CS300, KAIST
Fall 2020
Homework 5
Due: 2020/11/20 (Fri) 17:59:99 KST

TA: Jinyoung Oh (jinyoungoh@kaist.ac.kr)

- You can write either in English or Korean.
- Please make sure your answer is readable.
- When you post a question on KLMS regarding this homework, please write in English and post publicly to help your classmates as well (of course as long as it does not include your own solution).

---

1. (40 points) There is a sequence S that consists of n integers, (i.e., $a_1, a_2, \cdots, a_n$). Given the sequence, you can pick any adjacent integers and multiply them to *synthesize* a new sequence. However, each integer could be multiplied with only one integer. For example, when a sequence is '$1, 1, 2, 3$', '$1, 1 \times 2, 3$' is a valid *synthesized* sequence but '$1 \times 1 \times 2, 3$' is not. With this multiplication, you need to maximize the sum of a sequence. For example, in case of '$1, 1, 2, 3$', the answer is '$1, 1, 2 \times 3$'. Give an **efficient algorithm to find such a *synthesized* sequence that maximizes the sum of its elements**, starting from a sequence S with n integers, (i.e., $a_1, a_2, \cdots, a_n$). You don't have to find all cases with the same (largest) sum.

   To get full credits from this problem, your algorithm should at least as efficient as TA's sample solution.

   (a) (5 points) What is the *synthesized* sequence that maximizes the sum of its elements, starting from $S := (1, 3, 2, 1, 2, 4, 1)$?

   (b) (5 points) Define the entries of your table for DP in words. You can use multiple tables if you wish. (e.g., $T(i, j, \cdots)$ is ... and $V(i, j, k)$ is ...)

   (c) (10 points) Define the relationship between entries in your table for DP, and briefly justify. (e.g., $T(i) = T(i-1)$ because I want it.)

   (d) (15 points) Write pseudo-code for your algorithm.

   (e) (5 points) Analyze time-complexity of the algorithm.

2. (60 points) A country named *the United States of Wonderland(USW)* uses a method similar to that of the United States of America to elect its president (indirect election). There are n states in the USW and each state i has p_i people who can vote for their preferred candidate. Each state selects a single candidate by Popular Vote (the candidate who gets the most votes wins in the state i). When a candidate wins in a state i, he or she can nominate e_i *electors* who pledged to vote for him or her at the final election. Unlike the USA, the USW is not democratic, so e_i is not proportional to p_i. The candidate who gets the most votes from the *electors* becomes the president (to make your life easier, let's assume there is no 'faithless elector'.)

   Currently, there are only two candidates in the election, Mr. Cake from Birthday Party and Ms. Wine from Surprise Party. You are working at Birthday Party, and your Party's candidate Mr. Cake wants to reduce the campaign cost. Obviously, the easiest way to reduce the cost should be focusing on **the smallest number of people who can make him win.** Give an algorithm to find out the smallest **total number of people** who can make him win in the election.

   To get full credits from this problem, your algorithm should at least as efficient as TA's sample solution.

   (a) (10 points) Define the entries of your table for DP in words. You can use multiple tables if you wish. (e.g., $T(i, j, \cdots)$ is ... and $V(i, j, k)$ is ...)

   (b) (15 points) Define the relationship between entries in your table for DP, and briefly justify. (e.g., $T(i) = T(i-1)$ because I want it.)

   (c) (30 points) Write pseudo-code for your algorithm.

   (d) (5 points) Analyze time-complexity of the algorithm.