### Lecture 23

**CSE 331** 

Oct 27, 2021

### Please have a face mask on

#### Masking requirement



LIR 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

## Apologies on website being down



## Project deadlines coming up

Fri, Oct 29	Counting Inversions P19 P18 P17 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 x3	[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.

Carly to post notice

### Preliminary grading rubric



note @401 @ in is +

amp bileary 5 www

#### and present

#### Preliminary rubrics for reflections problems up

We have added preliminary grading rubrics for each reflection question:

http://www-atudent.cse.buffalo.edu/~atri/sse331/fall21/project/reflection.html

As noted in the page above, please leep 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.

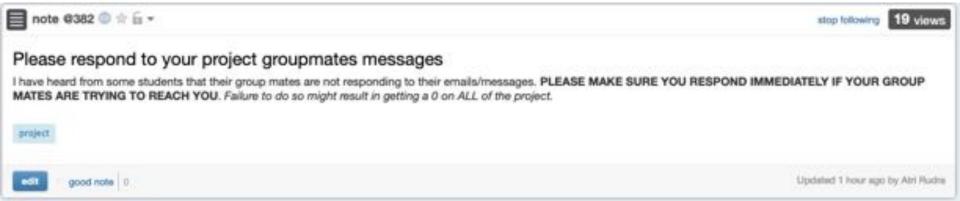


#### **Preliminary Grading Guidelines**

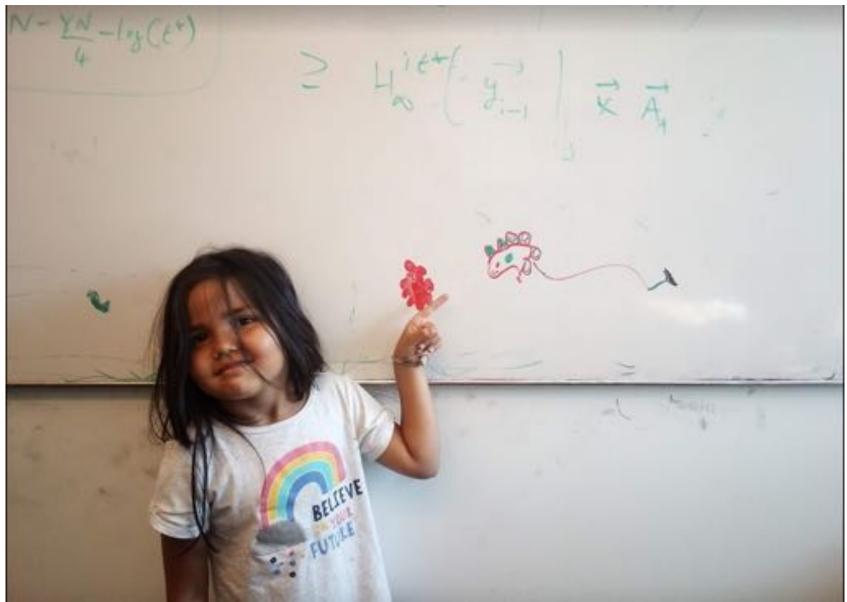
Below is a preliminary instantiation of the generic grading rubric above for (all ten parts of) Problem 1. In actual grading, we will use a grading rubric that expands on the preliminary grading rubric below.

- · Level 0
  - 1. The authors did not respond with all 10 stakes; OR
  - Answers may not be entirely relevant to the assignment.
- · Level 1
  - The authors did respond with all 10 stakes. Although, the responses may be underdeveloped; AND
  - 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
  - The authors respond with all 10 stakes thoroughly and thoughtfully; AND
  - The authors clearly demonstrate their grasp of the questions and the various perspectives each stakeholder might have on the same design; AND
  - They demonstrate that what stakeholders' value differs depending on their own contaxt.

### Please be in touch w/ your group



# Questions/Comments?



### Mergesort algorithm

Input: a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub> Output: Numbers in sorted order

```
MergeSort( a, n )

If n = 1 return the order a_1

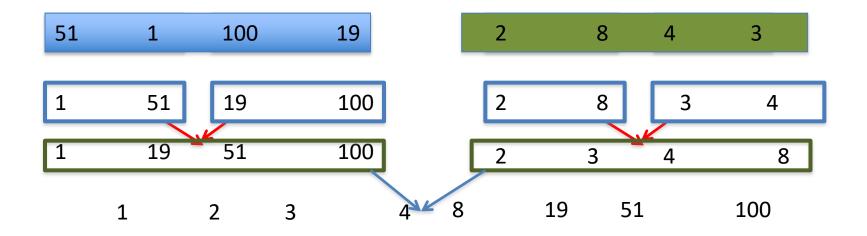
If n = 2 return the order min(a_1,a_2); max(a_1,a_2)

a_L = a_1,..., a_{n/2}

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

return MERGE ( MergeSort(a_L, n/2), MergeSort(a_R, n/2) )
```

### An example run



```
MergeSort( a, n )

If n = 1 return the order a_1

If n = 2 return the order 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

return MERGE ( MergeSort(a_L, n/2), MergeSort(a_R, n/2) )
```

### Correctness

Input: a<sub>1</sub>, a<sub>2</sub>, ..., a<sub>n</sub> Output: Numbers in sorted order

```
MergeSort( a, n )

If n = 1 return the order a_1

If n = 2 return the order 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

return MERGE (MergeSort(a_L, n/2) MergeSort(a_R, n/2)
```

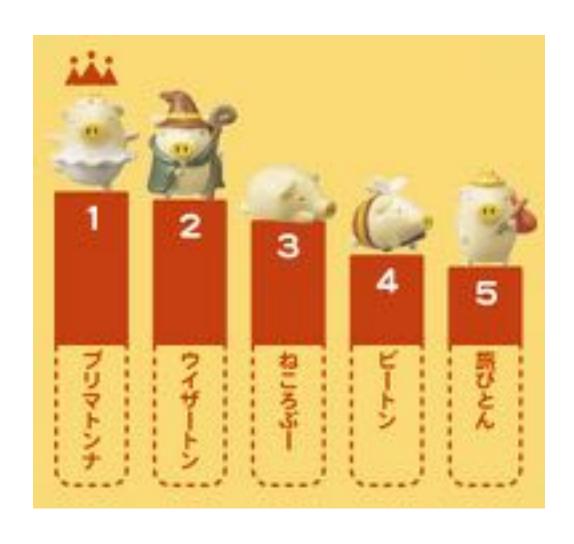


Inductive step follows from correctness of MERGE

# Runtime analysis on the board...



# Rankings



## How close are two rankings?

