COMP3121 Algorithms and Programming Tech 2022T3 assignment 1
Chen En (z5335039)

Question 4 Red & Yellow Flowers
[20 marks] You are given an array of n flowers that are either red or yellow. Your goal is to find the number of subarrays where there are more red flowers contained within the subarray, than there are yellow flowers outside the subarray.

4.1 [6 marks] Describe a method that runs in O(n) time, which pre-processes the input array such that you can then calculate the number of red flowers within any contiguous subarray in O(1) time.

We create an array R to store the numbers of red flowers, and the index i presents the current index which we count it to. (R[0] means the number of red flowers from index 0 to index 0, R[i] means the numbers of red flowers that we accumulate from index 0 to index i).

Check is the first element is red or yellow:
Case 1: If the first element is yellow flower, then R[0] = 0.
Case 2: If the first element is red flower, then R[0] = 1.

Calculating the R[i] (the numbers of red flowers that we accumulate from index 0 to index i.)

If the i-th flower is yellow, R[i] =R [i-1] + 0.
Otherwise if the i-th flower is red , R[i] = R[i-1] + 1.
There is base case when i = 0, we return R[i] since we can’t count R[0-1]. (The case is counting the number of red flowers from index 0 to index 0)

Time complexity
The time complexity we get accessed into the element in the array takes O(1). And since we create the array which traverses R from 0 to n-1, the time complexity is O(n). As the result, the final time complexity is O(n).

4.2 [14 marks] Design an algorithm that achieves your goal in O(n log n) time.
Let the cumulative number of red and yellow flowers be R[n] and Y[n] respectively.

Our goal is to find the number of index pairs {i, j} (0 <= i < j < n) such that R[j] - R[i] > Y[j] - Y[i].
The inequality above can be transformed into Y[j] - R[j] < Y[i] - R[i].

We can create a new array Diff[1 … n] = Y[1 … n] - R[1 … n], therefore the original goal can be rewritten to: Find the number of pairs {i, j} (0 <= i < j < n) such that Diff[j] < Diff[i]. That is, to find the number of inversion pairs of the array Diff, which can be solved by merge sort in O(nlogn).

Let count be the number of subarrays which contain more red flowers than yellow flowers.
For base case, if Diff[i] < 0 then increase count by 1.

Complexity
Since we apply the merge sort, the time complexity is O(nlogn).