CS 577: Introduction to Algorithms — Homework 3

Out: 02/10/17 — **Due: 02/17/17**

Ground Rules

- Review problems will be discussed at the following week's review.
- Self-graded problems will be discussed at the following week's review, and are to be self-graded by you during the review. Only students present at the review session and turning in their graded solutions in person will earn points.
- Graded problems are to be turned in on the due date at the beginning of the lecture.
- Self-graded problems should be done individually.
- Graded problems should be done in pairs.
- Please follow page limits for all self-graded and graded problems.
- Write your name(s) clearly on your submissions, and turn in each problem on **separate sheets of paper**.

Review Problems

1. You are in charge of planning the bus route for a single bus for your local school district. The school and all of the bus stops on the route are located at exits along an east-west highway, and the bus route begins at the school. Think of the highway as a straight line with the school somewhere in the middle. You know how many students will get off of the bus at each exit. The bus may drive along the highway in either direction, switching directions as often as necessary. Your goal is to design a route such that the sum over the children of the total distance each child travels is minimized.

   For example, suppose there are two bus stops, one at a distance of 1 from the school in one direction along the highway with 2 children getting off, and the other at a distance of 2 in the other direction with 5 children getting off. If the bus route goes to the first stop and then the second, the total distance traveled by the kids is $1 \times 2 + (1 + 1 + 2) \times 5 = 22$. If the bus route goes to the second stop and then to the first, the total distance traveled is $2 \times 5 + (2 + 2 + 1) \times 2 = 20$. So the second route is preferable.

   Design a dynamic program to output the optimal bus route. Your input consists of a sorted list L of size n, where $L[i]$ specifies the distance of the ith exit from the east-most exit (so, in particular, the first distance in the list is 0). You are also given the location of the school, again in the form of its distance from the east-most exit. Finally, you are given a list S of size n, where $S[i]$ specifies the number of students to be dropped off at the ith exit in east to west order. Your algorithm should run in time polynomial in the number of exits.

2. In class, we developed an $O(nW)$ time algorithm for the Knapsack problem, where n is the number of items and W is the capacity of the knapsack. Recall that in this problem our goal is to fill the knapsack with items whose weights add up to no more than W and whose total value is maximized. Each item can only be picked once.

   Suppose that the capacity W is very large (say, exponential in n), so that the running time $O(nW)$ is prohibitively large. Suppose further that each item has a small integral value. Design a different dynamic program for this problem that runs in $O(nV)$ time, (that is, a running time independent of W,) where $V = \sum_i v_i$ is the sum of values of all the items. You may assume that integer operations take $O(1)$ time. Give a brief argument of correctness and an analysis of the running time of your algorithm.

   (Hint: you will need to modify the recursive definition we used to construct the $O(nW)$ time algorithm.)

Self-graded Problems

3. **(Page limit: 1 sheet; 2 sides)** Oh, No! You have been appointed as the organizer of Giggle, Inc.'s annual mandatory holiday party! The employees at Giggle are organized into a strict hierarchy, that is, a tree with the company president at the root. The all-knowing oracles in Human Resources have assigned a real number to each employee measuring how "fun" the employee is. In order to keep things social, there is one restriction on the guest list: an employee cannot attend the party if their immediate supervisor is also present. Give an algorithm that makes a guest list for the party that maximizes the sum of the "fun" ratings of the guests.

Graded Problems

4. **(Page limit: 2 sheet; 3 sides)**

   A primary task for a typesetter is to break up text into lines and insert white spaces between words in a way that the paragraph is fully justified, meaning that every line starts and ends at the same pixel on the page. Suppose that the typesetter is given a paragraph with n words, with the ith word containing p_i characters. Each word must fit fully on a line and cannot be broken across two lines. The typesetter determines how many words to place on each line. It then inserts white space between the words so that each line occupies space equivalent to exactly L characters. This means that if a line contains the ith through the jth words, then the typesetter needs to insert $L - \sum_{k=i}^{j} p_k - (j - i)$ spaces, where the extra term $j - i$ is due to the $j - i$ spaces already present between the words. The more the extra spaces, the worse the line looks. Accordingly, we say that the line has *slop* equal to $(L - j + i - \sum_{k=i}^{j} p_k)^3$. The slop of a paragraph is the sum over all of the lines, *except the last*, of the slop of each line.

   You are given as input a list of lengths of the n words, $\{p_1, \cdots, p_n\}$, and the desired line length L. You may assume that all of the lengths p_i are smaller than L. Design a dynamic program for the typesetter that outputs a paragraph with the minimum slop. (Note that you cannot change the order of the words, and no line can be longer than L characters.) Your algorithm should run in polynomial time in n and L. Prove the correctness of your algorithm and analyze its running time.

5. **(Page limit: 1 sheet; 2 sides)**

   A complex linear structure is to be assembled out of n smaller pieces. The pieces can be assembled in any order, with each step connecting the ith piece for some $i \in \{1, \cdots, n-1\}$ to the $(i + 1)$th piece. After each step, the newly constructed subpart must be tested. The cost of the testing depends on the subpart being tested. For example, if a step combines the subpart $\{i, \cdots, j\}$ with subpart $\{j + 1, \cdots, k\}$, it produces the subpart with pieces $\{i, \cdots, k\}$ and incurs a cost of $f(i, k) > 0$ for testing.

   Different assembly orders potentially have different total testing cost. For example, suppose that the structure consists of three pieces, and the cost of testing is given by: $f(1, 2) = 3$, $f(2, 3) = 1$, and $f(1, 3) = 5$. Then assembling the first and second pieces first and then joining them with the third has a total testing cost of $f(1, 2) + f(1, 3) = 8$, whereas assembling the second and third pieces first and then joining them with the first has a total testing cost of $f(2, 3) + f(1, 3) = 6$. Therefore, the second assembly order is preferable.

   Design an $O(n^3)$ algorithm to find an assembly order that incurs the least total testing cost. Give a brief argument of correctness, and analyze the running time.