Lecture 24

CSE 331

Oct 28, 2022

Coding P1 due TODAY!

Fri, Oct 28	Counting Inversions F21 F19 F18 F17 x^2	[KT, Sec 5.3] (**Project (Problem 1 Coding) in**)
Mon, Oct 31	Multiplying large integers F21 F19 F18 F17 x^2	[KT, Sec 5.5] (**Project (Problem 1 Reflection) in**) *Reading Assignment:* Unraveling the mystery behind the identity
Wed, Nov 2	Closest Pair of Points F21 F19 F18 F17 x^2	[KT, Sec 5.4]
Fri, Nov 4	Kickass Property Lemma F21 F19 F18 F17 x^2	[KT, Sec 5.4] (**Project (Problem 2 Coding) in**)
Mon, Nov 7	Weighted Interval Scheduling F21 F19 F17 x^2	[KT, Sec 6.1] (**Project (Problem 2 Reflection) in**)

Group formation instructions

Autolab group submission for CSE 331 Project

The lowdown on submitting your project (especially the coding and reflection) problems as a group on Autolab.

Follow instructions **EXACTLY** as they are stated

The instruction below are for Coding Problem 1

You will have to repeat the instructions below for EACH coding AND reflection problem on project on Autolab (with the appropriate changes to the actual problem).

Form your group on Autolab

Groups on Autolab will NOT be automatically created

You will have to form a group on Autolab by yourself (as a group). Read on for instructions on how to go about this.

Click to add notes

Make sure you are in your group

note @366 stop following **2 views**

Actions

Coding P1 due today

A gentle reminder that the first coding problem is due by 11:59pm tonight!

Finally, make sure that you are officially included in your group on Autolab for the coding problem 1 *before* your group submits its code. **If you are not included in the group on Autolab, you will get a ZERO on coding problem 1.**

