Lecture 25

CSE 331

Nov 1, 2021

Please have a face mask on

Masking requirement

UB requires all students, employees and visitors – regardless of their vaccination status – to wear face coverings while inside campus buildings.

https://www.buffalo.edu/coronavirus/health-and-safety/health-safety-guidelines.html

Reflection P1 due TODAY!

Fri, Oct 29	Counting Inversions F19 F18 F17 x²	[KT, Sec 5.3] (**Project (Problem 1** `Coding` **) in**)
Mon, Nov 1	Multiplying large integers F19 F18 F17 x²	[KT, Sec 5.5] (**Project (Problem 1** `Reflection` **) in**) *Reading Assignment:* Unraveling the mystery behind the identity
Wed, Nov 3	Closest Pair of Points F19 F18 F17 x²	[KT, Sec 5.4]
Fri, Nov 5	Kickass Property Lemma F19 F18 F17 x²	[KT, Sec 5.4] (**Project (Problem 2** `Coding` **) in**)
Mon, Nov 8	Weighted Interval Scheduling 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.

Including accepting the invitation

note @416 — stop following — 35 views

Issues with Group formation for Coding Problem 1

I took a quick glance at the groups formed for the coding problem 1 for the project. Most of you did follow the instructions for forming group, which is great.

Unfortunately, some of you did not. In particular, there are three potential issues for some groups:

1. **Not all group members confirmed their memberships**. I.e. after one group member has created the group on Autolab, the *other* groups members have to ***explicitly accept*** the invitation (see the section titled "Accepting an invitation to the group" in the instructions) but they did not. If this happens then students who *did not* accept the invitation ***WILL GET A ZERO*** (this is because as far as Autolab is concerned that student did not submit anything).
   - Easiest way to check if you fall in this category is to either check your group setting on the Coding Problem 1 page on Autolab or see your score on Autolab for coding problem #1.
   - If you fall in this category, you have till **10pm tonight (Sat, Oct 30) to accept the invitation** and then email me once you are done.
     - After I hear back from you, I'll make sure you get the correct number of points for Coding Problem #1 (currently you should have a zero).
     - If for some reason Autolab does not allow you to accept the invitation because the deadline is past, email me with that information as well.
2. **Group size is not exactly three**. For some of the groups, I gave explicit permission to submit in groups of size <3. However, *if I did not* ***explicitly give you permission by email*** to submit in groups of size <3 (but you still submitted in a group of size <3) you are in potential **VIOLATION OF ACADEMIC INTEGRITY**. Note that a group member not responding in time is **NOT** a valid reason to submit **without getting my explicit permission.**
   - If you fall in the above category you have till **10pm tonight (Sat, Oct 30) to email me and get my explicit permission** to submit in groups of size <3.
   - If you do not do the above, then **YOU WILL GET A ZERO ON THE *ENTIRE* PROJECT** and potentially a letter grade reduction in your final CSE 331 grade. So make sure you send me that email by 10pm tonight.
3. This actually I have *not* verified but **please make sure that your group on Autolab is EXACTLY the group you either signed up for or were assigned**. The only exception is

Preliminary grading rubric

note @401

5 views

Preliminary rubrics for reflections problems up

We have added **preliminary** grading rubrics for each reflection question:

http://www-student.cse.buffalo.edu/~atri/cse331/fall21/project/reflection.html

As noted in the page above, please keep in mind that in **actual grading, we will use a grading rubric that expands on the preliminary grading rubric,** i.e. you are NOT seeing the final rubric that will be used to grade your submissions.

We hope this preliminary grading rubric helps as y'all start working on the reflection questions.

