CSE100 Algorithm Design and Analysis — Lab 03, Spring 2017
Last update: 4pm, 1/30/2017 — Deadline: 11:59pm, 2/15/2017

Solving the Max Subarray Problem via divide-and-conquer

Description In this lab assignment, your job is to implement the $O(n \log n)$ time divide-and-conquer algorithm for the Max-Subarray-Problem; for the pseudo-code, see page 72 in the textbook. 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^* ($1 \leq i^* \leq j^* \leq n$) such that $A[i^*] + \cdots + A[j^*]$ is maximized.

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

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

Examples of input and output:

Input

```
6
-3
11
-2
-3
10
-5
```

Output

```
16
```

Note that in this example, the max subarray is $A[2 \cdots 5]$. (The array's starting and ending indices are 1 and 6, respectively.) 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.