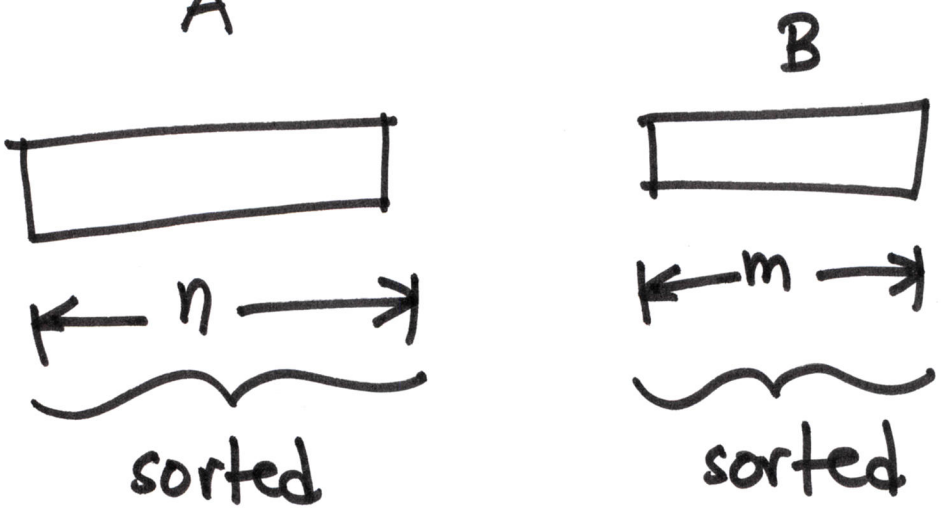
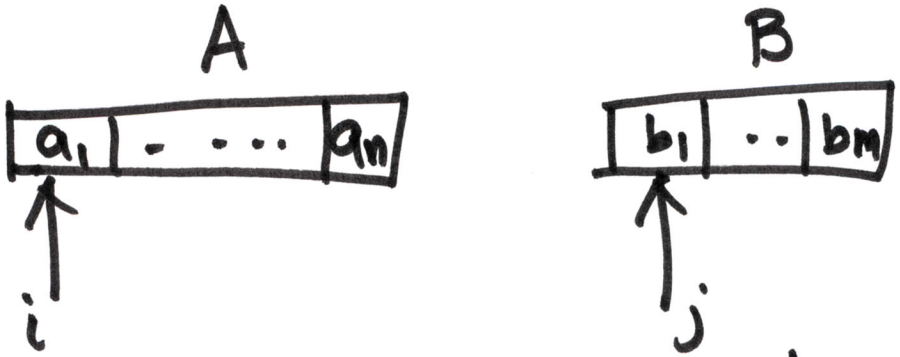
November 14, 2011

MERGE-COUNT

Input:



Ex.:



Case 1: $a_1 < b_1 < b_2 < \dots < b_m$

0 ~~inverse~~ inversions

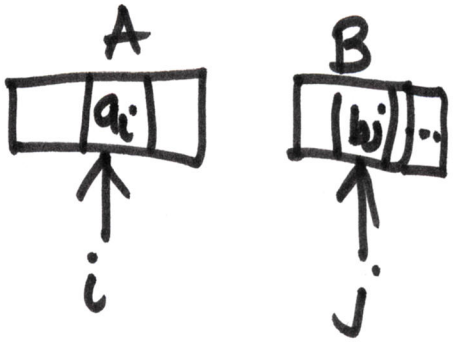
Add a_1 to C (output) i++

Case 2: $a_n > \dots a_3 > a_2 > a_1 > b_1$

n inversions!

MERGE-COUNT $(A, n; B, m)$

$O(n+m)$ time

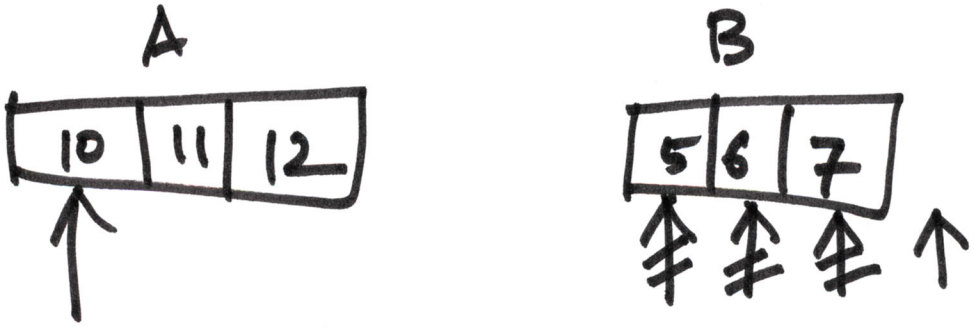


---

0. $i, j \leftarrow 1$
1. $inv \leftarrow 0$
2. While $i \leq n$ AND $j \leq m$
   If $a_i < b_j$
      Add a_i to C ; i++
   Else ($b_j < a_i$)
      inv += n-i+1
      Add b_j to C ; j++
3. If $i > n$ & $j \leq m$ append rest of B to C
   else if $j > m$ & $i \leq n$ ——— A to C
4. Output (inv, C)

$O(n+n)$ time

Ex.



