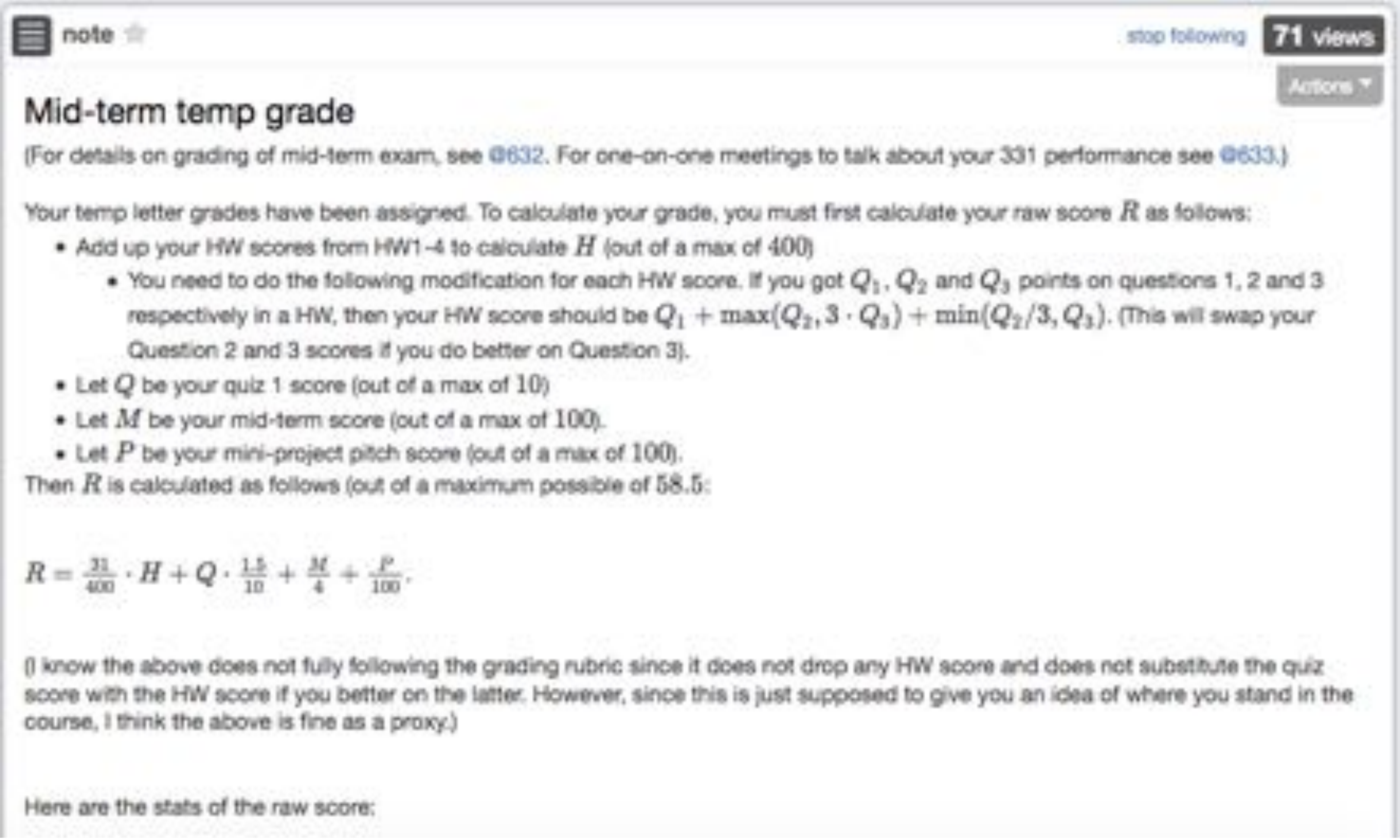
Lecture 26

CSE 331

Nov 1, 2017

Temp Grades assigned



Mid-term temp grade

(For details on grading of mid-term exam, see @632. For one-on-one meetings to talk about your 331 performance see @633.)

Your temp letter grades have been assigned. To calculate your grade, you must first calculate your raw score R as follows:

- Add up your HW scores from HW1-4 to calculate H (out of a max of 400)
    - You need to do the following modification for each HW score. If you got Q_1, Q_2 and Q_3 points on questions 1, 2 and 3 respectively in a HW, then your HW score should be $Q_1 + \max(Q_2, 3 \cdot Q_3) + \min(Q_2/3, Q_3)$. (This will swap your Question 2 and 3 scores if you do better on Question 3).
- Let Q be your quiz 1 score (out of a max of 10)
- Let M be your mid-term score (out of a max of 100).
- Let P be your mini-project pitch score (out of a max of 100).

