Solving the Max Subarray Problem via Divide-and-conquer

Lab 03, Spring 2020

Description In this lab assignment, your job is to implement the $O(n \log n)$ time divideand-conquer algorithm for the Max Subarray Problem; for the pseudo-code, see page 72 in the textbook or the lecture slides. Recall that in the problem, we are given as input an array $A[1 \cdots n]$ of n integers, and would like to find i^* and j^* $(1 \le i^* \le j^* \le n)$ such that $A[i^*] + A[i^* + 1] + a[i^*]$ $\cdots + A[j^*]$ is maximized.

Input structure The input starts with an integer number n, which indicates the array size. Then, the integers, $A[1], A[2], \dots, A[n]$, follow, one per line.

Output structure Output the sum of integers in the max subarray, i.e., $A[i^*] + A[i^* + 1] + A[i^* + 1]$ $\cdots + A[j^*].$

Examples of input and output:

Input

6

-3

11 -2

-3

10

-5

Output

16

Note that in this example, the max subarray is $A[2\cdots 5]$. So, we output $A[i^*] + \cdots + A[j^*] =$ 11-2-3+10=16. The output is only one number and has no white space.

See the lab guidelines for submission/grading, etc., which can be found in Files/Labs.