

# Maximum Contiguous Subsequence

Kwankamol Nongpong, Ph.D.

CSX3009 Algorithm Design

Department of Computer Science

Assumption University

# Maximum Contiguous Subsequence

- Find the maximum sum of the subsequence/subarray.

2 -3 4 -1 -2 1 5 -3

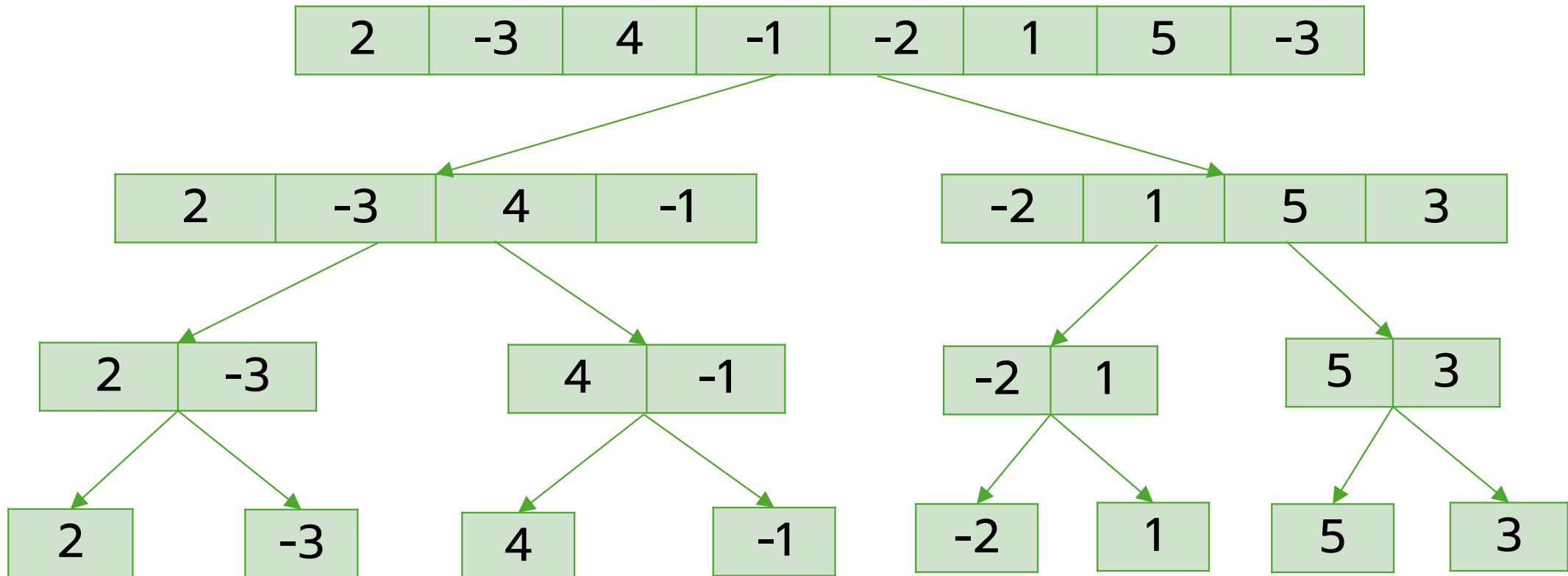
# What we have tried...

- Take 1: Brute-Force
- Take 2: Accumulation Technique
- Take 3: Kadane's Algorithm

