

## Week 1

This problem is a greedy-able variant of subsequence sum. The greedy strategy is to select, for each same-sign segment, its largest value (i.e. for segment  $\{1, 3, 2\}$  select 3;  $\{-1, -3, -2\}$  select  $-1$ ). The sum of all max segments is the final result.

The (slightly) tricky part of the question is its implementation. It is not necessary to implement two distinct maxes to synchronize. We only need to add to the sum at each flip of sign, the added amount is the `max` of the segment. Then we reset the `max` to appropriate values (0 for positive, `INT_MIN` for negative; technically this could be done indiscriminantly of cases).

The runtime of the algorithm is linear  $O(n)$ , directly correlated to the input size.