CSE 431/531: Algorithm Analysis and Design Spring 2018

Homework 4

Instructor: Shi Li **Deadline: 4/22/2018**

Problems	1	2	3	Total Score
Max. Score	15	15	40	70
Your Score				

Collaboration Policy You are allowed to discuss the homework problems with classmates. However, it is highly recommended that you first think about each problem for enough time before the discussion. You must write your solutions by yourself, in your own words. You need to write down the names of the students you collaborated with. For the programming problem, you must implement the algorithm by yourself.

Problem 1(15 points) An independent set of a graph $G = (V, E)$ is a set $U \subseteq V$ of vertices such that there are no edges between vertices in U. Given a graph with node weights, the maximum-weight independent set problem asks for the independent set of a given graph with the maximum total weight. In general, this problem is very hard. Here we want to solve the problem on trees: given a tree with node weights, find the independent set of the tree with the maximum total weight. For example, the maximum-weight independent set of the tree in Figure 1 has weight 47.

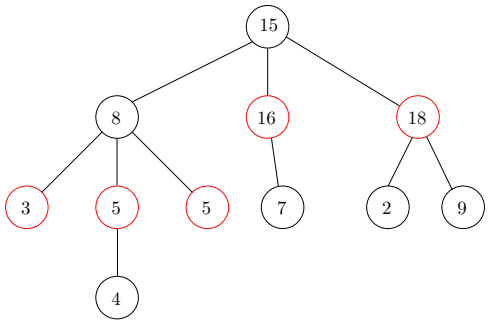

Figure 1: The maximum-weight indpendent set of the tree has weight 47. The red vertices give the independent set.

Design an $O(n)$-time algorithm for the problem, where n is the number of vertices in the tree. We assume that the nodes of the tree are $\{1, 2, 3, \cdots, n\}$. The tree is rooted at vertex 1, and for each vertex $i \in \{2, 3, \cdots, n\}$, the parent of i is a vertex $j < i$. In the input, we specify the weight w_i for each vertex $i \in \{1, 2, 3, \cdots, n\}$ and the parent of i for each $i \in \{2, 3, \cdots, n\}$.

Problem 2(15 points), Longest N-shape subsequence Given a sequence $A[1 \,..\, n]$ of numbers, we say that A is an N-shaped sequence if there are two indices i, j such that $1 < i < j < n$ and

- $A[1] < A[2] < A[3] < \cdots < A[i]$,
- $A[i] > A[i+1] > A[i+2] > \cdots > A[j]$,
- $A[j] < A[j+1] < A[j+2] < \cdots < A[n]$.

For example $(3, 6, 9, 12, 11, 10, 12, 13, 17)$ is an N-shaped sequence.

Design an $O(n^2)$-running time algorithm that, given an array A of n numbers, outputs the length of the longest N-shaped subsequence of A. (If no such subsequence exists, your algorithm can output $-\infty$). For example, if the input sequence is $(3, 1, 4, 6, 5, 7, 2)$, your algorithm should output 5 ($(3, 4, 6, 5, 7)$ is the longest N-shaped subsequence).

Problem 3(40 points) This is the programming problem of the homework. You need to implement the dynamic programming algorithm for the longest common subsequence problem. You can use C++, Java or Python to implement the algorithm.

Input You need to read the input from the console. It contains two lines, each containing one string. You can assume each string only contains upper and lower case letters and numbers; the length of each string is at most 1000.

Output You need to output to the console. The first line of the file is an integer indicating the length of the longest common subsequence between the two strings. The second line contains the longest common subsequence (which may not be unique).

Sample input (from the course slides)	**Output for sample input**
bacdca	4
adbcda	adca