

Algorithms Homework 7*

Fundamental techniques: Dynamic programming and greedy algorithms

Reading: GT Chapter 5

1. Suppose you are in charge of planning a party for Bowdoin College. The college has a hierarchical structure, which forms a tree rooted at President Mills. On the very last level are the faculty, grouped by department. (I have such a chart in my office, come by if you want to take a look). Each faculty has “underneath” all students taking a class with him/her that particular semester. Assume that every person is listed at the highest possible position in the tree and there are no double affiliations (everybody has one and only one supervisor in this hierarchy and no student is in more than one class).

You have access to a secret database which ranks each faculty/staff/student with a conviviality rating (a real number, which can be negative if the person is really difficult or boring). In order to make the party fun for everybody, President Mills does not want both a faculty/staff/student and his or her immediate “supervisor” to attend.

You are given a tree that describes the structure of Bowdoin College. Each node also holds a name and a conviviality ranking. Describe an algorithm to make up a guest list that maximizes the sum of the conviviality rankings of the guests. Analyze the running time of your algorithm.

2. (*text adapted from B. Thom, Harvey Mudd College*) You’ve decided to become a ski bum, and hooked up with Sugarloaf Ski Resort. They’ll let you ski for free all winter, in exchange for helping their ski rental shop with an algorithm to assign skis to skiers.

Ideally, each skier should obtain a pair of skis whose height matches his or her own height exactly. Unfortunately, this is generally not possible. We define the disparity between a skier and his/her skis as the absolute value of the difference between the height of the skier and the height of the skis. The objective is to find an assignment of skis to skiers that minimizes the sum of the disparities.

(a) First, let’s assume that there are n skiers and n skis. Consider the following algorithm: consider all possible assignments of skis to skiers; for each one, compute the sum of the disparities; select the one that minimizes the total disparity. How much

*Collaboration is allowed, even encouraged, provided that the names of the collaborators are listed along with the solutions. Write up the solutions on your own.

time would this algorithm take on a 1GHz computer, if there were 50 skiers and 50 skis?

(b) One day, while waiting for the lift, you make an interesting discovery: if we have a short person and a tall person, it would never be better to give to the shorter person a taller pair of skis than were given to the taller person. Prove that this is always true.

(c) Describe a greedy algorithm to assign the skis to skiers, and argue why this algorithm is correct. What is the time complexity of your algorithm?

(d) Your next task is to design an efficient dynamic programming algorithm for the more general case where there are m skiers and n pairs of skis ($n \geq m$).

Here is some notation that may help you. Sort the skiers and skis by increasing height. Let h_i denote the height of the i th skier in sorted order, and s_j denote the height of the j th pair of skis in sorted order. Let $\text{OPT}(i, j)$ be the optimal cost (disparity) for matching the first i skiers with skis from the set $\{1, 2, \dots, j\}$. The solution we seek is simply $\text{OPT}(m, n)$. Define $\text{OPT}(i, j)$ recursively.

(e) Now describe a dynamic programming algorithm for the problem. What is the running time of your algorithm? Explain.

(f) Briefly describe how your algorithm can be modified to allow you to find an actual optimal assignment (rather than just the cost) of skis to skiers. How does this affect the running time of your algorithm?

(g) Illustrate your algorithm by explicitly filling out the table $\text{OPT}(i, j)$ for the following sample data: ski heights 1, 2, 5, 7, 13, 21. Skier heights 3, 4, 7, 11, 18.