk is at most R?

Show that Galactic Shortest Paths is NP-Complete.