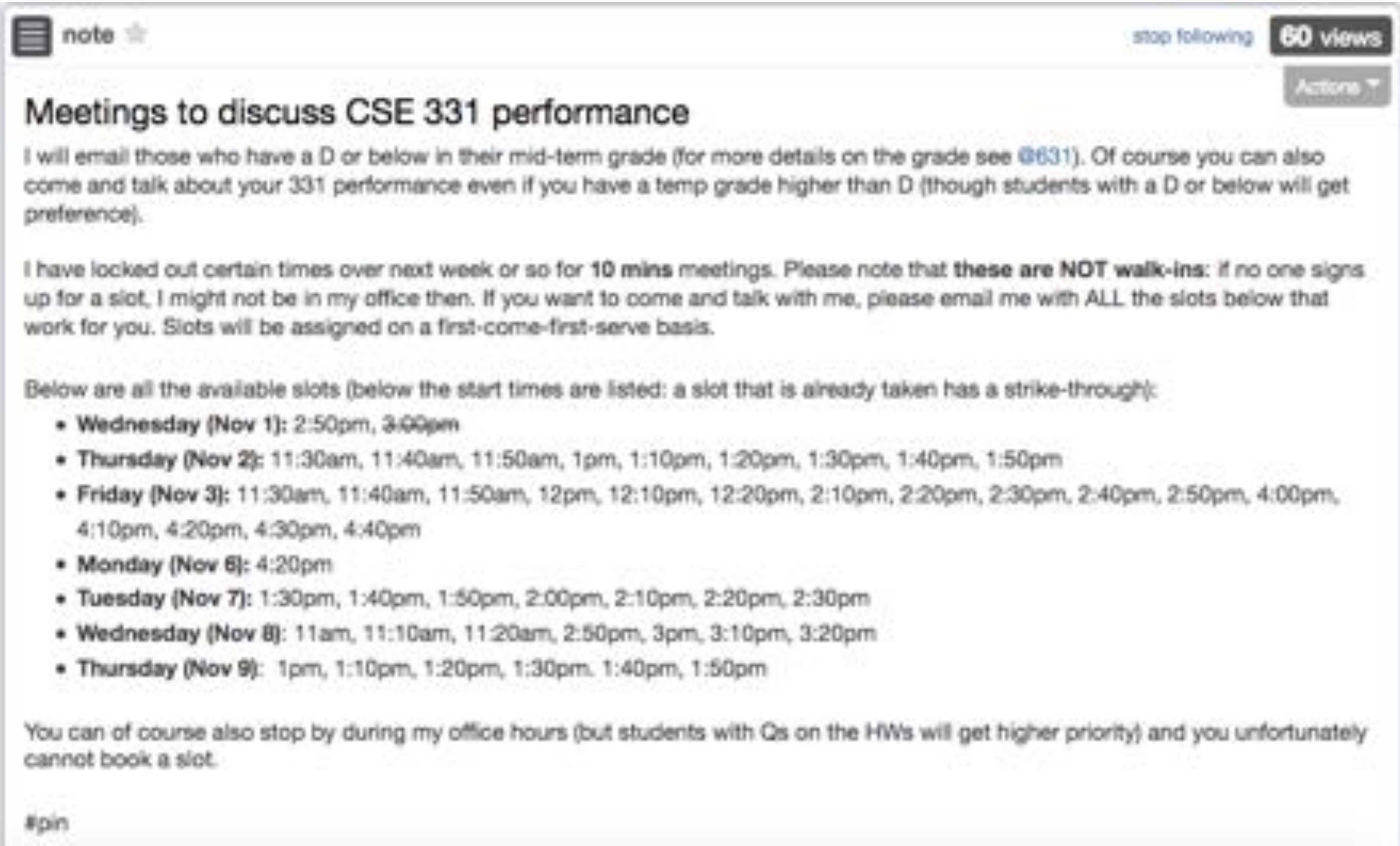
Then R is calculated as follows (out of a maximum possible of 58.5:

$$R = \frac{31}{400} \cdot H + Q \cdot \frac{1.5}{10} + \frac{M}{4} + \frac{P}{100}.$$

(I know the above does not fully following the grading rubric since it does not drop any HW score and does not substitute the quiz score with the HW score if you better on the latter. However, since this is just supposed to give you an idea of where you stand in the course, I think the above is fine as a proxy.)

Here are the stats of the raw score:

One-on-one meetings



Meetings to discuss CSE 331 performance

I will email those who have a D or below in their mid-term grade (for more details on the grade see @631). Of course you can also come and talk about your 331 performance even if you have a temp grade higher than D (though students with a D or below will get preference).

I have locked out certain times over next week or so for **10 mins** meetings. Please note that **these are NOT walk-ins:** if no one signs up for a slot, I might not be in my office then. If you want to come and talk with me, please email me with ALL the slots below that work for you. Slots will be assigned on a first-come-first-serve basis.