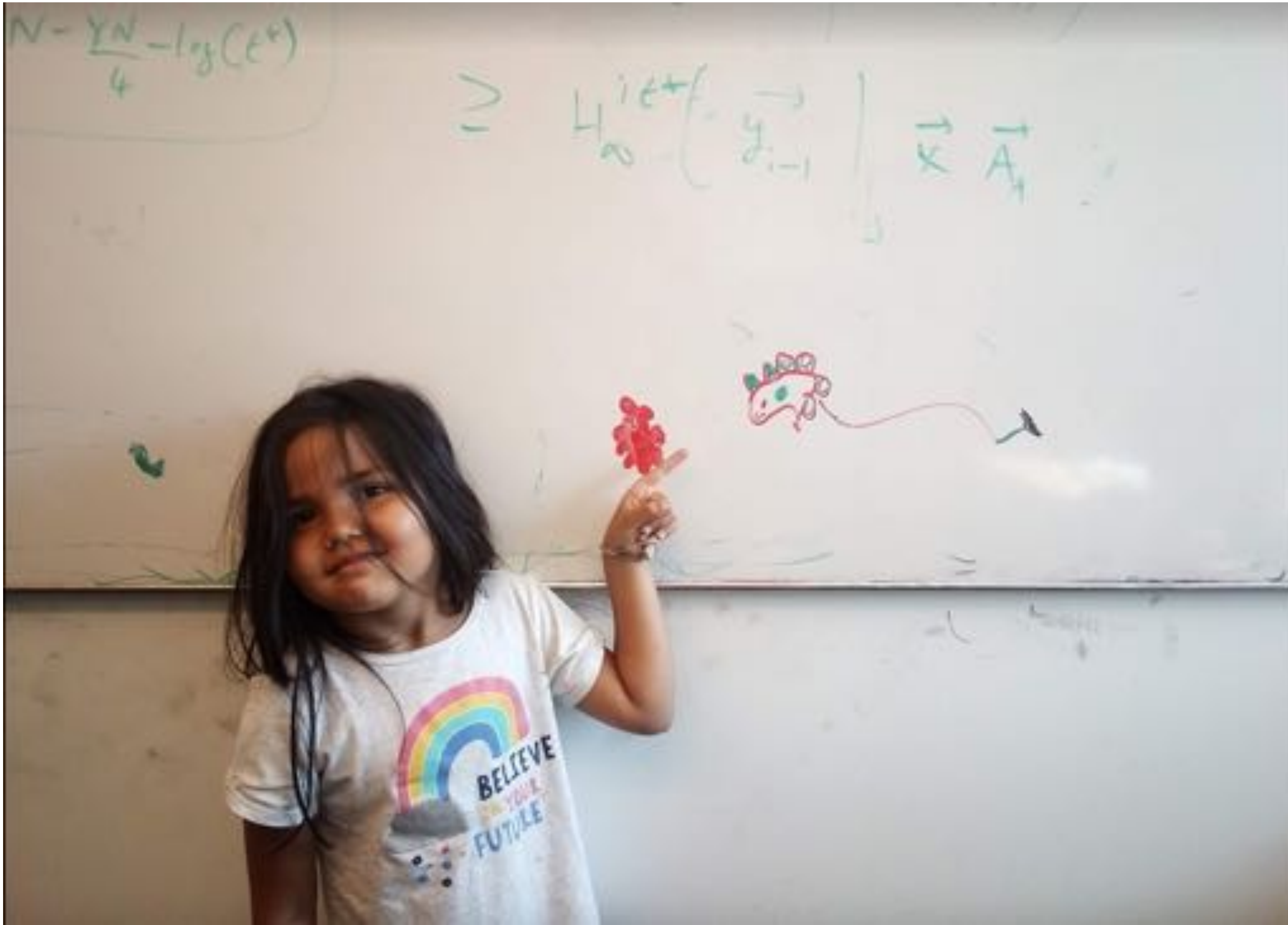
Please make sure that you verify that you see a submission for yourself on Autolab. It is your PERSONAL RESPONSIBILITY to make sure that this is the case. *If your group forgets to do this is it your responsibility to reminder them that you need to be included.*

If your group has already submitted without you, make sure you are included in the group on Autolab and then someone from *your group should re-submit.*

project

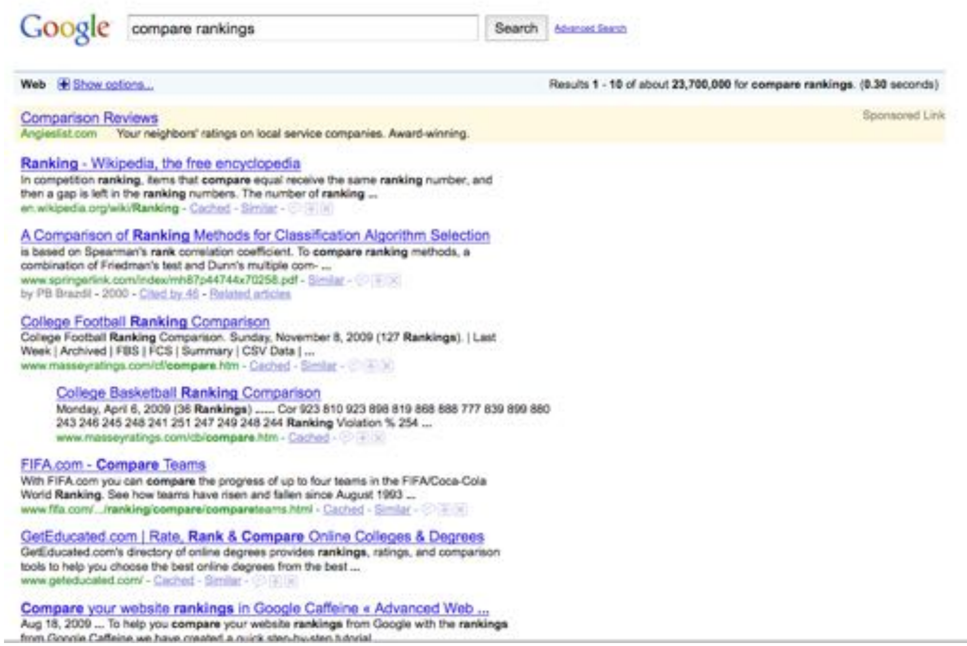
Edit good note | 0 Updated 2 minutes ago by Atri Rudra

