Final exam practice problems

Problems

1. Planning for our next season of political advertising, a particular candidate has hired you to advise them on how to best spend their advertising budget. The candidate wants a combination of online, radio and television ads that maximize total impact, subject to his budgetary constraints of at most \$1,000,000 to spend overall, and available airtime and print space.

   You get the following table of information, where impact is the estimated number of people that will view that ad if it runs:

type	impact per ad	cost per ad	max ads per week
online	10,000	10,000	25
radio	20,000	70,000	20
tv	50,000	110,000	15

   In addition, the radio and television ads are run by the same company, who will only allow a total of 25 ads of those two types to run in total.

   (a) Design a linear program in canonical form to determine the best combination of ads for the campaign. (Note: you don't need to solve this, and yes, I'm allowing fractional ads! Remember, Integer Linear Programming is hard.)

   (b) Give the dual formation of the linear program you wrote in part (a).

2. Let $A[1..n]$ be an array obtained by taking a sorted array of distinct numbers and rotating it by an unknown amount k (where $1 \leq k \leq n$).

   Example: [1,2,3,4,5,6,7] $\rightarrow [4, 5, 6, 7, 1, 2, 3]$

   Your goal is to find the index of the minimum element. You may perform comparisons of the form: "Is $A[i] < A[j]$?"

   Give an algorithm along with a matching lower bound for this problem.

3. You just discovered your best friend from elementary school on Twitbook. You both want to meet as soon as possible, but you live in two different cites that are far apart. To minimize travel time, you agree to meet at an intermediate city, and then you simultaneously hop in your cars and start driving toward each other. But where exactly should you meet?

   You are given a weighted graph $G = (V, E)$, where the vertices V represent cities and the edges E represent roads that directly connect cities. Each edge e has a weight $w(e)$ equal to the time required to travel between the two cities. You are also given a vertex p, representing your starting location, and a vertex q, representing your friend's starting location. Describe and analyze an algorithm to find the target vertex t that allows you and your friend to meet as soon as possible, assuming both of you leave home right now.

4. Let T be a full binary tree, meaning that every node has either two children or no children. For this problem, you can assume that you are given a root node r, and then every vertex v can access its left and right children, as well as its parent, if they exist. (You're welcome to use either v.left or $v \rightarrow$ left for your notation, as you prefer!)

- Recall that the height of a vertex v in T is the length of the longest path in T from v down to a leaf. In particular, every leaf of T has height zero.
- A vertex v is *AVL-balanced* if v is a leaf, or if the heights of v's children differ by at most 1. (You might recall from data structures that an AVL-tree is a binary search tree in which every vertex is AVL-balanced, but that's not what this problem is about.)

Describe and analyze an algorithm to compute the number of AVL-balanced vertices in T.

5. Give an example of a network that has a unique maximum s to t flow, but does *not* contain a unique minimum (s,t)-cut.

6. The University has hired you to write an algorithm to schedule final exams. Each semester, the school offers n different classes. There are r different rooms on campus and t different time slots in which exams can be offered. You are given two arrays $E[1..n]$ and $S[1..r]$, where $E[i]$ is the number of students enrolled in the i^{th} class, and $S[j]$ is the number of seats in the j^{th} room. At most one final exam can be held in each room during each time slot. Class i can hold its final exam in room j only if $E[i] < S[j]$. Describe and analyze an efficient algorithm to assign a room and a time slot to each class (or report correctly that no such assignment is possible).

7. Ad-hoc networks are made up of low-powered wireless devices. In principle, these networks can be used on battlefields, in regions that have recently suffered from natural disasters, and in other hard-to-reach areas. The idea is that a large collection of cheap, simple devices could be distributed through the area of interest (for example, by dropping them from an airplane); the devices would then automatically configure themselves into a functioning wireless network. These devices can communicate only within a limited range. We assume all the devices are identical; there is a distance D such that two devices can communicate if and only if the distance between them is at most D.

   We would like our ad-hoc network to be reliable, but because the devices are cheap and low-powered, they frequently fail. If a device detects that it is likely to fail, it should transmit its information to some other backup device within its communication range. We require each device x to have k potential backup devices, all within distance D of x; we call these k devices the backup set of x. Also, we do not want any device to be in the backup set of too many other devices; otherwise, a single failure might affect a large fraction of the network.

   So suppose we are given the communication radius D, parameters b and k, and an array $d[1..n][1..n]$ of distances, where $d[i,j]$ is the distance between device i and device j. Describe an algorithm that either computes a backup set of size k for each of the n devices, such that no device appears in more than b backup sets, or reports (correctly) that no good collection of backup sets exists.

8. You have a collection of n lockboxes and m gold keys. Each key unlocks at most one box. Without a matching key, the only way to open a box is to smash it with a hammer. Your baby brother has locked all your keys inside the boxes! Luckily, you know which keys (if any) are inside each box; this is given to you in the form of $box[1 \ldots n]$, where each $box[i]$ is a list of the keys saved in that box. You may assume that each box stores at least one key, and that $m \geq n$.

   (a) Your baby brother has found the hammer and is eagerly eyeing one of the boxes. Describe and analyze an algorithm to determine if it is possible to retrieve all the keys without smashing any box except the one your brother has chosen.

   (b) Describe and analyze an algorithm to compute the minimum number of boxes that must be smashed to retrieve all the keys.

9. A graph is *tonian* if there is a cycle in the graph that visits at least half of the vertices. Show that decided if a graph G is tonian is NP-Complete.

10. Let $G = (V, E)$ be a graph. A dominating set in G is a subset S of the vertices such that every vertex in G is either in S or adjacent to a vertex in S. The *dominating set* problem asks, given a graph G and an integer k as input, whether G contains a dominating set of size k. Prove that this problem is NP-hard.

11. Consider the following problem: You are managing a communication network, modeled by a directed graph $G = (V, E)$. There are c users who are interested in making use of this network. User i (for $i \in \{1, \ldots c\}$) issues a request to reserve a specific path P_i in G on which to transmit data. You are interested in accepting as many of these path requests as possible, subject to the following restrictino: if you accept both P_i and P_j, then the two paths cannot share any nodes.

    This defines the *path selection problem*: Given a direct graph $G = (V, E)$, a set of requests $P_1, \ldots, P_c$, each of which is a path in G, and a number k, is it possible to select at least k of the paths so that no two of the paths share any nodes?

    Prove that path selection is NP-Complete.