University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Exercise Sheet 4

Due: Friday, 17th of November, 2023, 10:00 am

Remark: You are required to use the principle of dynamic programming in all of your algorithms. It is preferable if you write the algorithms in pseudocode and to write down the recursions.

Exercise 1: Leaf to Leaf *(10 Points)*

Given a rooted tree $T = (V, E)$ with n nodes such that each node is assigned a *unique* integer value. Note that we define a *rooted tree* to be a tree where one node has been designated the root.

(a) Find an algorithm that runs in $\mathcal{O}(n)$ and computes the maximum sum of the node values from the *root* to any of the leaves without re-visiting any node. Argue correctness and analyze its running time.

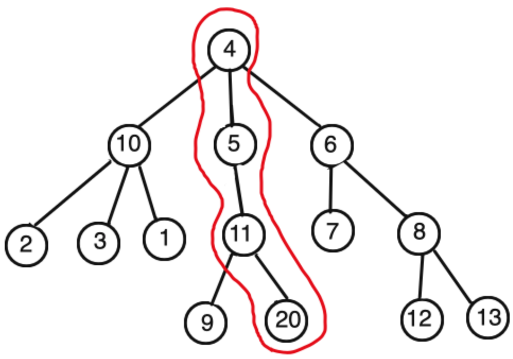

Figure 1: This is an example to our problem where the node with value 4 is the root. Thus, the path marked in red has the maximum sum of values of nodes in its path from the root to a leaf, hence your algorithm should output 4+5+11+20=40.

(4 Points)

(b) We generalize the problem and ask you to design an algorithm that runs in $\mathcal{O}(n)$ and calculates the maximum sum of the node values from *any leaf* to any other leaf without re-visiting any node. Argue correctness and analyze its running time. *(6 Points)*

Exercise 2: Mutually Involved Bitstrings *(10 Points)*

Consider the following bitstrings $A = a_1a_2...a_m$, $B = b_1b_2....b_n$, and $C = c_1c_2...c_{m+n}$. We say that A and B are *mutually involved* in C if C can be obtained by rearranging the bits in A and B in a way that maintains the left-to-right order of the bits in A and B.
For example $A = 010$ and $B = 10$ are mutually involved in $C = 01100$ but not in 00101.

Give an algorithm that runs in $\mathcal{O}(m \cdot n)$ and determines whether A and B are mutually involved in C or not. Argue correctness of your algorithm and analyze its running time.