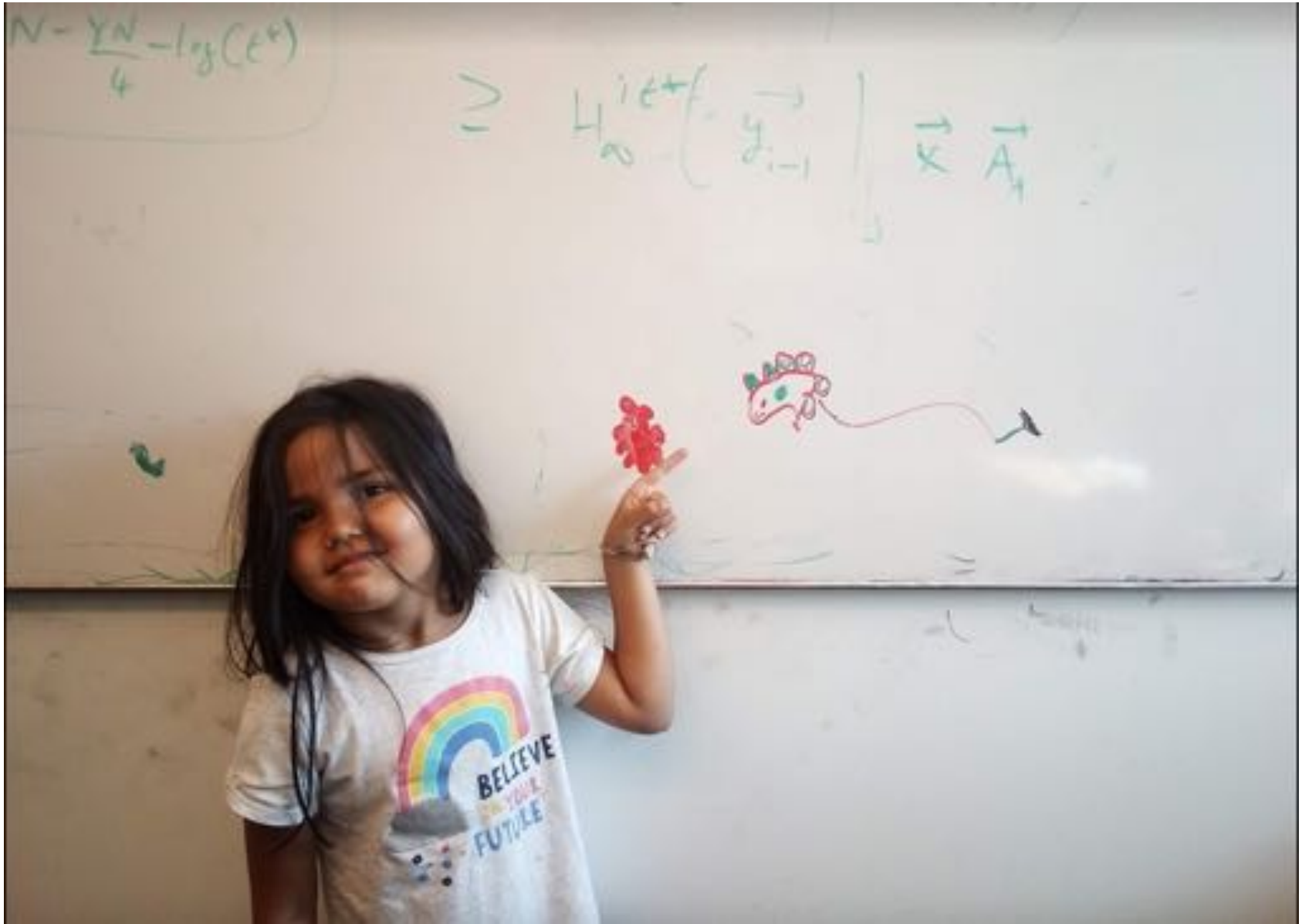
project

Preliminary Grading Guidelines

Below is a *preliminary* instantiation of the generic grading rubric above for (all ten parts of) Problem 1. In **actual grading, we will use a grading rubric that expands on the preliminary grading rubric below**.

- `Level 0`
  1. The authors did not respond with all 10 stakes; OR
  2. Answers may not be entirely relevant to the assignment.
- `Level 1`
  1. The authors did respond with all 10 stakes. Although, the responses may be underdeveloped; AND
  2. The authors clearly understand the questions, but have not demonstrated much effort in thinking through the different interests each stakeholder would have. Answers may seem perfunctory.
- `Level 2`
  1. The authors respond with all 10 stakes thoroughly and thoughtfully; AND
  2. The authors clearly demonstrate their grasp of the questions and the various perspectives each stakeholder might have on the same design; AND
  3. They demonstrate that what stakeholders' value differs depending on their own context.

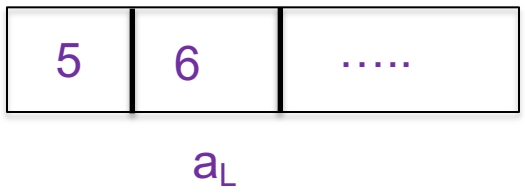
Questions/Comments?

