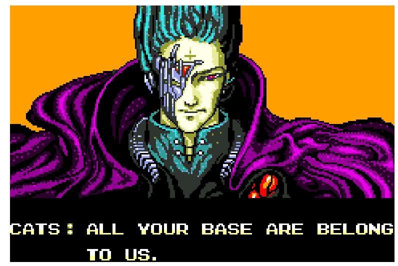
EE360C: Algorithms
University of Texas at Austin
Dr. David Soloveichik

Homework #8
Due: November 29, 2018 (*in-class quiz*)

Homework #8

You do not need to turn in these problems. The goal is to reinforce what we learned in class, as well as to cover material we didn't have time to cover in class. The in-class quiz that will cover the same or similar problems. Material on homework can also appear on exams.



Problem 1: All your base are belong to us

NOTE: Problem 1 is part of Programming Assignment 3. Consequently, there will NOT be a solution posted for Problem 1 with the rest of this homework.

You are on the planning committee of a major telecommunication company and you are given the task of setting up k base station antennas in a town with n houses along a line. Being an electrical engineer, you know that the further an antenna is from the house it covers, the more power it wastes. You'd like to minimize the maximum distance that your antennas need to cover, ensuring that you do not want to leave any house uncovered.

More precisely, you are given a set X of n houses sorted by position along a line, i.e, House 1 is at x_1, House 2 is at x_2 and so on with $x_1 < \cdots < x_n \in \mathbb{R}$. You are also given an integer k, the number of antennas. You need to find the set of base station positions, C, of k points $c_1, ..., c_k \in \mathbb{R}$ that minimizes the antenna range. The antenna range is defined as the minimum distance r such that every x_i is at most r from some c_j. Note that all base stations are identical and cover the same distance. In other words, the antenna range is:

$$r = \max_{1 \leq i \leq n} \left(\min_{1 \leq j \leq k} |x_i - c_j| \right)$$

where $\min_{1 \leq j \leq k} |x_i - c_j|$ represents the distance from point x_i to the closest base station, and thus a lower bound on the antenna range required by that base station, and the max condition identifies the required antenna range—the largest distance between any house and its closest base station.

Hint: The problem might make you scratch your head, but it isn't as difficult as it seems! Think of the possible subproblems that could be used. You could construct a $n \times k$ table indexed $r[t, j]$ that stores the optimal antenna range for the subproblems and another indexed $c[t, j]$ that stores a set of base stations for the subproblems. What can these sub-problems represent?

Problem 2: In it to Win it

Consider a row of n coins of values $V(1)\cdots V(n)$ where n is even. We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. We consider two ways that the opponent can behave:

(a) Assume the opponent is greedy and always simply chooses the larger of the two coins.

(b) Assume the opponent "plays optimally" and chooses the coin which maximizes the amount of money they can win.

Give a dynamic programming algorithm for cases (a) and (b) to determine the maximum possible amount of money we can definitely win (for the entire game) if we move first.

(Hint: Let $W[i,j]$ be the maximum value we can definitely win if it is our turn and only coins $i\cdots j$, with values $V(i)\cdots V(j)$, remain).

Problem 3: Natural Disasters

Consider the following scenario. Due to a large-scale natural disaster in a region, a group of paramedics have identified a set of n injured people distributed across the region who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hours driving time of their current location (so different people will have different options for hospitals, depending on their locations). At the same time, we dont want to overload any one of the hospitals by sending it too many patients. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the injured people in such a way that the load on the hospitals is balanced. i.e Each hospital receives at most $[n/k]$ people. Given the information about the people's locations, and we want to determine whether this is possible. Describe how this problem can be framed as a network flow problem and how the Ford-Fulkerson algorithm can be used to solve it.

Problem 4: Deleting Edges

Consider the following problem. You are given a flow network with unit capacity edges: It consists of a directed graph $G=(V,E)$, a source $s\in V$, and a sink $t\in V$; and $c_e=1$ for all $e\in E$. You are also given a parameter k.

The goal is to delete k edges so as to reduce the maximum s-t flow in G by as much as possible. In other words, you should find a set of edges $F\subseteq E$ so that $|F|=k$ and the maximum s-t flow in $G'=(V,E-F)$ is as small as possible subject to this.

Give a polynomial time algorithm to solve this problem. Argue (prove) that your algorithm does in fact find the graph with the smallest maximum flow.