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

Algorithms Theory
Exercise Sheet 4

Due: Tuesday, 1st of December 2020, 4 pm

Exercise 1: Number Picking Game *(10 Points)*

Consider an array $A[0..n-1]$ of positive integer numbers. Consider the following game. The goal is to pick numbers from A with a sum as large as possible. In the first turn you can pick *one* number $A[0]$ or $A[n-1]$ from either end of the array $A[0..n-1]$. In the second turn an adversary picks the first or last entry in the *remaining* array (either from $A[1..n-1]$ or $A[0..n-2]$). Generally speaking, each player picks in its turn either the number $A[i]$ or $A[j]$ from the remaining array $A[i..j]$.

(a) Assuming that both players play optimally (and know that their respective adversary does as well), give a *recursive function* $w(i,j)$ describing the *winnings* the player who picks next from one end of $A[i..j]$ can still make for the rest of the game. Do not forget the base cases. Explain the correctness of the recursion. *(5 Points)*

(b) We want to compute your *total winnings* provided that you are allowed to pick first and both you and your adversary always play optimally (and both knowing that). Use $w(i,j)$ and the principle of dynamic programming to provide pseudo code that computes your *total winnings* under these circumstances in $O(n^2)$ time. Explain the running time. *(5 Points)*

Exercise 2: Breaking Eggs *(10 Points)*

Imagine a building with n floors. We are given a supply of k eggs. For some reason we need to find out at which floor eggs start breaking when dropped from a window on that floor.

Suppose that dropping an egg from a certain floor always produces the same result, regardless of which egg is used and any other conditions. Initially, we do not have any knowledge at which height eggs might break. If an egg does not break, then it does not take any harm and can be fully reused.

If an egg breaks when dropped from a floor, it also breaks when dropped from higher floors. If eggs survive being dropped from a floor, they survive being dropped from lower floors. We call the floor from which dropped eggs break but survive at all floors beneath, the *critical floor*. The goal is to find the critical floor with minimum number of attempts (number of times eggs are being dropped).

(a) Suppose we have only one egg. Give a strategy to always find the critical floor, if it exists. *(1 Point)*

(b) We want some advance knowledge before starting to drop eggs. For inputs n and k, we want to compute the *exact* number of attempts $a(n,k)$ which an optimal strategy requires *in the worst case*, until it finds the critical floor (if it exists). Give a recursive relation for $a(n,k)$. Do not forget the base cases. Explain the correctness of the recursion. *(5 Points)*

(c) Give an algorithm that uses the principle of dynamic programming to compute $a(n,k)$ in $O(kn^2)$ time. Explain the running time. *(4 Points)*