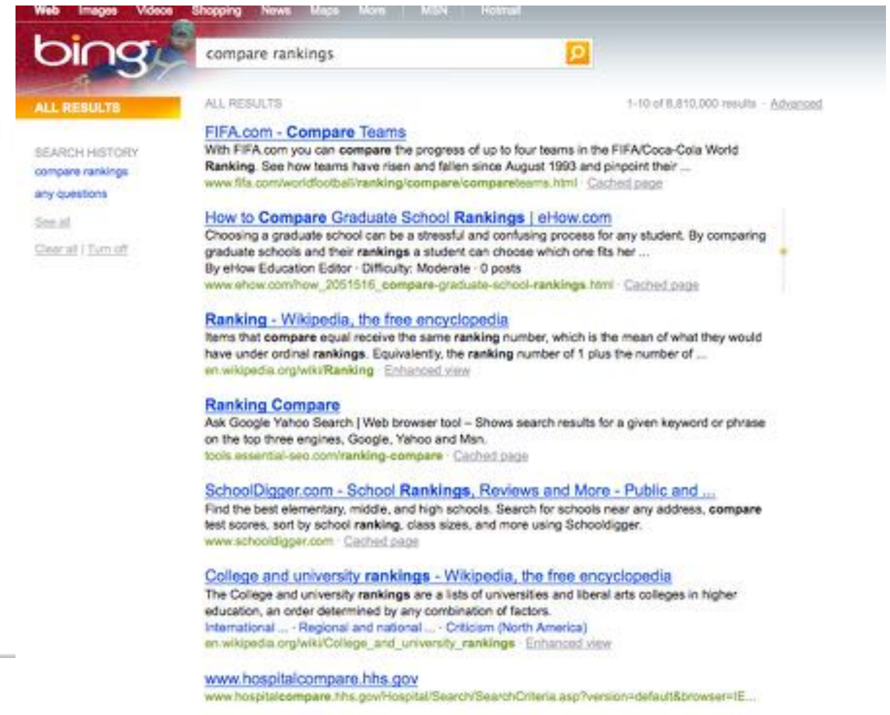
Questions/Comments?



Rankings

How close are two rankings?





Rest of today's agenda

Formal problem: Counting inversions

Divide and Conquer algorithm

Problem definition on the board…

Solve a harder problem

Input: $a_1, .., a_n$

Output: LIST of all inversions

```
L = ϕ
for i in 1 to n-1
    for j in i+1 to n
        If a_i > a_j
            add (i,j) to L

return  L
```

Optimal for the listing problem

Example 1: All inversions-- (2i-1,2i)

2	1	3	4	6	5	7	8

Only check (i,i+1) pairs

Q1: Solve listing problem in O(n) time?

Q2: Recursive divide and conquer algorithm to count the number of inversions?

CountInv (a,n)

if $n = 1$ return 0

if $n = 2$ return $a_1 > a_2$

$a_L = a_1 , .., a_{[n/2]}$

$a_R = a_{[n/2]+1} , .., a_n$

return CountInv(a_L, [n/2]) + CountInv(a_R, n- [n/2])

Can be horribly wrong in general

CountInv (a,n)

if $n = 1$ return 0

if $n = 2$ return $a_1 > a_2$

$a_L = a_1 , .., a_{[n/2]}$

$a_R = a_{[n/2]+1} , .., a_n$

return CountInv(a_L, $[n/2]$) + CountInv(a_R, $n - [n/2]$)

Example where instance has non-zero (can be $\Omega(n^2)$) inversions and algo returns 0?

