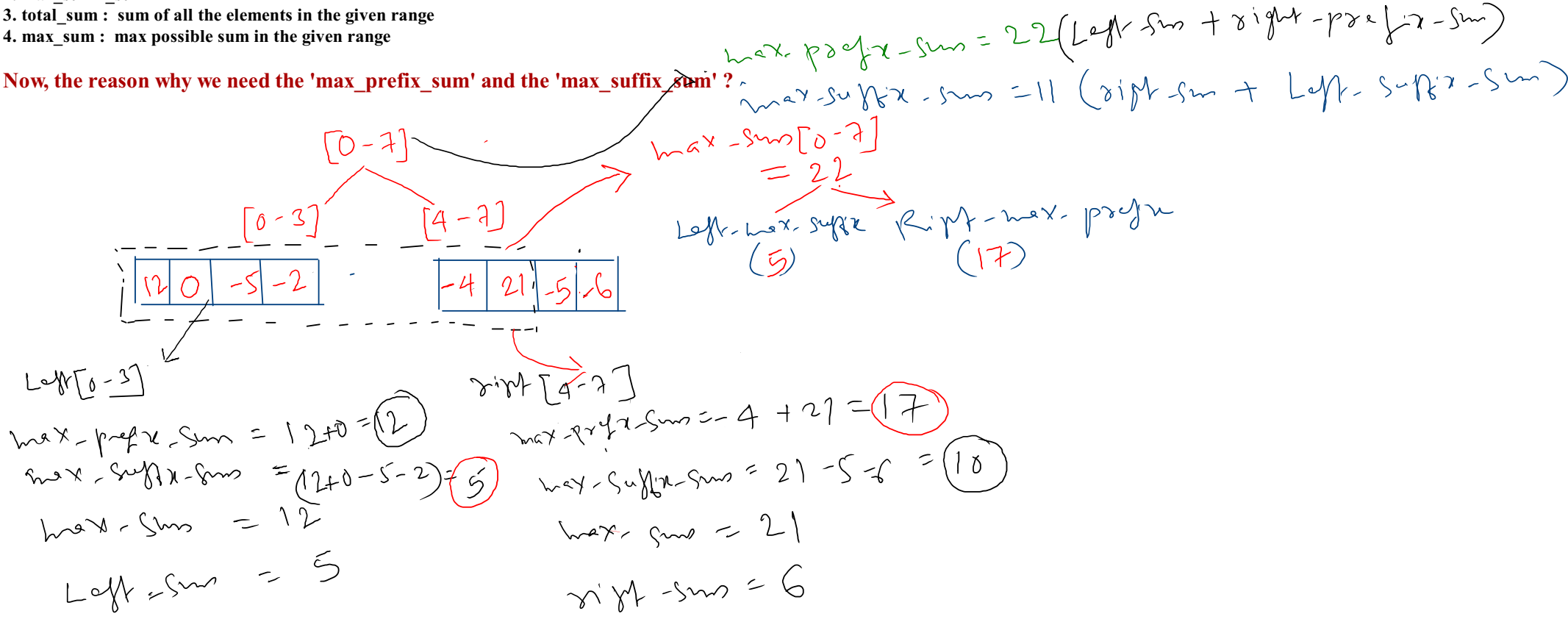
Finding subsegments with the maximal sum in a given range

The elements of the array can be negative, and the optimal subsegment can be empty (e.g. if all elements are negative).

To calculate the maximum sub-segment sum in a segment tree, we need the following data for every node:

1. **max_prefix_sum**
2. **max_suffix_sum**
3. **total_sum : sum of all the elements in the given range**
4. **max_sum : max possible sum in the given range**

Now, the reason why we need the 'max_prefix_sum' and the 'max_suffix_sum' ?



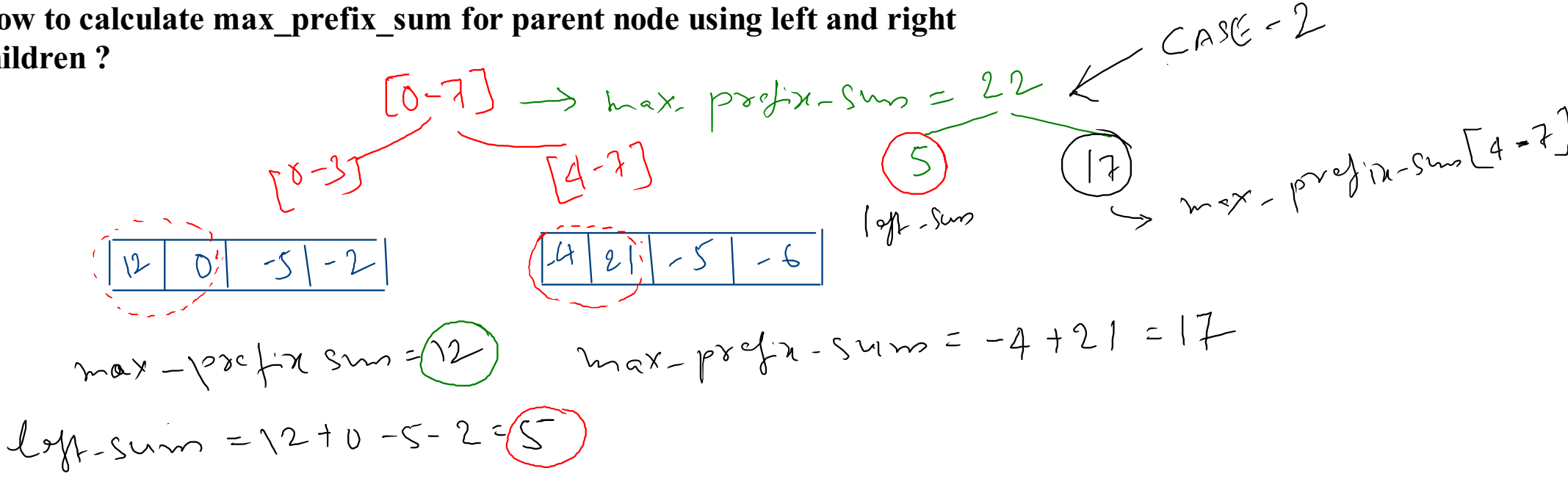
There are following 3 possible cases to have max_sum in range [0,7]:

