Lecture 25

CSE 331

Oct 31, 2022

Response to feedback up!

note @371

stop following 1 view

Actions

Reponse to feedback

Thanks to everyone who give feedback (@221)!

Below, I will post some pie-charts that I think give some interesting overall picture of how y'all feel about the course and then some responses to the written comments. I apologize for the delay in doing this and I understand that some of this feedback could have been useful if given earlier-- sorry about that :-(

First some pie-charts:

Overall your feeling about CSE 331

30 responses

While of course having unhappy/very unhappy students is not ideal, at least the fraction of students who are very unhappy are (comfortably larger) than those that are not very unhappy. This was not the case in the few couple of offerings so I'm glad to see the "tide turn" time around. Also 50% of the respondents are not unhappy. Again not ideal but better than where this was few course offerings ago.

Reflection P1 due TODAY!

Fri, Oct 28	Counting Inversions F21 F19 F18 F17 x²	[KT, Sec 5.3] (**Project (Problem 1** `Coding` **) in**)
Mon, Oct 31	Multiplying large integers F21 F19 F18 F17 x²	[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 x²	[KT, Sec 5.4]
Fri, Nov 4	Kickass Property Lemma F21 F19 F18 F17 x²	[KT, Sec 5.4] (**Project (Problem 2** `Coding` **) in**)
Mon, Nov 7	Weighted Interval Scheduling F21 F19 F17 x²	[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

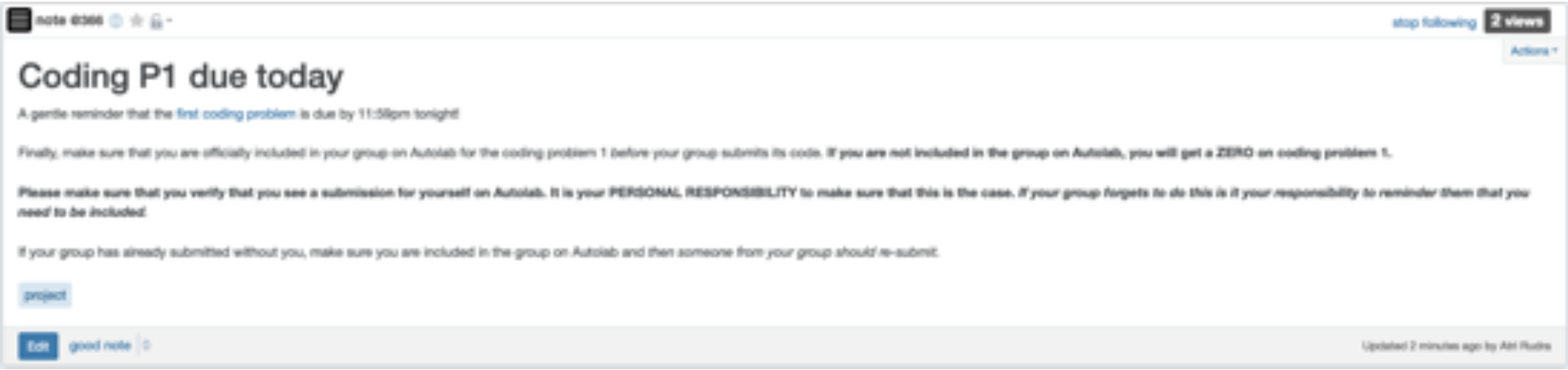
You will have to repeat the instructions below for EACH coding AND reflection problem on project on Autolab (with the appropriate changes to the actual problem).

