

ADA 2022 Tutorial 3

February 10/11, 2022

This tutorial is more warmup on DPs. What we would like you to do for each of the problems is:-

1. Define the subproblems clearly
2. Write a recursion using the above definition and argue properly about why the recursion is correct (this is the optimal substructure property)
3. Implement using tables and argue runtime.

1 Maximum Sum Subarray

You are given an array of size n containing integers. Design a linear time algorithm to find the contiguous sub-array such that the sum of elements is as large as possible.

2 Maximum Monotonically non-decreasing Subsequence

You are given an array of size n containing integers. Design a linear time algorithm to find the sub-array (not necessarily contiguous) of maximum length such the values are non-decreasing.

3 Maximum Switching Subsequence

Suppose you are given an array of distinct real numbers, $A[1 : n]$. Define a switching subsequence of A as a sequence $A[k_1], A[k_2], \dots, A[k_\ell]$ such that ,

$$\begin{aligned} A[k_i] &< A[k_{i+1}], \text{ for odd } i \\ A[k_i] &> A[k_{i+1}], \text{ for even } i \end{aligned}$$

In plain English, the sequence switches between increasing and decreasing, starting with increasing. The goal is to find the longest switching sub-sequence of A . Design a linear time algorithm for this.

4 Recurrence using Substitution Method

$$T(n) = 2T(n/3) + T(n/2) + 5n$$