CS325: DP project

Your report must be typeset, printed and stapled. *Each team member's name must be listed as well as any resources used to finish the project.*

For this project, you will try to improve on your algorithms for the maximum subarray problem in the past project using *dynamic programming*. Remember the problem:

> Given array of small integers $a[1, \ldots, n]$, compute
>
> $$\max_{i \le j} \sum_{k=i}^{j} a[k]$$
>
> For example, MaxSubarray$([31, -41, \mathbf{59}, \mathbf{26}, \mathbf{-53}, \mathbf{58}, \mathbf{97}, -93, -23, 84]) = 187$

It is highly recommended that you implement this algorithm in the same language used for your divide & conquer algorithm in the last project. Be sure that your algorithm correctly finds the maximum subarray of an input array with all negative numbers.

Instructions

Your dynamic programming algorithm should be based on the following idea:

> The maximum subarray either uses the last element in the input array, or it doesn't.

Your tasks are:

Recursive function Describe the solution to the maximum subarray problem recursively and mathematically based on the above idea.

Pseudocode Give pseudocode for a *dynamic programming* algorithm based on this function. Your implementation should create a dynamic programming **table**.

Running time Analyze the time complexity of your algorithm.

Theoretical correctness Write a formal proof of correctness for your algorithm using induction.

Implement Implement your algorithm and include the *relevant section of code* in your project report. This should *not be lengthy*. Your implementation should only return the value (sum) of the maximum subarray and not the indices bounding the maximum subarray. I'm thinking 20 lines MAX.

Test You should make sure your program is correct, although you will not be submitting evidence (beyond the pseudocode *and* code above).

Compare Perform tests to compare this dynamic programming algorithm to the divide & conquer algorithm of the last project. You are in charge of deciding what tests would be reasonable. Discuss the comparative benefits and drawbacks of these two algorithms.

Your project report should include all of the above sections.