Counting inversions problem

Bakh Khoussainov

Counting inversions

Let $a_1, a_2, \ldots, a_n$ be a list of integers.

Two indices i and j form an inversion if

(1) $i < j$

(2) $a_i > a_j$

For instance, $2, 4, 1, 3, 5$ has three inversions.

The problem

Input: A list A of integers

$$A = a_1, a_2, \ldots, a_n.$$

Output:

The number of inversions of the list A.

There is a "brute force" algorithm that solves the problem:

Initialize Count = 0.

For each (i, j) if i and j form an inversion increment Count.

There is a better way to solve the problem.

Idea:

Embed the Mergesort algorithm into our solution.

How can this be done?

On input $A = a_1, a_2, \dots a_n$

(1) Divide A into two equal sized lists X, Y.

(2) Count inversions in X And Y.

(3) Sort X AND Y

(4) Count inversions a_i and a_j such that $a_i \in X$ And $a_j \in Y$.

(5) Output the sum of the inversions

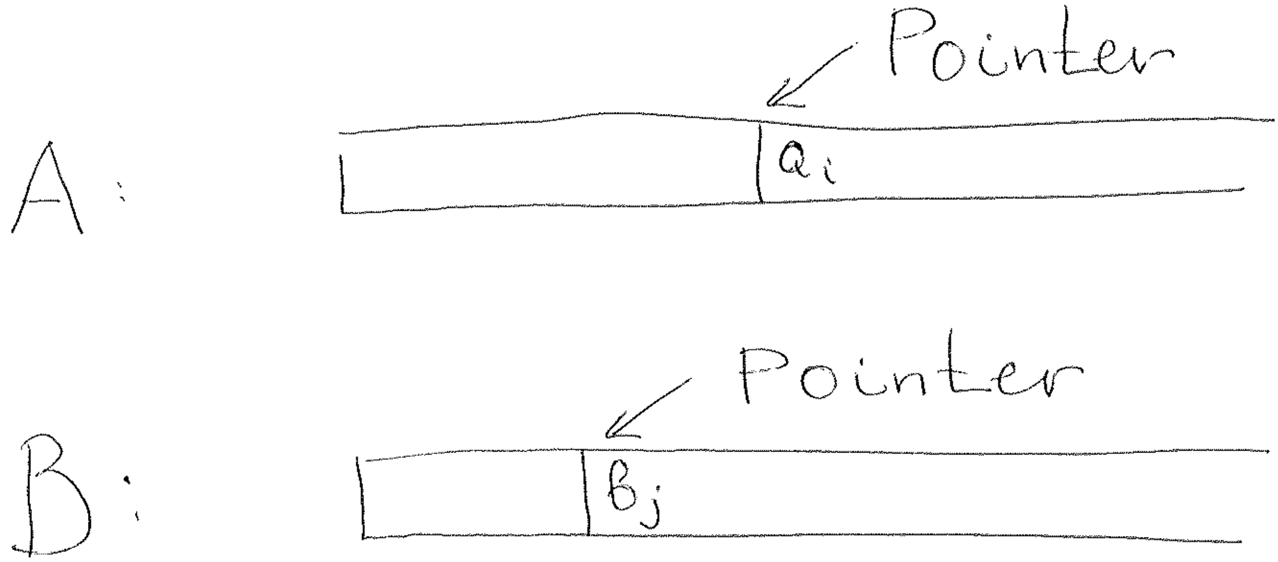
Merge-and-Count(A,B) algorithm.

Ingredients:

A, B sorted lists,

Counter

Current pointer



While both A, B nonempty

Append the smaller of a_i, b_j to new list C.

If $b_j < a_i$

Count $\leftarrow$ Count + the remaining items in A.

Advance the pointer in the list from which the smaller element was selected.

Sort-and-Count (L):

If $|L| \leq 1$, then there are no inversions.

Divide L into two equal halves: A and B.

$(r_A, A) \leftarrow$ Sort-and-count (A)

$(r_B, B) \leftarrow$ Sort-and-Count (B)

$(r, L) \leftarrow$ Merge-and-Count (A,B)

Output $r_A + r_B + r$.

Correctness is proved by induction on $|L|$.