CS 314 - Divide + Conquer

Note Title 1/29/2010

Announcements

- Oral grading for HW 2 this week
- No office hours next Thursday (2/11)
  (sorry - doctor's appointment)

Recap: Merge sort

How does it work:
class demo!

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$$= O(n \log n)$$

Quicksort: Worst case $T(n) = T(n-1) + O(n)$
$= O(n^2)$

Counting Inversions (Section 5.3)

Application: 2 people list a ranking of n things (of movies, books, etc.)

My list:	Bryan
Buffy the Vampire Slayer	Lost
Doctor Who	Buffy
Veronica Mars	Legend of the Seeker
Lost	Doctor Who
Battlestar Galactica	BSG
Vampire Diaries	Gray's Anatomy
Gray's Anatomy	Vampire Diaries
Legend of the Seeker	Veronica Mars

How can we compare?

Idea: Count the number of pairs which are ordered one way in my list + another in Bryan's.

We call this counting inversions.

Ex: Sequences: 1, 2, 3, 4, 5

2, 4, 1, 3, 5

How many inversions?

3 inversions

How many inversions are possible? (# are 1 to n)

$n \quad n-1 \quad n-2 \; \dots \quad 1$

$$O(n^2) = (n-1)+(n-2)+(n-3)+\dots+1 = \frac{n(n-1)}{2}$$

How could we do this?

Find n, & count # inverted pairs involving n.

Find $n-1$, " " " using $< n-1$,

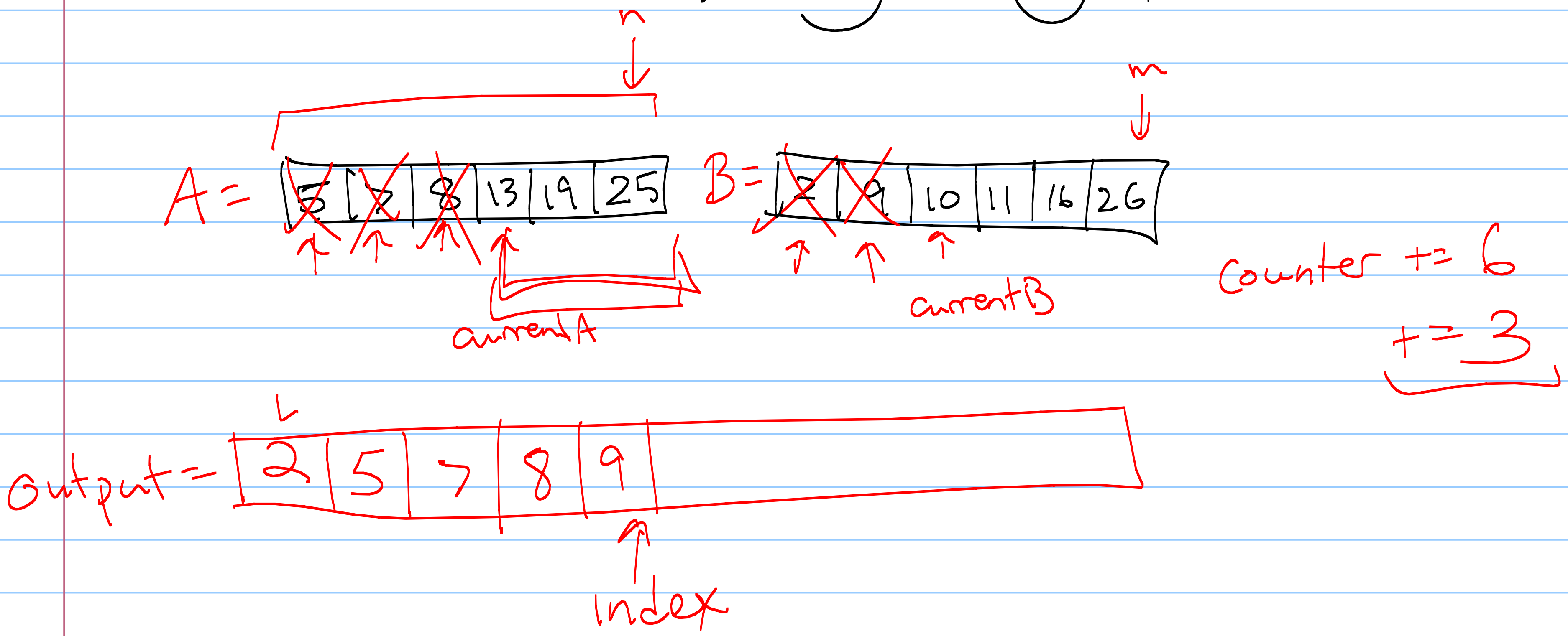
⋮

For every i, find # inverted pairs with i.