CASE 1: when max_sum lies in left segment then max_sum of [0,3] is the max_sum of [0,7]

CASE2 : when max_sum lies in right segment then max_sum of [4,7] is the max_sum of [0,7]

CASE 3: When the max_sum spans across both the segment, then max_suffix_sum of [0,3] and the max_prefix_sum of [4,7] combined will give us the desired result.

How to calculate max_prefix_sum for parent node using left and right children ?

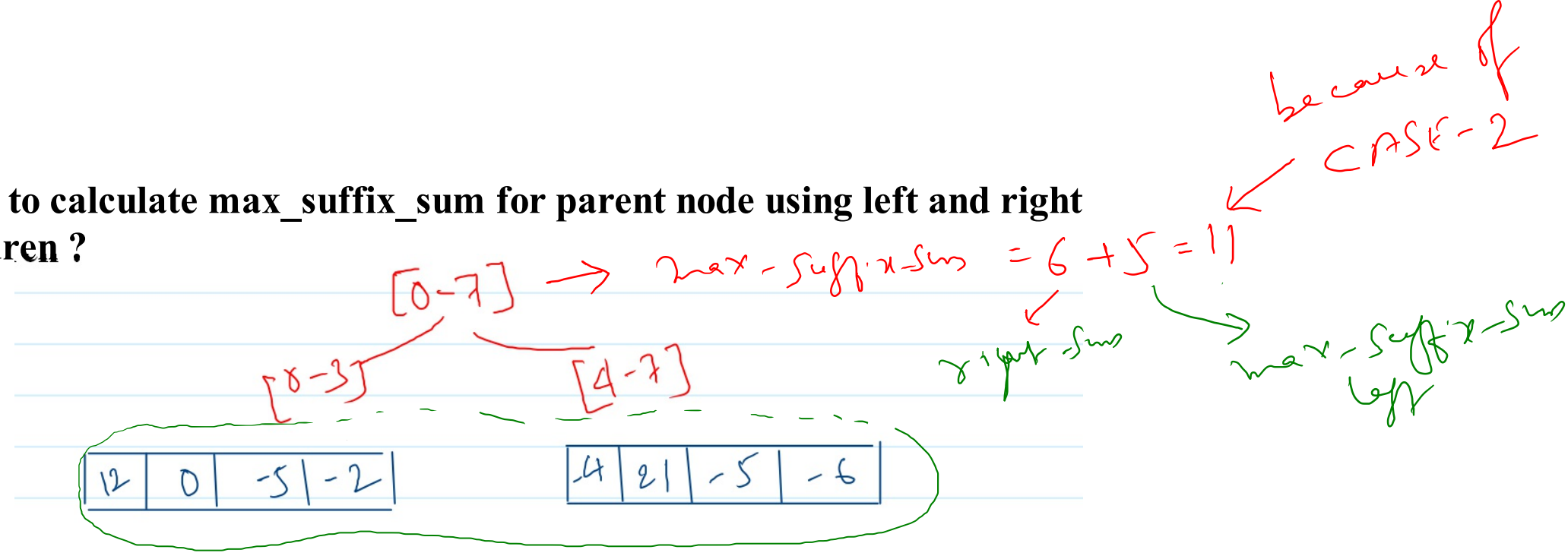


There are following two possible cases to calculate the max_prefix_sum of [0-7] using left[0-3] and right[4-7] segments

CASE 1: When max_prefix_sum lies in left segment then max_prefix_sum of [0-3] is the max_prefix_sum of [0-7]

CASE 2: When max_prefix_sum spans to right segment then the max_prefix_sum_[0-7] = total_sum_[0-3] + max_prefix_sum_[4-7]

How to calculate max_suffix_sum for parent node using left and right children ?



There are following two possible cases to calculate the max_suffix_sum of [0-7] using left[0-3] and right[4-7] segments

CASE 1: When max_suffix_sum lies in right segment then max_suffix_sum of [4-7] is the max_suffix_sum of [0-7]

CASE 2: When max_suffix_sum spans to left segment then the max_suffix_sum_[0-7] = total_sum_[4-7] + max_suffix_sum_[0-3]