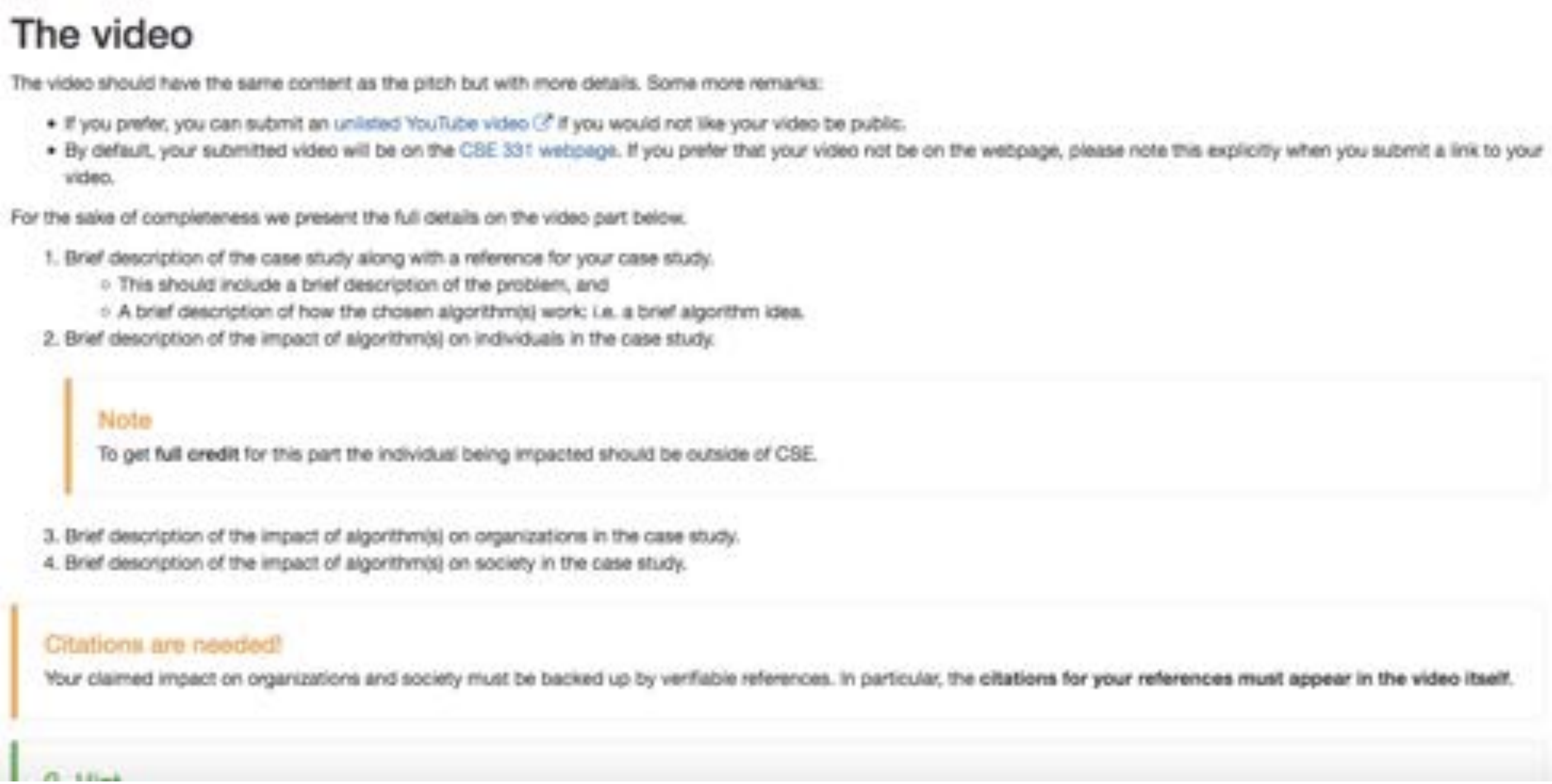
Below are all the available slots (below the start times are listed: a slot that is already taken has a strike-through):

- **Wednesday (Nov 1):** 2:50pm, ~~3:00pm~~
- **Thursday (Nov 2):** 11:30am, 11:40am, 11:50am, 1pm, 1:10pm, 1:20pm, 1:30pm, 1:40pm, 1:50pm
- **Friday (Nov 3):** 11:30am, 11:40am, 11:50am, 12pm, 12:10pm, 12:20pm, 2:10pm, 2:20pm, 2:30pm, 2:40pm, 2:50pm, 4:00pm, 4:10pm, 4:20pm, 4:30pm, 4:40pm
- **Monday (Nov 6):** 4:20pm
- **Tuesday (Nov 7):** 1:30pm, 1:40pm, 1:50pm, 2:00pm, 2:10pm, 2:20pm, 2:30pm
- **Wednesday (Nov 8)**: 11am, 11:10am, 11:20am, 2:50pm, 3pm, 3:10pm, 3:20pm
- **Thursday (Nov 9)**: 1pm, 1:10pm, 1:20pm, 1:30pm. 1:40pm, 1:50pm

You can of course also stop by during my office hours (but students with Qs on the HWs will get higher priority) and you unfortunately cannot book a slot.

#pin

Mini project video due ~1.5 weeks



The video

The video should have the same content as the pitch but with more details. Some more remarks:

- If you prefer, you can submit an unlisted YouTube video if you would not like your video be public.
- By default, your submitted video will be on the CSE 331 webpage. If you prefer that your video not be on the webpage, please note this explicitly when you submit a link to your video.

For the sake of completeness we present the full details on the video part below.

1. Brief description of the case study along with a reference for your case study.
   - This should include a brief description of the problem, and
   - A brief description of how the chosen algorithm(s) work; i.e. a brief algorithm idea.
2. Brief description of the impact of algorithm(s) on individuals in the case study.

> Note
>
> To get **full credit** for this part the individual being impacted should be outside of CSE.