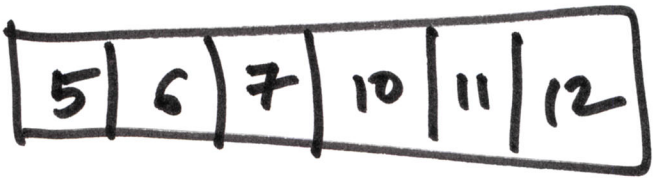
$a_1 > b_1 \rightarrow$ inv = 3

$a_1 > b_2 \rightarrow$ inv = 3+3 = 6

$a_1 > b_3 \rightarrow$ inv = 6+3 = 9

Output = (9,

C



)

Ex.
Proof of correctness

INPUT: $(P_1, \ldots, P_n)$ $\quad P_i = (x_i, y_i)$

OUTPUT: p, q s.t $d(p,q)$ is minimized

$$d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Assumption:

All x co-ords are distinct

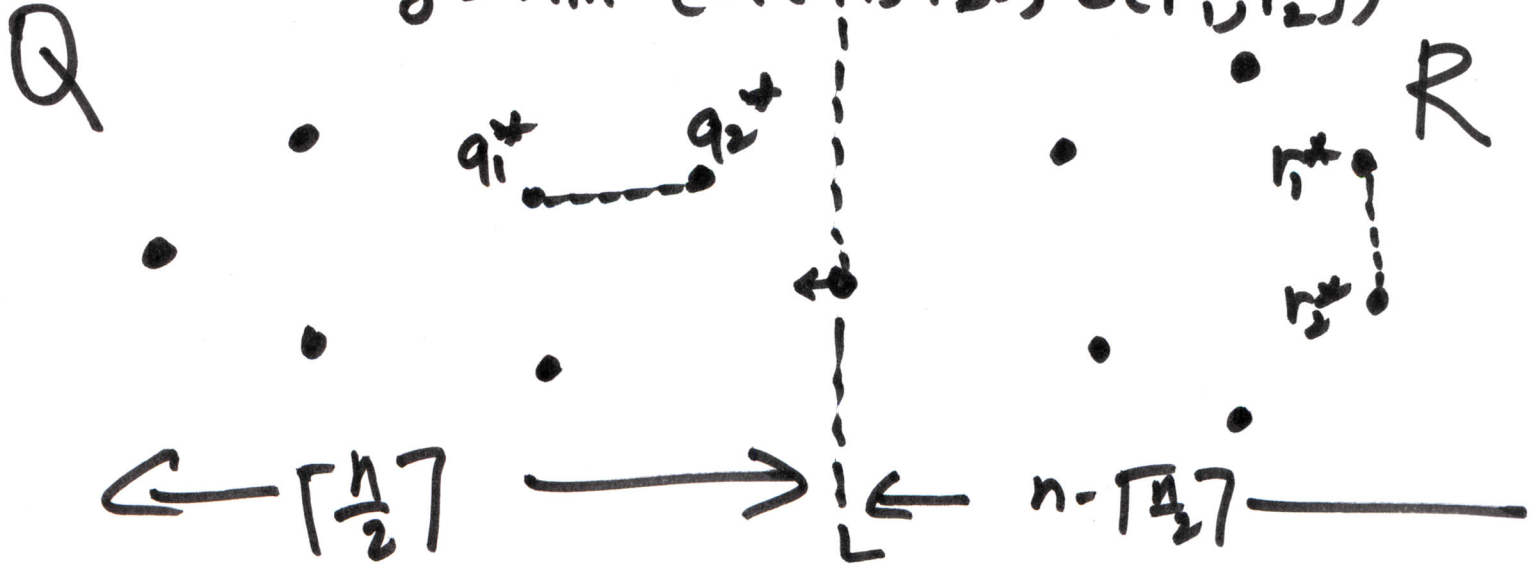
— y —

(can get this by rotating n pts)

Divide & Conquer algo

$n = 13$

$\delta = \min(d(q_1^*, q_2^*), d(r_1^*, r_2^*))$



from $P \longrightarrow P_x$, P_y

P_x: P sorted by x-coord

P_y: sorted by y-co-ord

$\rightarrow O(n \log n)$ time

$\rightsquigarrow$ Figure out L from P_x