Tutorial 5: Dynamic programming

1 Maximum subarray

In the *maximum subarray* problem, we are given as input an array $A[1], \ldots, A[n]$ of n integers. Our goal is to find the maximum value of any subarray of A. I.e., the valid solution is the value

$$\max_{i,j:1 \leq i \leq j \leq n} \sum_{\ell=i}^{j} A[\ell].$$

Design a dynamic programming algorithm that solves the maximum subarray problem.

2 Longest increasing subsequence

Given a sequence of non-negative integers $a_1, \ldots, a_n$, design an algorithm that computes the length of the longest subsequence of increasing numbers within the original sequence. For example, on instance

$$0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15$$

one of the longest increasing subsequences is $0, 4, 6, 9, 11, 15$ so the valid solution for this instance is 6. (Note that there are multiple increasing subsequences of length 6 for this example.)

Design an algorithm that solves the longest increasing subsequence in time $O(n^2)$.