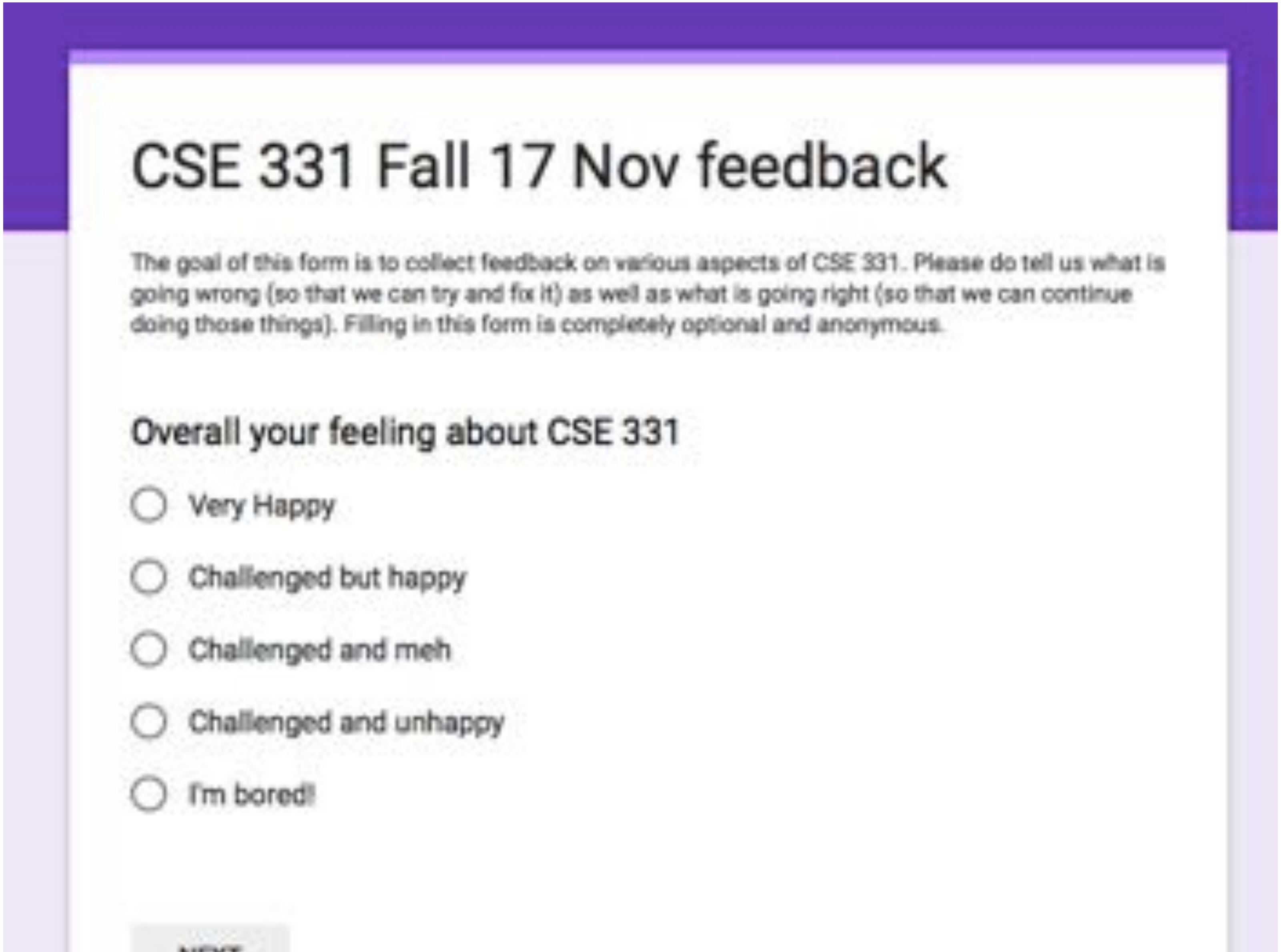
3. Brief description of the impact of algorithm(s) on organizations in the case study.
4. Brief description of the impact of algorithm(s) on society in the case study.

> Citations are needed!
>
> Your claimed impact on organizations and society must be backed up by verifiable references. In particular, the **citations for your references must appear in the video itself**.

Q. Hint

http://www-student.cse.buffalo.edu/~atri/cse331/fall17/mini-project/index.html

Give us feedback!



CSE 331 Fall 17 Nov feedback

The goal of this form is to collect feedback on various aspects of CSE 331. Please do tell us what is going wrong (so that we can try and fix it) as well as what is going right (so that we can continue doing those things). Filling in this form is completely optional and anonymous.