Form your group on Autolab

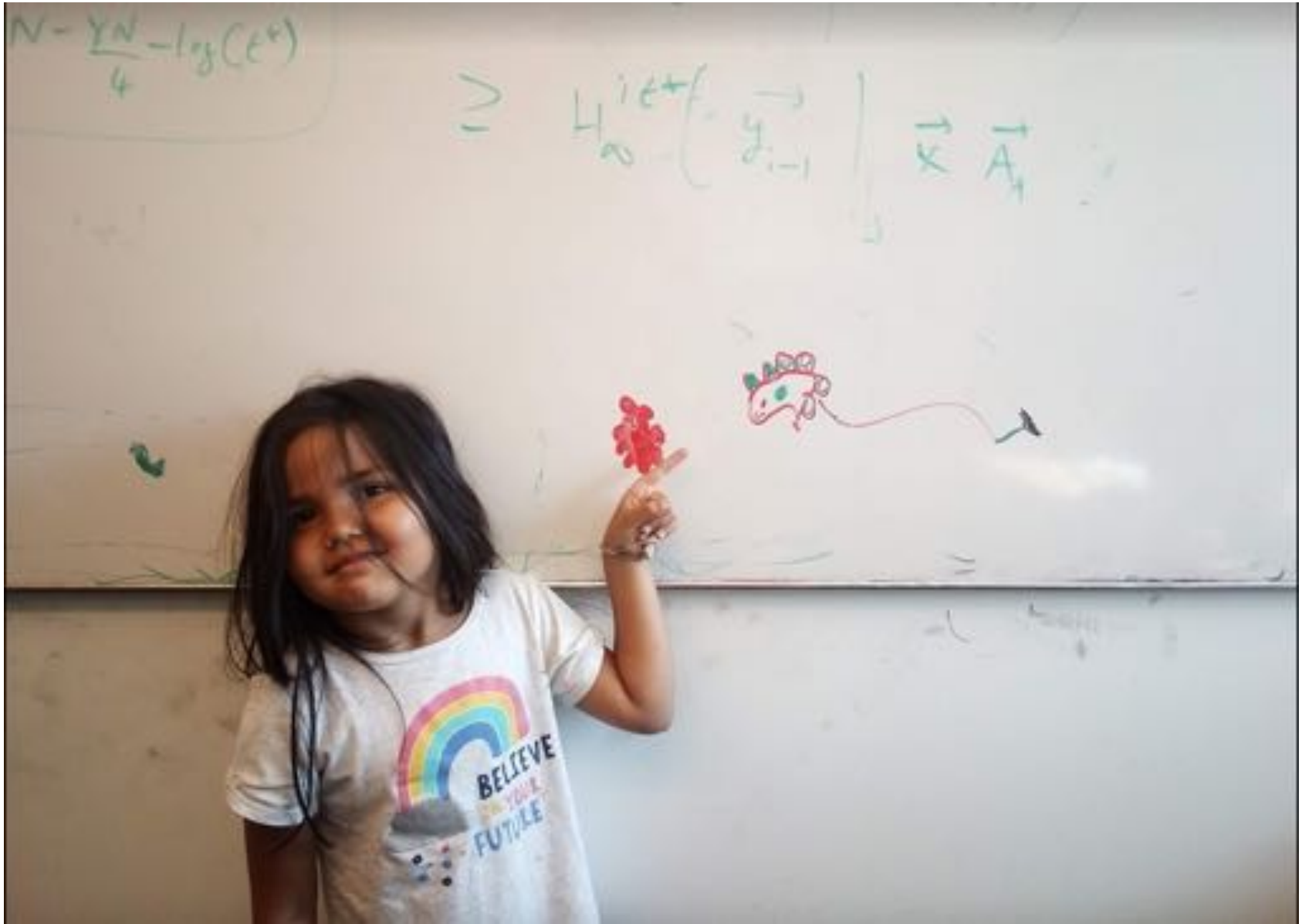
Groups on Autolab will NOT be automatically created

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

Make sure you are in your group



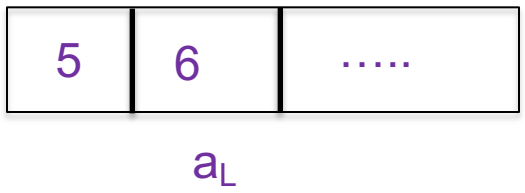
Questions/Comments?