5	6	1	2

All 4 “crossing” pairs are inversions

Bad case: "crossing inversions"

CountInv (a,n)

if $n = 1$ return 0

if $n = 2$ return $a_1 > a_2$

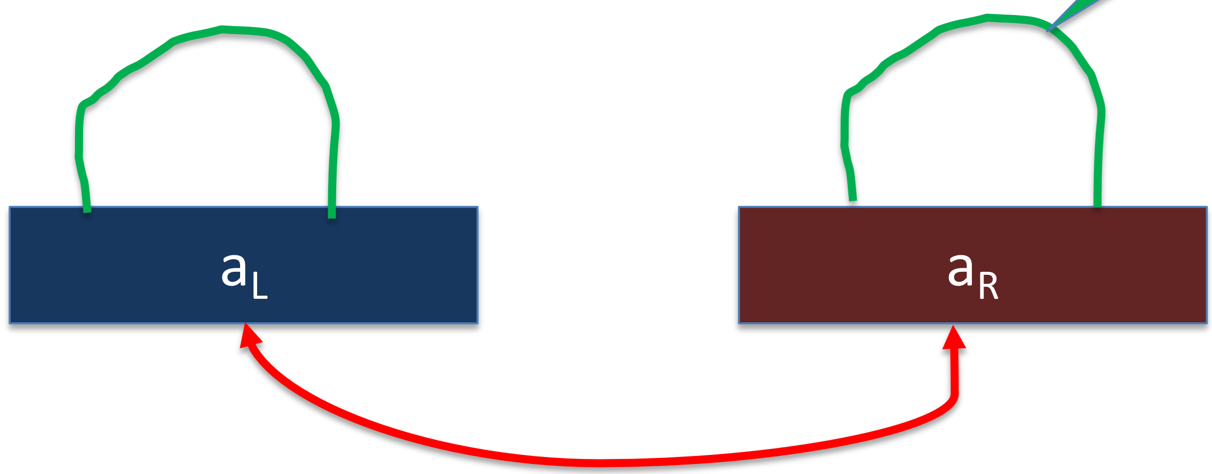
$a_L = a_1 , .., a_{[n/2]}$

$a_R = a_{[n/2]+1} , .., a_n$

return CountInv(a_L, $[n/2]$) + CountInv(a_R, $n- [n/2]$)

Yes!

Are a_L and a_R sorted?



Example 2: Solving the bad case

5	6	.....

a_L

1

a_R

a_L is sorted

First element is a_L is larger than first/only element in a_R

O(1) algorithm to count number of inversions?

return size of a_L

Example 3: Solving the bad case

1

a_L

5	6	.....

a_R

a_R is sorted

First/only element is a_L is smaller than first element in a_R

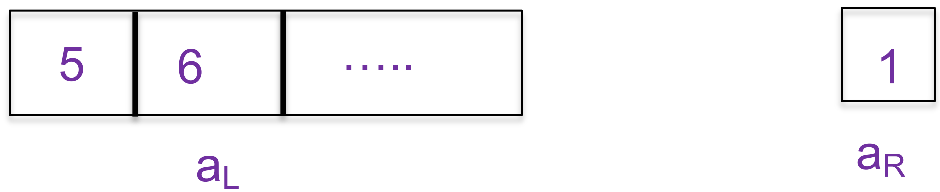
O(1) algorithm to count number of inversions?