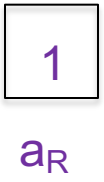


Solving the bad case

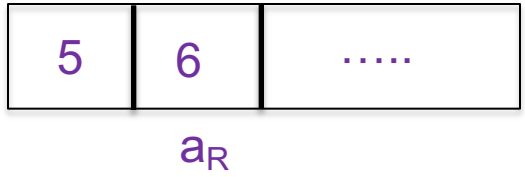
Try to modify the MERGE algorithm

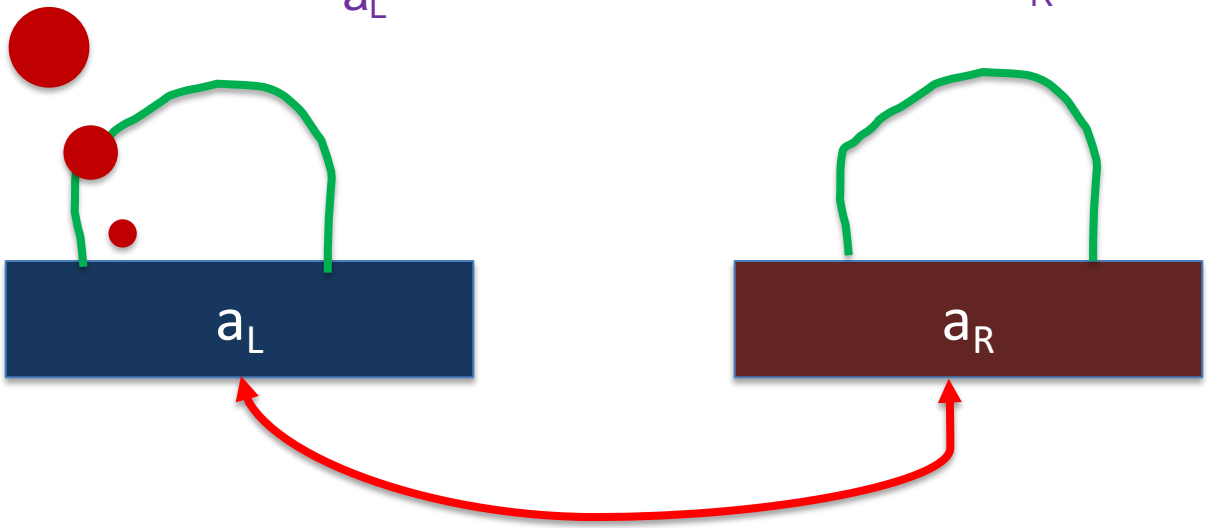
First element of a_L is larger than first element of a_R





First element of a_L is smaller than first element of a_R

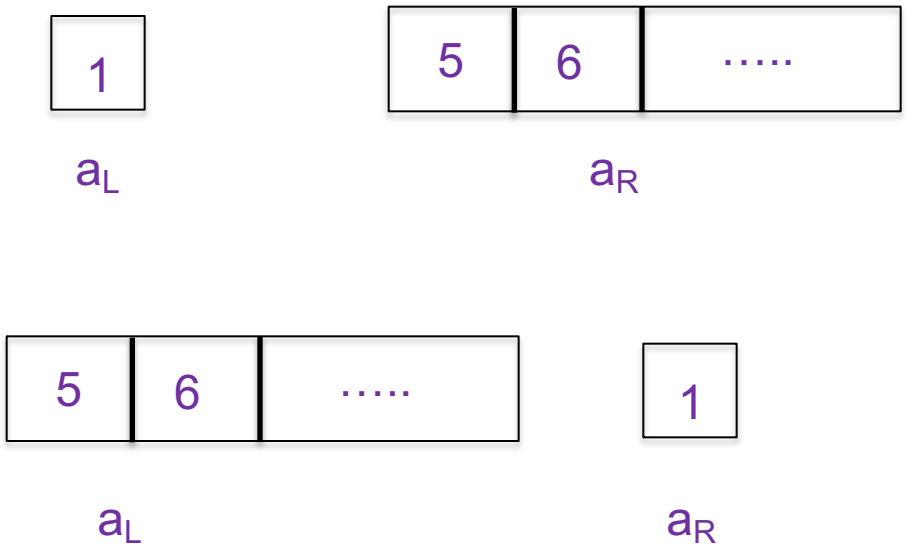




MERGE-COUNT(a_L,a_R)

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

```
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)
```