Overall your feeling about CSE 331

○ Very Happy

○ Challenged but happy

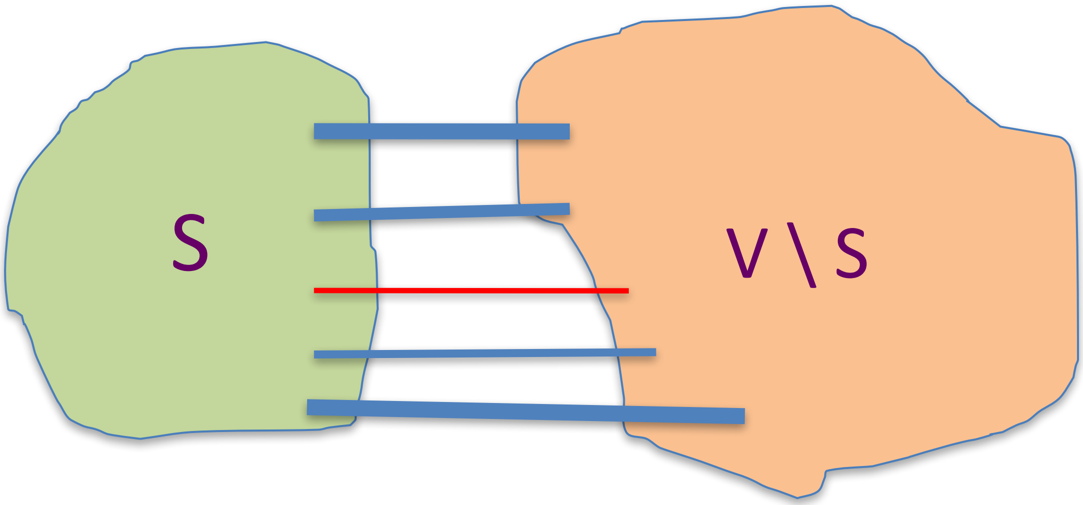
○ Challenged and meh

○ Challenged and unhappy

○ I'm bored!

Cut Property Lemma for MSTs

Condition: S and V\S are non-empty

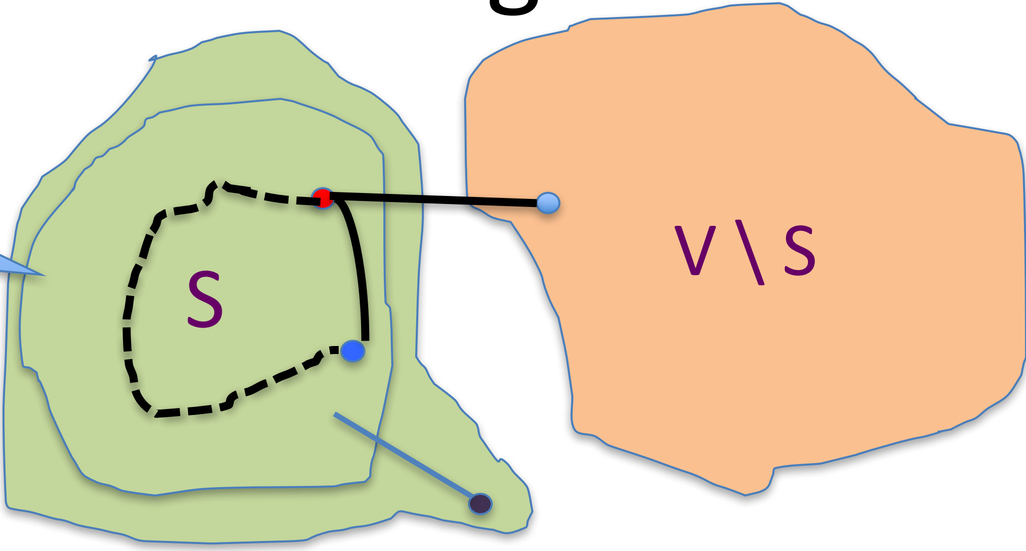


Cheapest crossing edge is in **all** MSTs

Assumption: All edge costs are distinct

Optimality of Kruskal’s Algorithm

Nodes connected to red in (V,T)