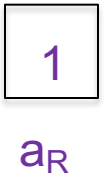


Solving the bad case

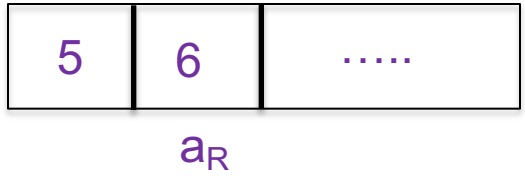
Try to modify the MERGE algorithm

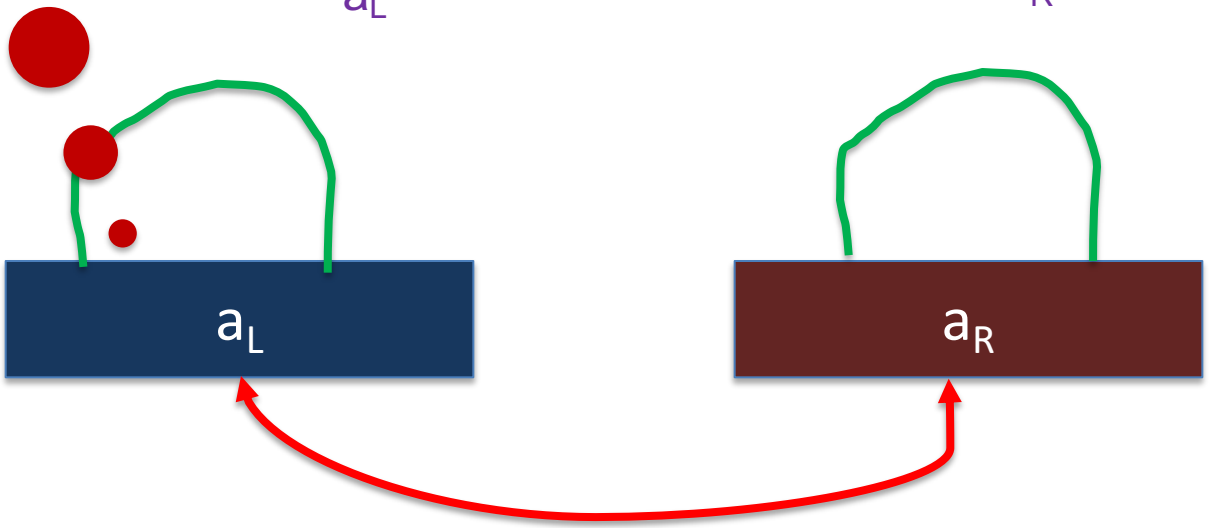
First element of a_L is larger than first element of a_R





First element of a_L is smaller than first element of a_R

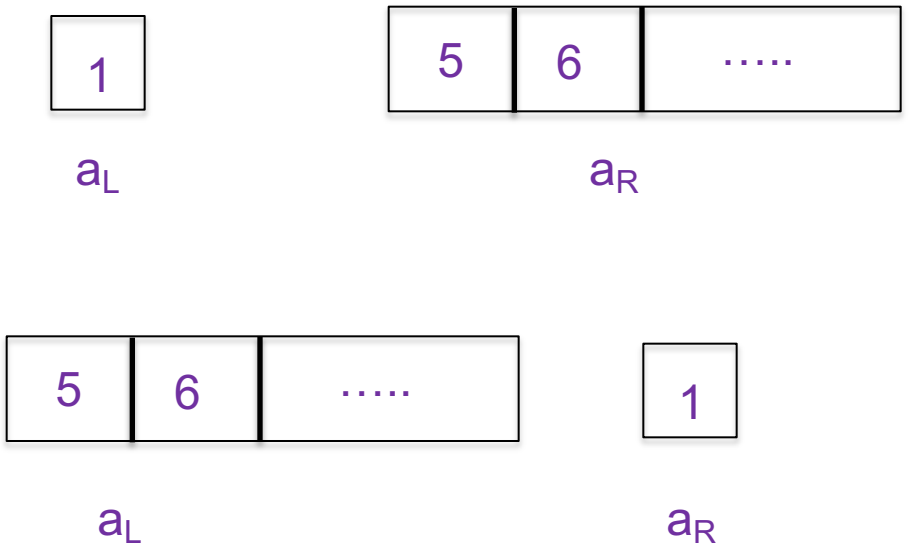




MERGE-COUNT(a_L,a_R)

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

```
c = 0
i,j = 1
while i ≤ n' and j ≤ m
        if l_i ≤ r_j
            i ++
            add l_i to output
        else
            add r_j to output
            j ++
            c += n'- i +1
Output any remaining items
return c
```