$T(2) = c$

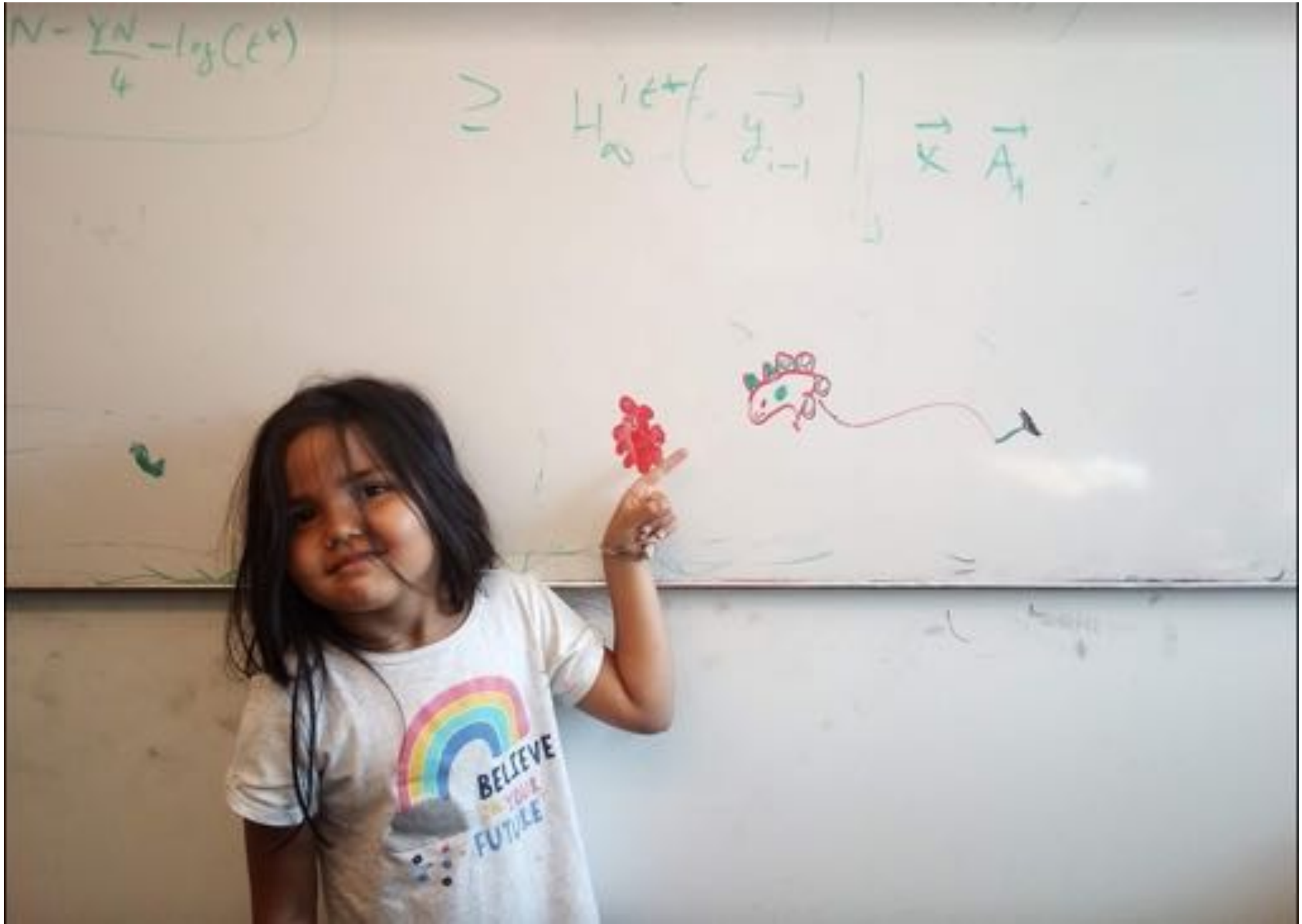
$T(n) = 2T(n/2) + cn$

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

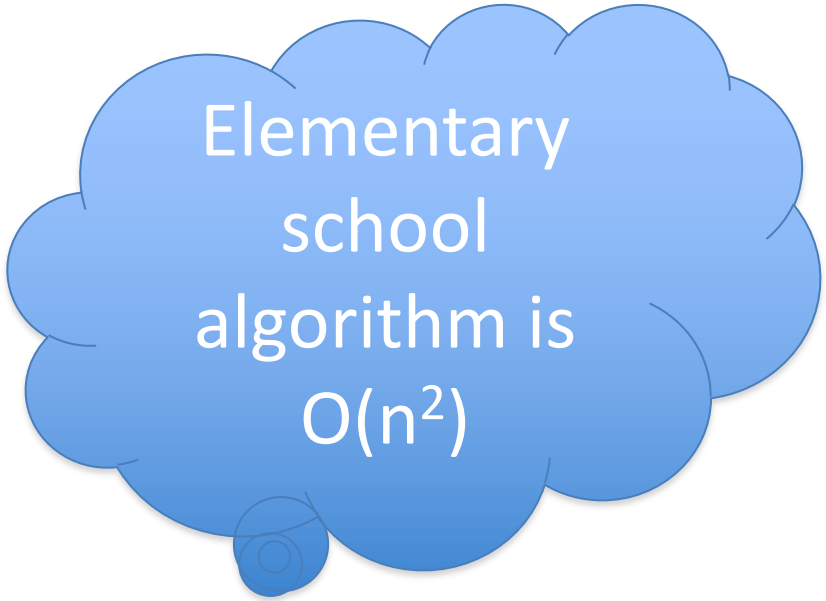
(Typical) Divide and Conquer: $O(n^2)$ → asymptotically smaller running time