Input: $G=(V,E)$, $c_e > 0$ for every e in E

$T = \emptyset$

Sort edges in increasing order of their cost

Consider edges in sorted order

If an edge can be added to T without adding a cycle then add it to T

S is non-empty

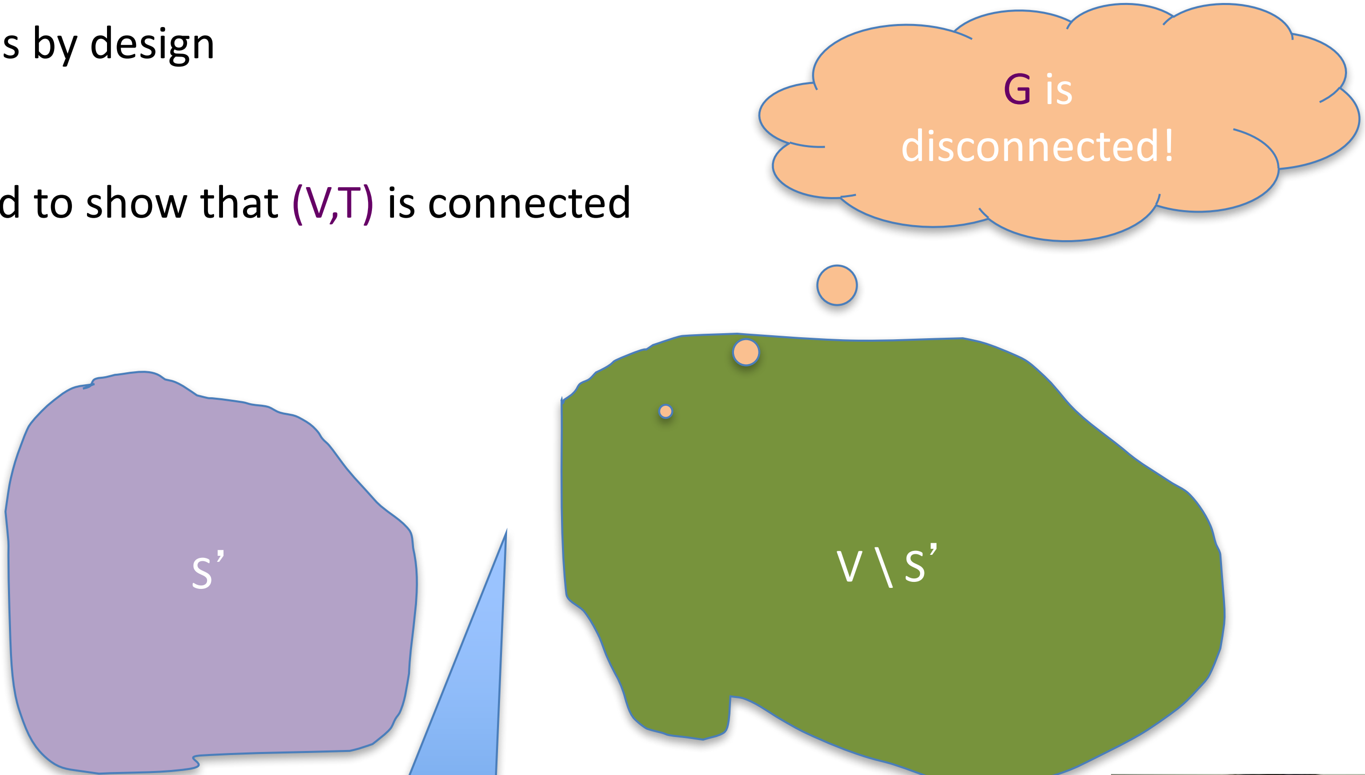
V\S is non-empty

First crossing edge considered

Is (V,T) a spanning tree?

No cycles by design

Just need to show that (V,T) is connected



G is disconnected!

S’

V \ S’

No edges here

Removing distinct cost assumption

Change all edge weights by very small amounts

Make sure that all edge weights are distinct

MST for “perturbed” weights is the same as for original

Changes have to be small enough so that this holds

EXERCISE: Figure out how to change costs

Running time for Prim’s algorithm

Similar to Dijkstra’s algorithm

$O(m \log n)$

```
Input: $G=(V,E)$, $c_e > 0$ for every $e$ in $E$
$S = \{s\}$, $T = \emptyset$
While $S$ is not the same as $V$
    Among edges $e = (u,w)$ with $u$ in $S$ and $w$ not in $S$, pick one with minimum cost
    Add $w$ to $S$, $e$ to $T$
```

Running time for Kruskal’s Algorithm

Can be implemented in O(m log n) time (Union-find DS)