How many inverted pairs? $\Rightarrow O(n^2)$ algorithm

Better idea:

How did we sort in Mergesort without comparing every pair?



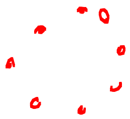
```
MERGE-AND-COUNT (A[1..n], B[1..m]) :
    CurrentA ← 1 , currentB ← 1
    Count ← 0
    index ← 1
    While (currentA ≤ n  and  currentB ≤ m)
        If (A[currentA] ≤ B[currentB])
            output[index] ← A[currentA]
            currentA ← currentA + 1                    O(max{n, m})
        Else
            output[index] ← B[currentB]
my only new line
            currentB ← currentB + 1
            count ← Count + (n − currentA)
        index ← index + 1
    endwhile
    Copy rest of nonempty list to output
    return (Count, output)
```

```
SORT-AND-COUNT ($L[1..k]$)
    If $k = 0$ or $1$
        no inversions, return $(0, L)$
    Else
        $A \leftarrow$ first $\lceil k/2 \rceil$ elements of $L$
        $B \leftarrow$ last $\lfloor k/2 \rfloor$ elements of $L$
$T(\frac{n}{2}) \rightarrow$   $(r_A, A) \leftarrow$ SORT-AND-COUNT $(A)$
$T(\frac{n}{2}) \rightarrow$   $(r_B, B) \leftarrow$ SORT-AND-COUNT $(B)$
$O(n) \rightarrow$   $(r, L) \leftarrow$ MERGE-AND-COUNT$(A, B)$
    endif
    return $(\underbrace{r_A + r_B + r}, L)$
```

Running time?

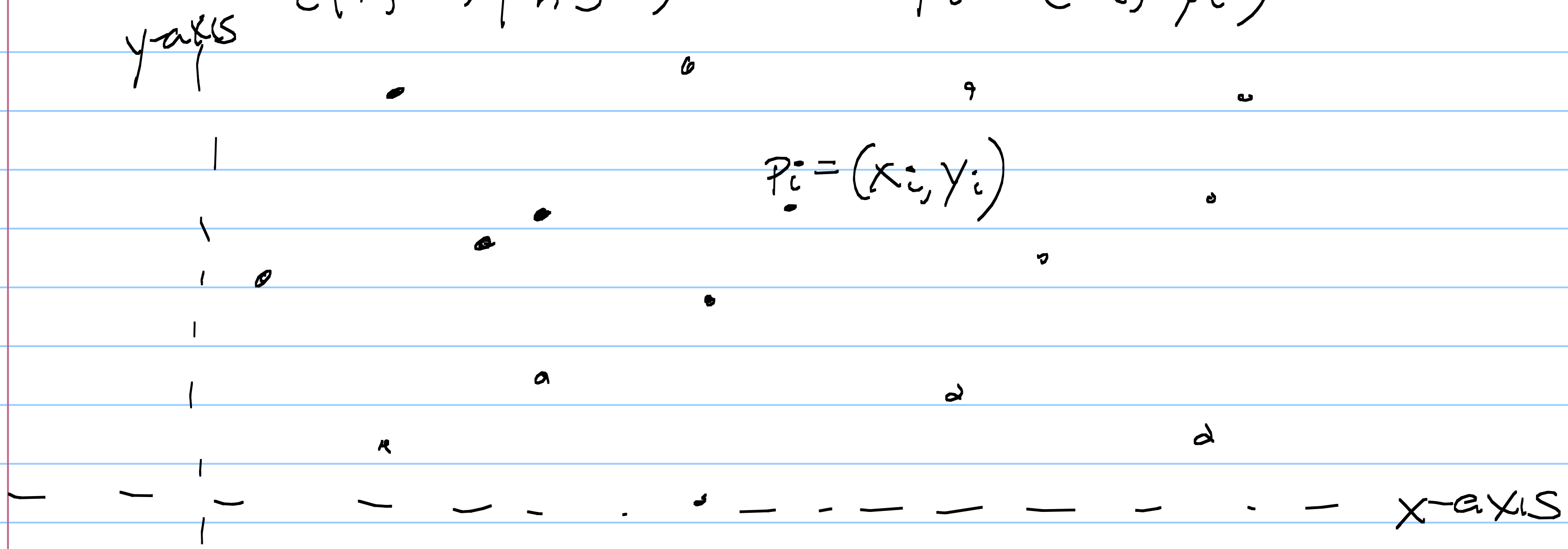
$$T(n) = 2T(\tfrac{n}{2}) + O(n) + \cancel{O(1)}$$