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 , a1)
    If n = 2 return  ( a1 > a2, min(a1,a2); max(a1,a2))
    aL = a1,..., an/2        aR = an/2+1,..., an
    (cL, aL) = MergeSortCount(aL, n/2)
    (cR, aR) = MergeSortCount(aR, n/2)
    (c, a) = MERGE-COUNT(aL,aR)
    return (c+cL+cR,a)
```

$T(2) = c$

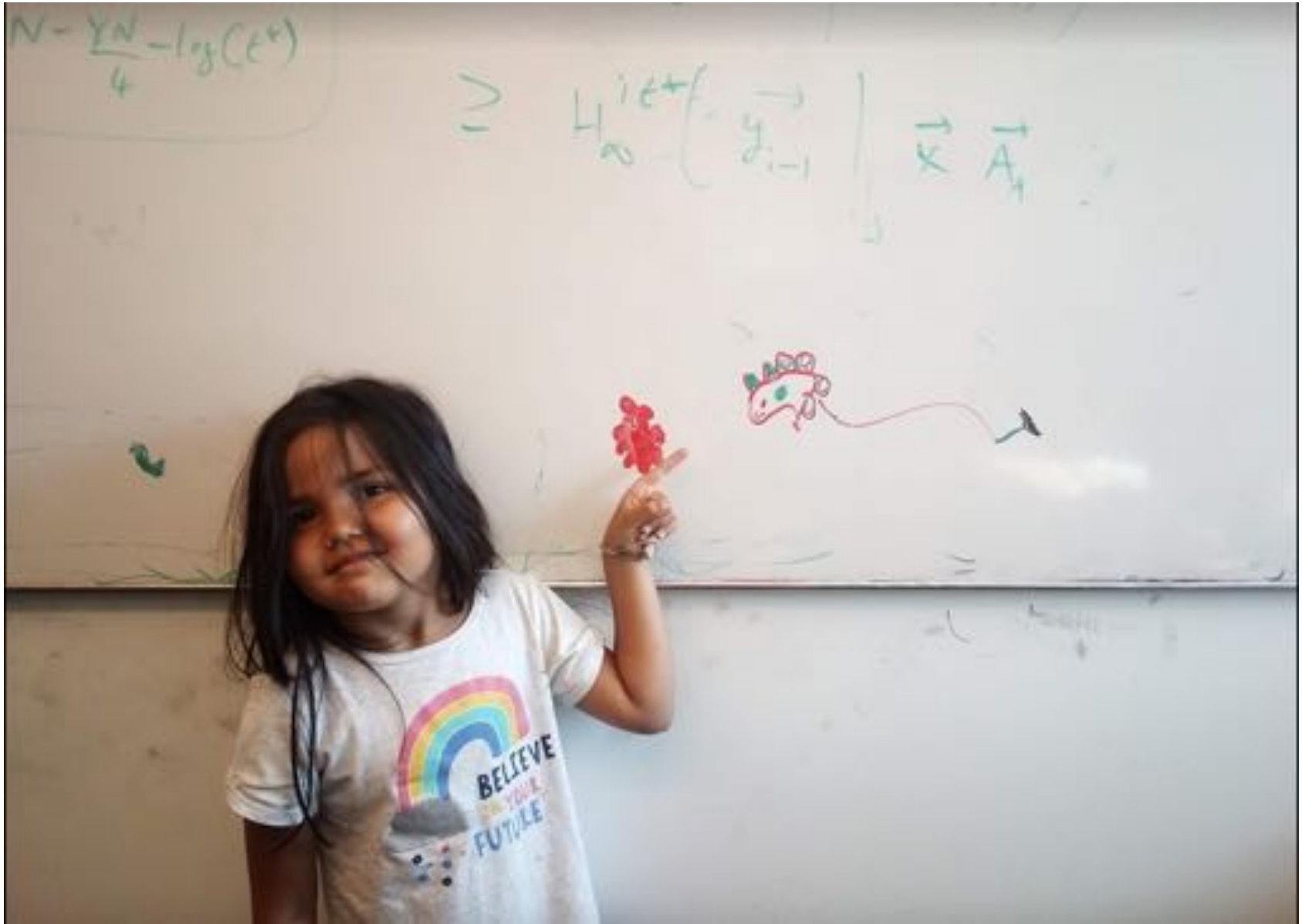
$T(n) = 2T(n/2) + cn$

O(n log n) time

O(n)

Counts #crossing-inversions+ MERGE

Questions/Comments?



Divide and Conquer

Divide up the problem into at least two sub-problems

