Problem: Maximum subarray

Problem Definition:

Given an array A of numbers, find the contiguous subarray that has the largest sum

Example: for input A=(4,-5,6,7,8,-10,5,2), what is the solution?

Brute force solution?

Divide and Conquer: high level idea

- Partition the array A into two rough equal sized part A_L and A_R

  A_L=(4,-5,6,7) A_R =(,8,-10,5,2)

- Recursively compute the maximum subarray A_L and A_R for respectively

- How to combine? What run time to aim for if we want O(n log n) overall run time?

Disclaimer: D&C does not provide the most efficient algorithm for this problem. We will look at it later.

Example Problem 2

Input: an array A of sorted integers that have been shifted.

Goal: find the largest element in A

Example:

(40, 57, 89, 2, 8, 25, 30)

shifted 3 positions

Bruteforce?

Can we do better?

Goal: O(log n) run time

Q: how can we achieve O(log n) time?

A: Each recursion

- do some constant time operation
- shrink the input size by a constant factor.
  e.g. to n/2 like binary search

Rough direction:

(40, 57, 89, 2, 8, 25, 30)

- Identify the middle element,
- Do constant time operation on it
- Eliminate half of the array