Closest Pair of Points



Let P be a set of points in $\mathbb{R}^2$.

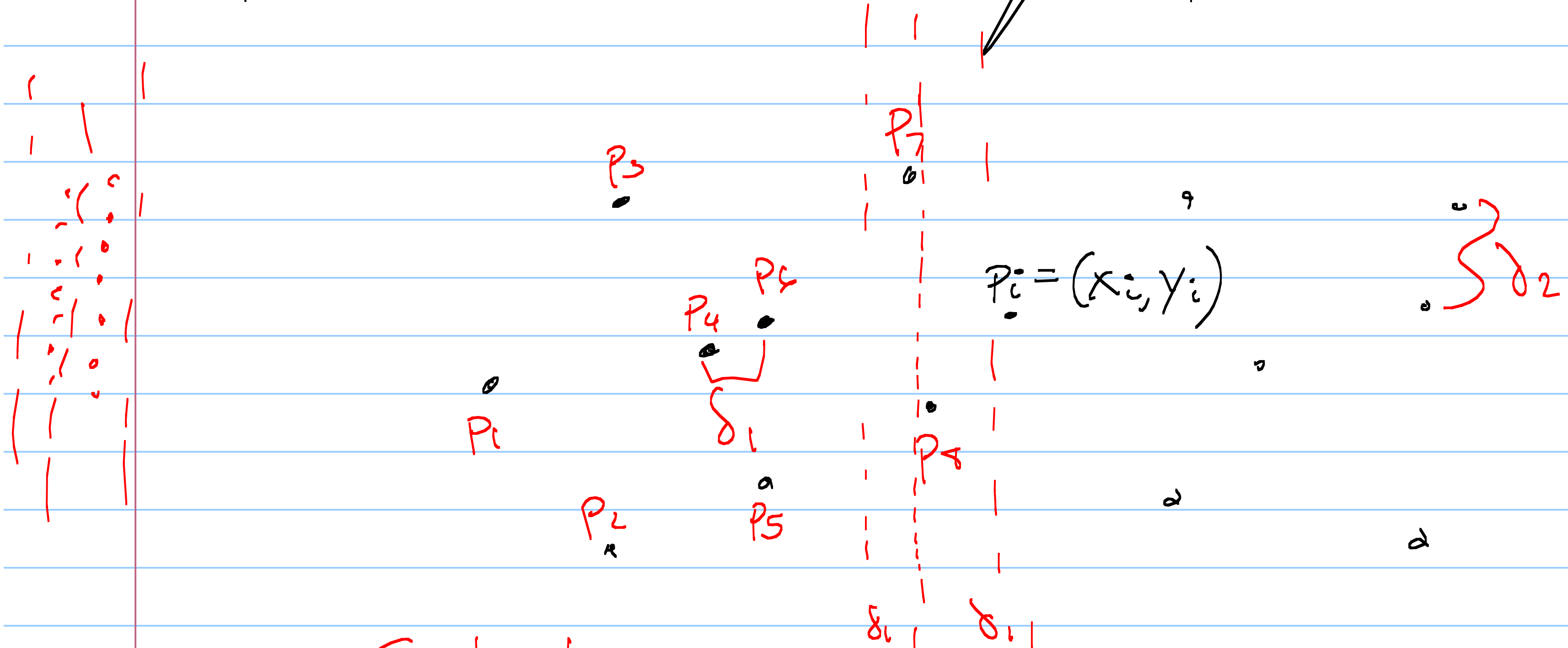
$P = \{p_1, \dots, p_n\}$, and $p_i = (x_i, y_i)$



Q: What is the closest pair of points?

Naive : $O(n^2)$

How to divide & conquer?



Sort by x-coordinate