Recursively solve the sub-problems

“Patch up” the solutions to the sub-problems for the final solution

Improvements on a smaller scale

Greedy algorithms: exponential → poly time

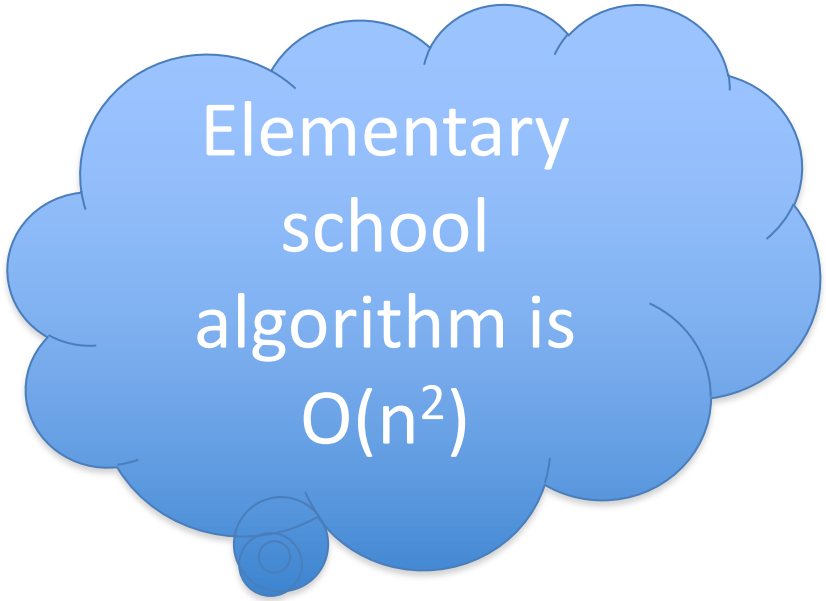
(Typical) Divide and Conquer: $O(n^2)$ → asymptotically smaller running time

Multiplying two numbers

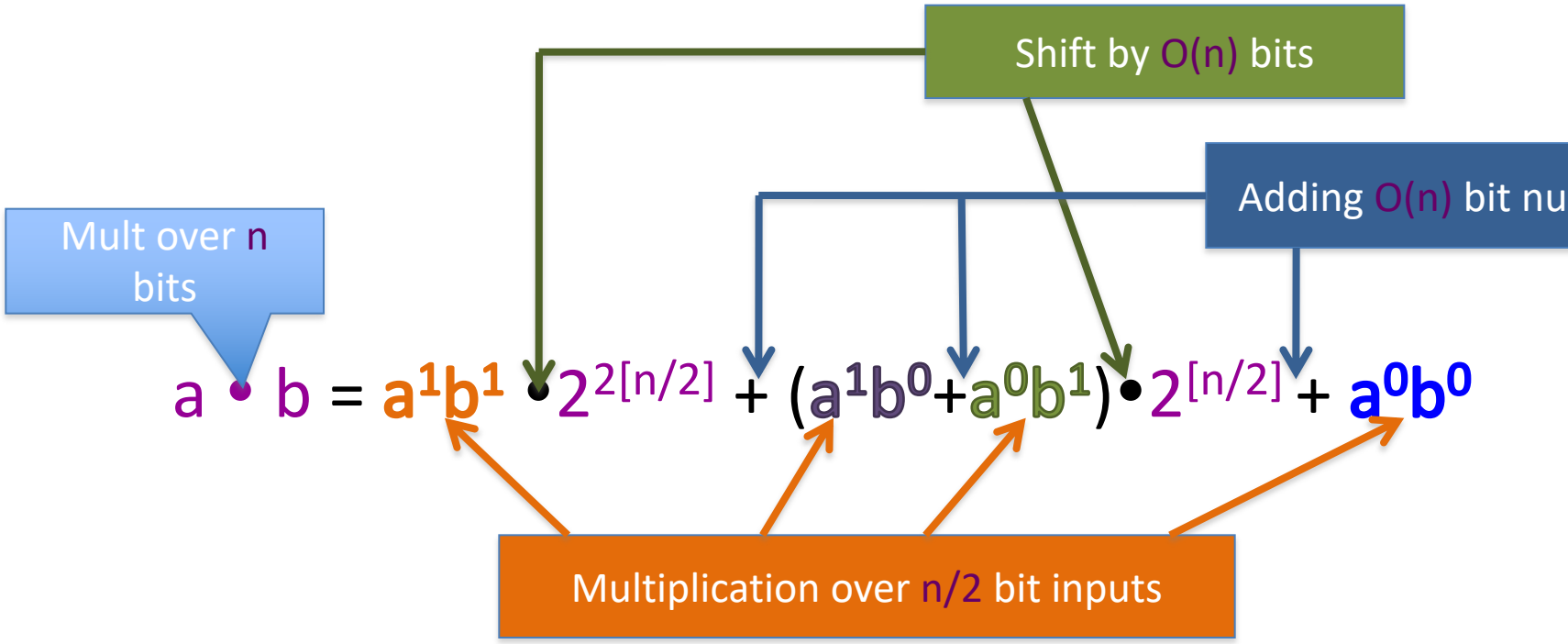
Given two numbers a and b in binary

$a=(a_{n-1},..,a_0)$ and $b = (b_{n-1},\ldots,b_0)$

Compute c = a x b



The current algorithm scheme



$T(n) \leq 4T(n/2) + cn$

$T(1) \leq c$

$T(n)$ is $O(n^2)$