Input: $G=(V,E)$, $c_e > 0$ for every e in E

$T = \emptyset$

Sort edges in increasing order of their cost

Consider edges in sorted order

If an edge can be added to T without adding a cycle then add it to T

O(m²) time overall

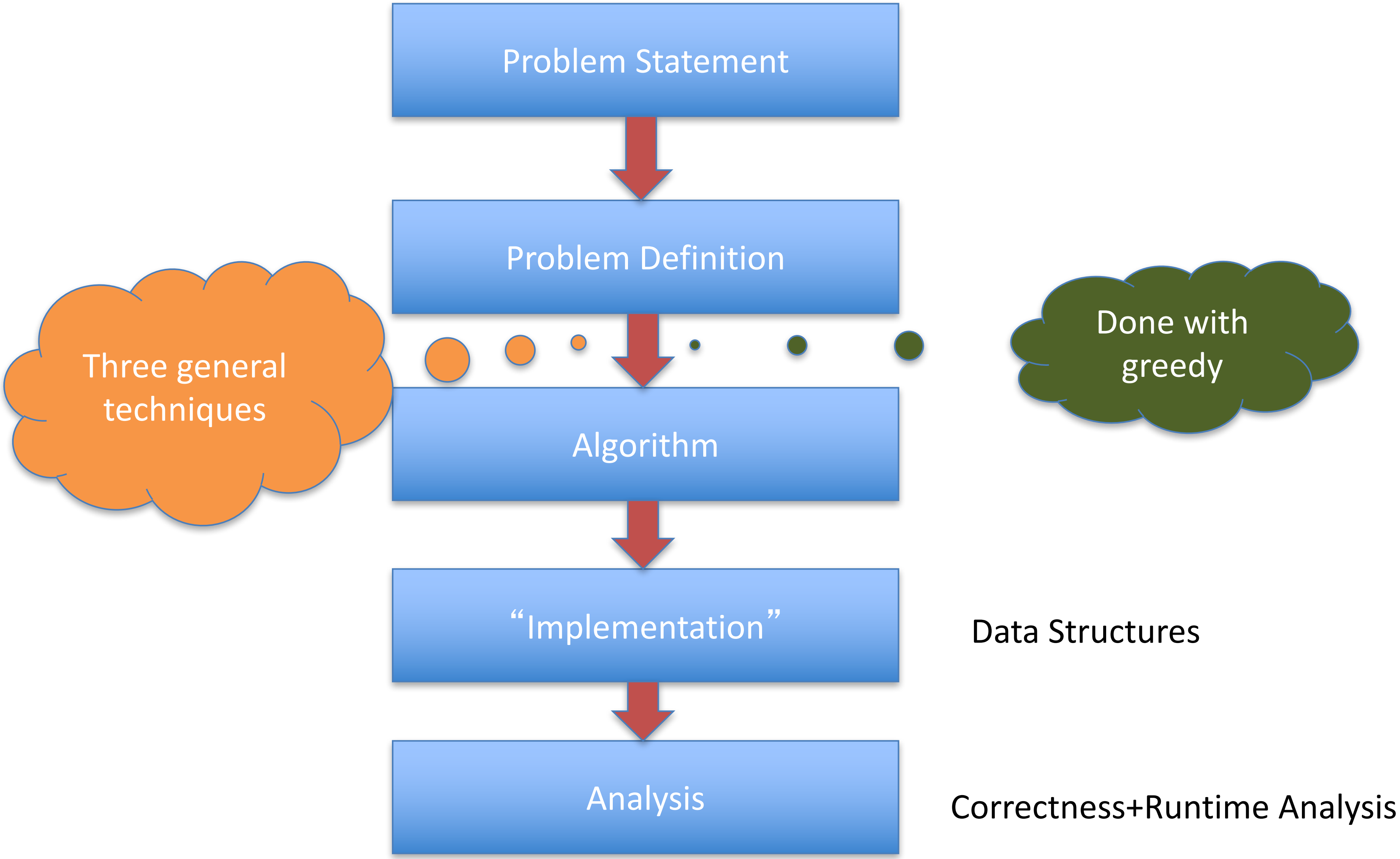
Can be verified in O(m+n) time

Joseph B. Kruskal

Reading Assignment

Sec 4.5, 4.6 of [KT]

High Level view of the course



Problem Statement

Problem Definition

Algorithm

“Implementation”

