University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger, P. Schneider

Algorithm Theory
Exercise Sheet 4

Due: Tuesday, 16th of November, 2021, 4 pm

Exercise 1: Bagging Marbles *(8 Points)*

We have n marbles and an array $A[1..n]$, where entry $A[i]$ equals the price of a bag with *exactly* i marbles. We want to distribute the n marbles into bags such that the profit, which is the total value of all bags (with at least one marble), is maximized.

(a) Give an efficient algorithm that uses the principle of dynamic programming to compute the maximum profit one can make. *(4 Points)*

(b) Argue why your algorithm is correct. Give a tight (asymptotic) upper bound for the running time of your algorithm and prove that it is an upper bound for your solution. *(4 Points)*

Exercise 2: Parenthesization *(12 Points)*

Consider a string B of $n \geq 3$ symbols over $\{0, 1, \wedge, \vee, \oplus\}$ which correspond to boolean true and false values and the logical operators and, or, xor, where B starts and ends with 0 or 1, and a 0 or 1 is always followed by one of the operators $\wedge, \vee, \oplus$ (unless it is the last symbol).

The goal is to count the number of possibilities you can put substrings of B *reasonably* into brackets, such that it evaluates to true (i.e., 1). Roughly speaking, by reasonable we mean that it is clear in which order to evaluate the operators of B and there are no unnecessary brackets.

Formally we define a *reasonable* parenthesization of such a string B (that does not have any brackets yet) recursively as follows. In the base case, if B has just one operator then there is no need for brackets, i.e., the only parenthesization of B that is reasonable is if there are *no* parenthesis.

If B has more than two operators we pick an operator o in B and define the substring either to o's left or to its right as B'. If $|B'| \geq 3$, we put the substring B' into brackets and recursively pick a reasonable parenthesization of B'. Then we pick a reasonable parenthesization of B where the substring (B') is replaced with a single symbol x (which we now consider as a boolean value 0,1). All parenthesizations of B that can be obtained this way are reasonable.[1]

(a) Let $B = 0 \oplus 1 \vee 0 \wedge 0 \vee 1$. Give all reasonable parenthesizations of B (as defined above) so that the resulting expression evaluates to true (it is sufficient to give their total number if you feel confident that it is correct). *(2 Points)*

(b) Give an efficient algorithm that uses the principle of dynamic programming to compute the *number* of reasonable parenthesizations of B that evaluate B to true. *(6 Points)*

(c) Argue why your algorithm is correct. Give a tight (asymptotic) upper bound for the running time of your algorithm and prove that it is an upper bound for your solution. *(4 Points)*

[1] Alternatively, we obtain the reasonable parenthesizations by splitting string B (with $|B| > 3$, otherwise use the base case) at some operator o into the substrings B_1, B_2, left and right of o, respectively. Then put each substring B_1, B_2 with at least 3 symbols into brackets and recursively determine reasonable parenthesizations.