CMPS 2200 Introduction to Algorithms – Fall 17

10/25/17

7. Homework

Due **11/1/17** at the beginning of class

Remember, you are allowed to turn in homeworks in groups of two. One writeup, with two names.

1. **Subsets of integers (8 points)**
   Given a positive integer S and an array $A[1..n]$ of n positive integers. The task is to decide whether there is a subset of integers in A that sum up to exactly S.

   (a) (2 points) Give a brute-force algorithm for this problem that runs in exponential time in n.

   (b) (3 points) Let $C[i, s]$ be true if there is a non-empty subset of integers in $A[1..i]$ which sum to s, and false otherwise. Develop a recursive formula for $C[i, s]$. You do not have to prove the correctness, but please justify your answer shortly.

   (c) (3 points) Use dynamic programming to solve the above problem using the recursive formula you have developed. What is the runtime of your algorithm in terms of n and S?

2. **Greedy Skis (5 points + 5 extra credit points)**
   Assume there are n people with heights $p_1, \ldots, p_n$ and there are n skis with heights $s_1, \ldots, s_n$. The task is to assign each person a ski in such a way that the average height difference between the person and their assigned ski is minimized. I.e., the task is to minimize

   $$\frac{1}{n} \sum_{i=1}^{n} |p_i - s_{f(i)}| ,$$

   where f is the assignment function.

   Consider the following two greedy algorithms:

   (a) Find the person and the ski with the minimum height difference. Assign this ski to this person. Repeat until every person has a ski.

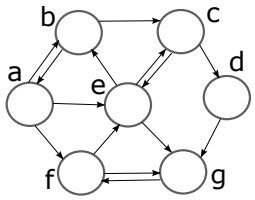
   (b) Sort the persons by height and sort the skis by height. Give the shortest person the shortest ski, and repeat.

   One of the algorithms above is correct, the other is incorrect. Find the one that is incorrect and disprove it with a counterexample (i.e., an example input where the greedy solution differs from the optimal solution).

   For the correct algorithm, you can get up to five extra credit points if you can provide a (partial) proof of its correctness.

FLIP OVER TO BACK PAGE ⟹

3. **Graph (10 points)**



(a) (4 points) Give the adjacency matrix representation and the adjacency lists representation for the graph above. Assume that vertices (e.g., in adjacency lists) are ordered alphabetically.

(b) (6 points) Run depth-first search on the graph above, starting on vertex a. Assume that vertices are ordered alphabetically in the adjacency lists. Write the discover and finish times into the vertices. Draw the tree edges into the graphs and show how the tree is stored in the predecessor array. In addition, show the edge classifications of all edges in the graph.