return 0

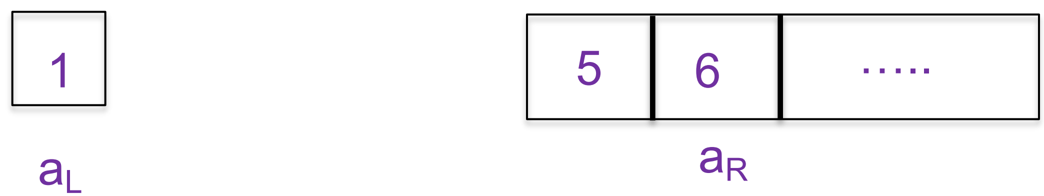
Solving the bad case

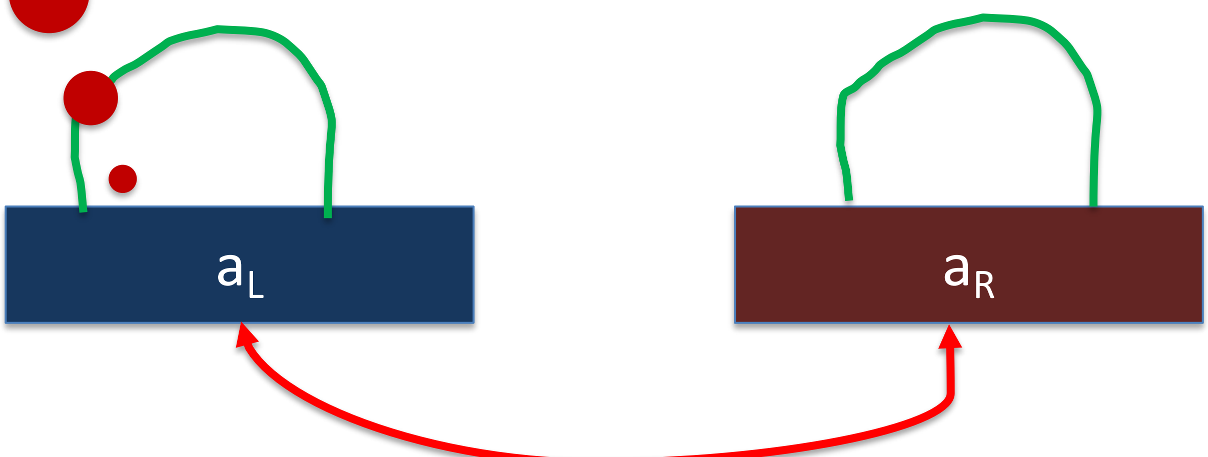
Try to modify the MERGE algorithm

First element of a_L is larger than first element of a_R



First element of a_L is smaller than first element of a_R





Divide and Conquer

Divide up the problem into at least two sub-problems

Solve all sub-problems: Mergesort

Recursively solve the sub-problems

Solve stronger sub-problems: Inversions

"Patch up" the solutions to the sub-problems for the final solution

MergeSortCount algorithm

Input: $a_1, a_2, \ldots, a_n$

Output: Numbers in sorted order+ #inversion

```
MergeSortCount( a, n )
    If n = 1 return  ( 0 , a_1)
    If n = 2 return  ( a1 > a2, min(a_1,a_2); max(a_1,a_2))

    a_L = a_1,..., a_{n/2}        a_R = a_{n/2+1},..., a_n

    (c_L, a_L) = MergeSortCount(a_L, n/2)

    (c_R, a_R) = MergeSortCount(a_R, n/2)

    (c, a) = MERGE-COUNT(a_L,a_R)

    return (c+c_L+c_R,a)
```

Counts #crossing-inversions+ MERGE

$T(2) = c$

$T(n) = 2T(n/2) + cn$

O(n log n) time

O(n)

MERGE-COUNT(a_L,a_R)

$a_L = l_1, \ldots, l_{n'}$ $\qquad$ $a_R = r_1, \ldots, r_m$

$c = 0$

$i,j = 1$

while $i \leq n'$ and $j \leq m$

  if $l_i \leq r_j$

    add l_i to output

    i ++

  else

    add r_j to output

    j ++

    c += $n' - i + 1$

Output any remaining items

return c