Multiplying two numbers

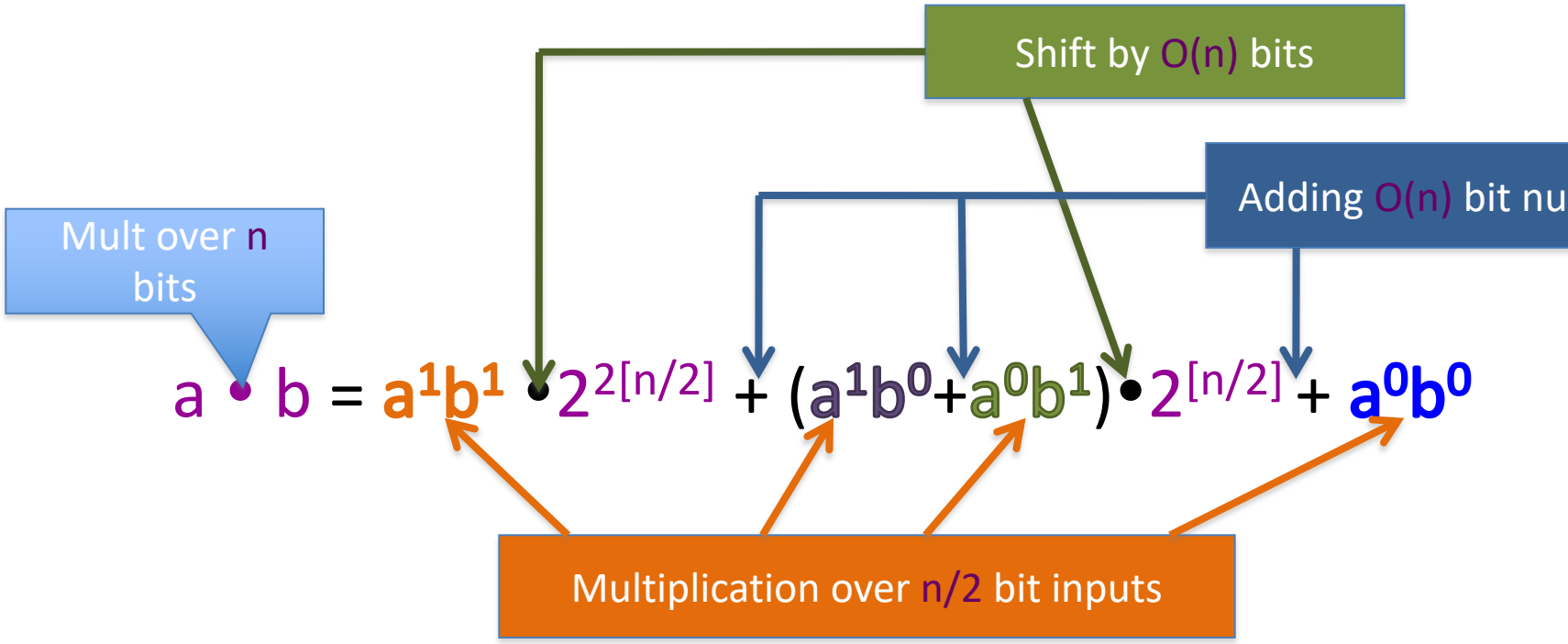
Given two numbers a and b in binary

$a=(a_{n-1},..,a_0)$ 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 $O(n^2)$