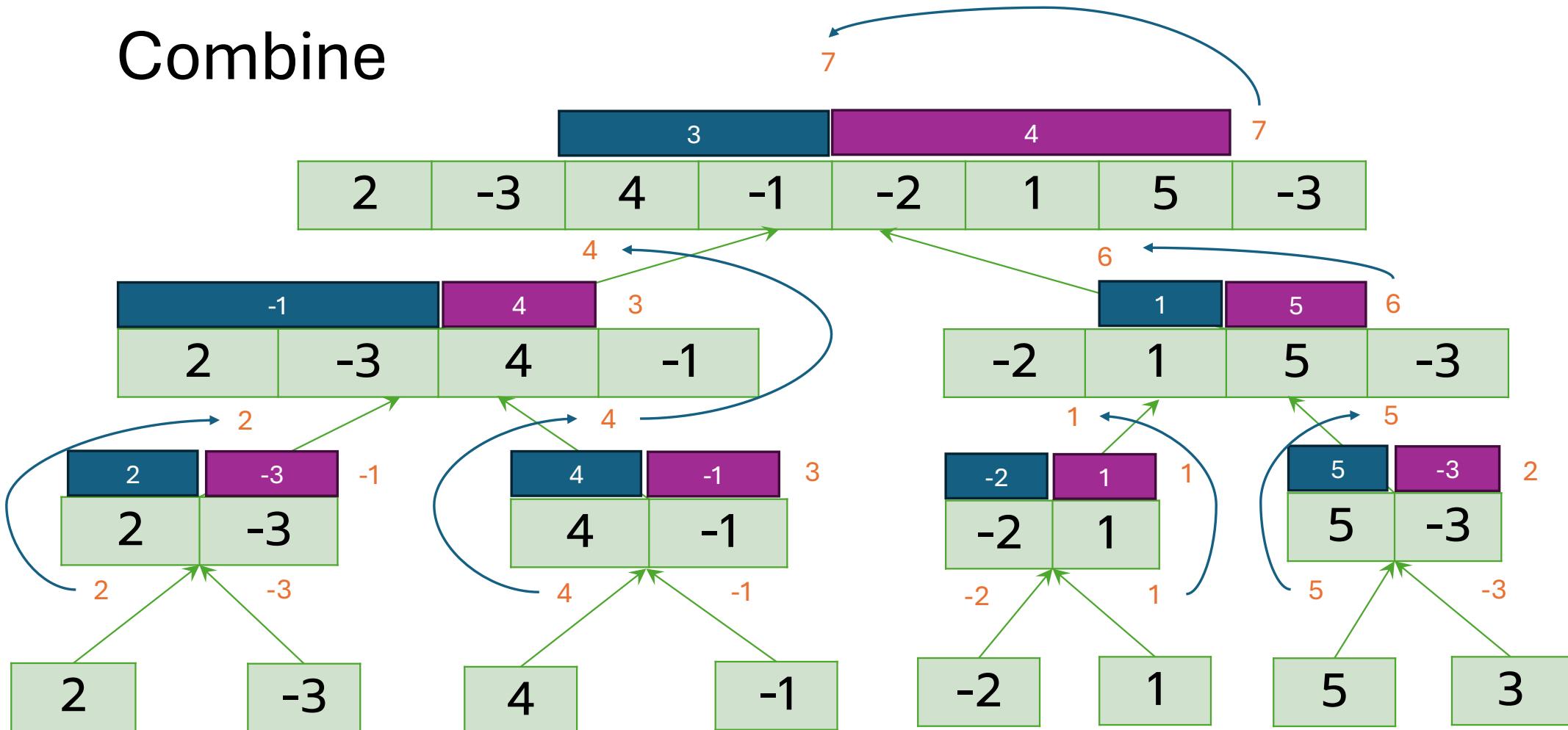
# Take 4: Divide and Conquer Approach

- Divide into two sublists.
- Consider the max of:
  - Max sum of the left sublist
  - Max sum of the right sublist
  - Max sum of the sublist that crosses the mid point
    - Add elements from mid to the left end and track the max sum
    - Add elements from mid+1 to the right end and track the max sum

# Divide



# Combine



# Implementation

- $\text{maxSubArraySum}(A, p, r)$  → Recursive function

- Base case: Your job
- Recursive case:
  - Calculate middle
  - Return the maximum of:
    - $\text{maxSubArraySum}(A, p, \text{middle})$
    - $\text{maxSubArraySum}(A, \text{middle} + 1, r)$
    - $\text{maxCrossingSum}(A, p, \text{middle}, r)$

- $\text{maxCrossingSum}(A, p, q, r)$

- Add elements from  $q$  to  $p$  (one by one) and update the max sum accordingly
- Add elements form  $q+1$  to  $r$  (one by one) and and update the max sum accordingly
- Return  $\text{maxLeftSum} + \text{maxRightSum}$

maxLeftSum

maxRightSum