Analysis

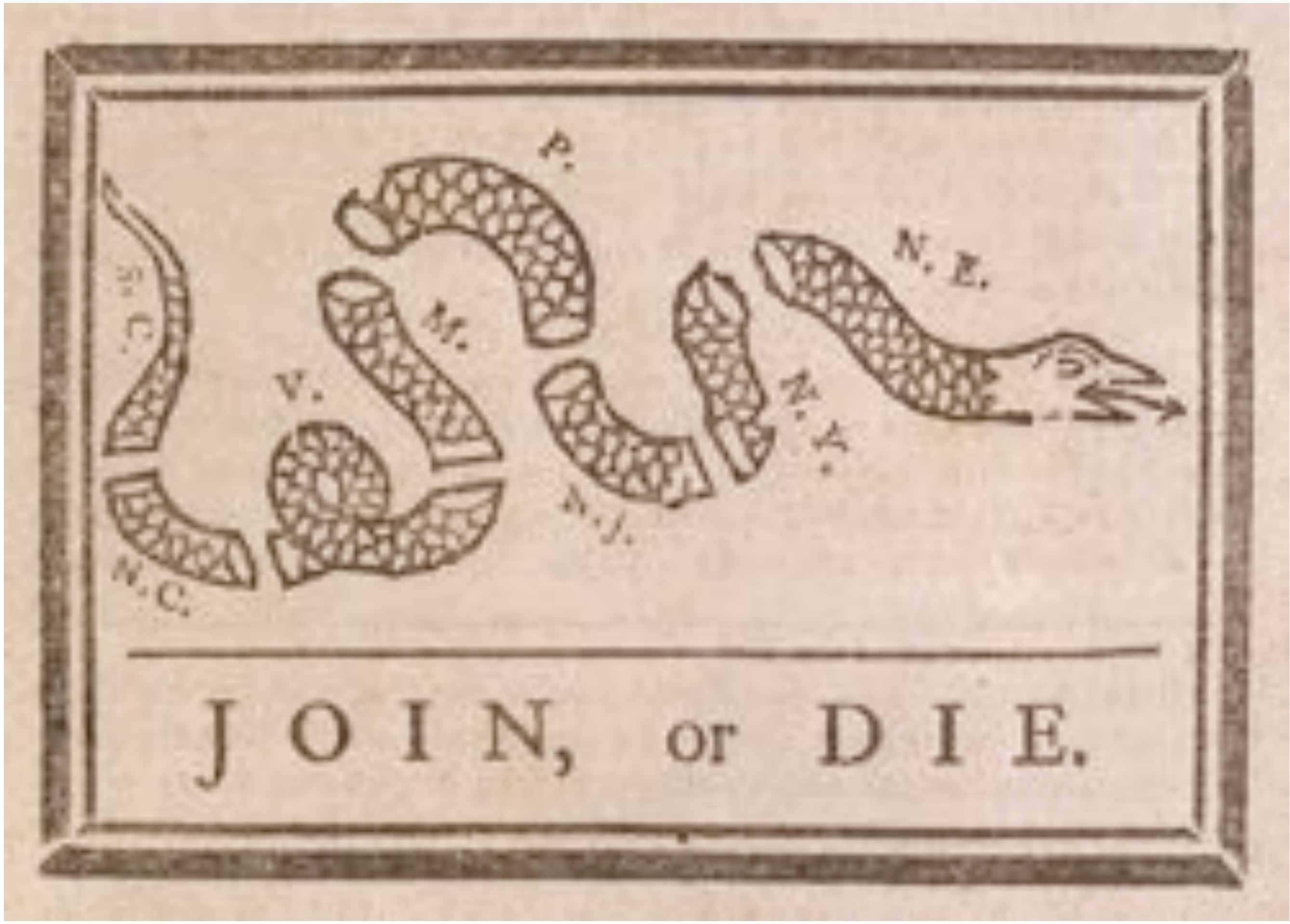
Three general techniques

Done with greedy

Data Structures

Correctness+Runtime Analysis

Trivia



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

Sorting

Given n numbers order them from smallest to largest

Works for any set of elements on which there is a total order

Insertion Sort

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

Output: $b_1,b_2,\dots,b_n$

```
$b_1 = a_1$
for i =2 … n
    Find $1 \le j \le i$ s.t. $a_i$ lies between $b_{j-1}$ and $b_j$
    Move $b_j$ to $b_{i-1}$ one cell "down"
    $b_j=a_i$
```

O(n^2) overall

O(log n)

O(n)

Make sure that all the processed numbers are sorted

a	b
4	1 2 3 4
3	2 3 4
2	3 4
1	4

Other $O(n^2)$ sorting algorithms

Selection Sort: In every round pick the min among remaining numbers

Bubble sort: The smallest number "bubbles" up

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

Mergesort Algorithm

Divide up the numbers in the middle

Unless n=2

Sort each half recursively

Merge the two sorted halves into one sorted output

How fast can sorted arrays be merged?

Group talk time

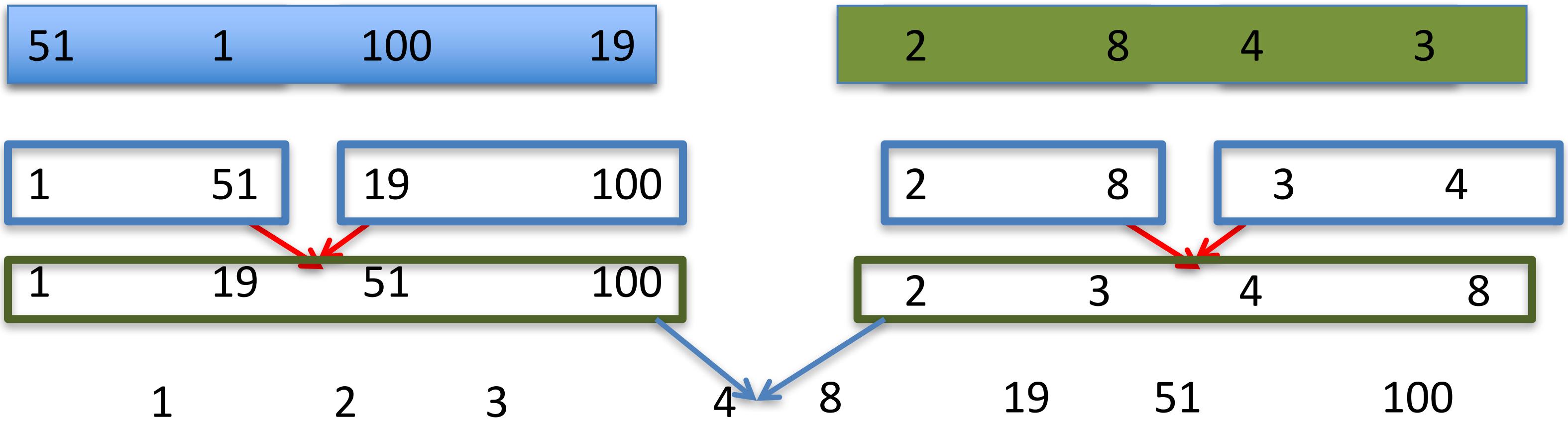
Mergesort algorithm

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

Output: Numbers in sorted order

```
MergeSort( a, n )

    If $n = 1$ **return** the order $a_1$
    If $n = 2$ **return** the order $\min(a_1,a_2); \max(a_1,a_2)$

    $a_L = a_1,\ldots, a_{n/2}$

    $a_R = a_{n/2+1},\ldots, a_n$

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

An example run



51 1 100 19

2 8 4 3

1 51

19 100

2 8

3 4

1 19 51 100

2 3 4 8

1 2 3 4 8 19 51 100

```
MergeSort( a, n )

    If $n = 1$ return the order $a_1$
    If $n = 2$ return the order $\min(a_1,a_2); \max(a_1,a_2)$
    $a_L = a_1,\ldots, a_{n/2}$

    $a_R = a_{n/2+1},\ldots, a_n$

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

Correctness

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

Output: Numbers in sorted order

```
MergeSort( a, n )
    If n = 1 return the order $a_1$
    If n = 2 return the order $\min(a_1,a_2); \max(a_1,a_2)$
    $a_L = a_1,\ldots, a_{n/2}$
    $a_R = a_{n/2+1},\ldots, a_n$
    return MERGE ( MergeSort($a_L$, n/2), MergeSort($a_R$, n/2) )
```

By induction on n

Inductive step follows from correctness of MERGE