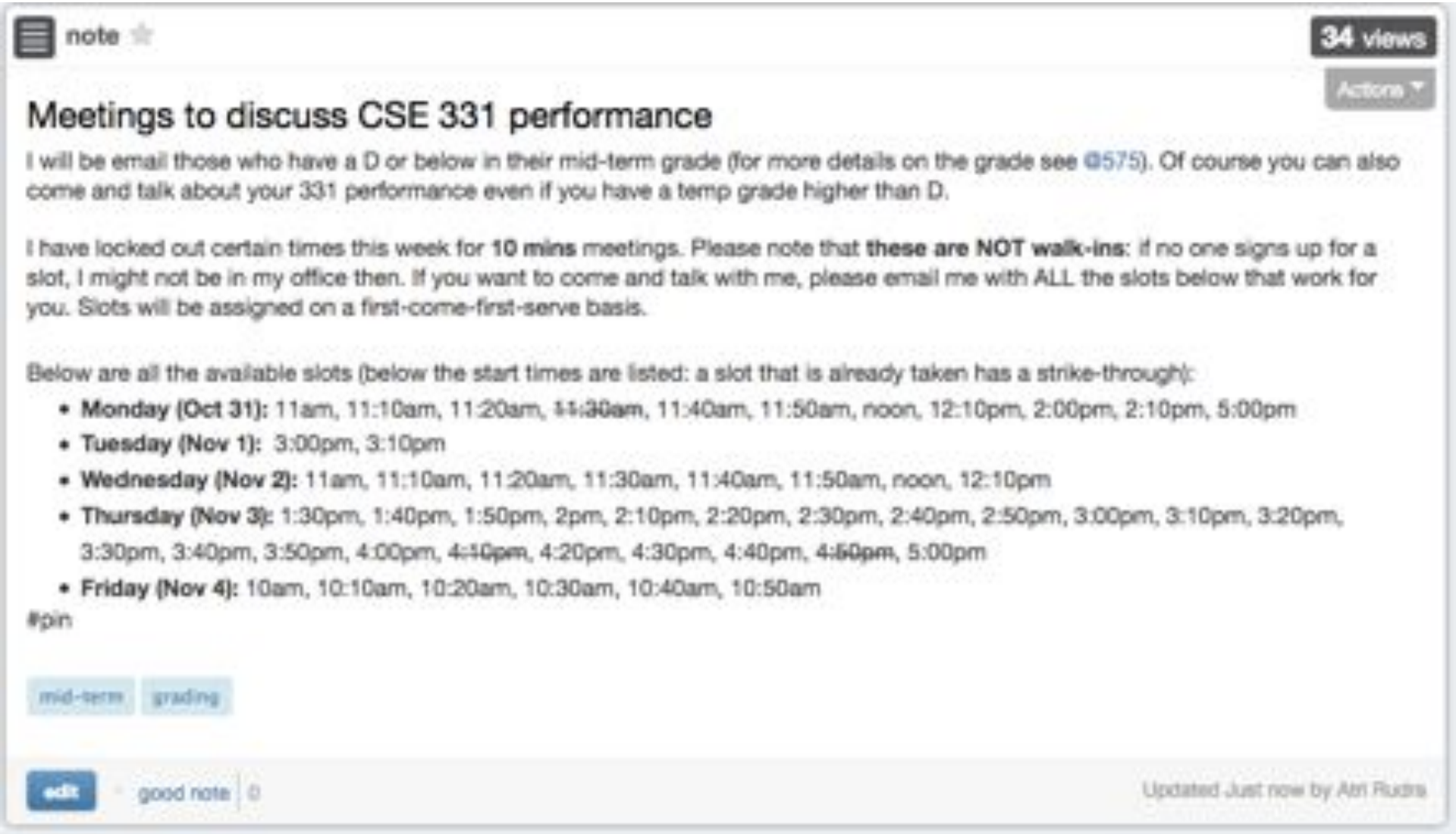
Lecture 26

CSE 331

Nov 2, 2016

One-on-one meetings



note

34 views

Actions

Meetings to discuss CSE 331 performance

I will be email those who have a D or below in their mid-term grade (for more details on the grade see @575). Of course you can also come and talk about your 331 performance even if you have a temp grade higher than D.

I have locked out certain times this week 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):

- **Monday (Oct 31):** 11am, 11:10am, 11:20am, ~~11:30am~~, 11:40am, 11:50am, noon, 12:10pm, 2:00pm, 2:10pm, 5:00pm
- **Tuesday (Nov 1):** 3:00pm, 3:10pm
- **Wednesday (Nov 2):** 11am, 11:10am, 11:20am, 11:30am, 11:40am, 11:50am, noon, 12:10pm
- **Thursday (Nov 3):** 1:30pm, 1:40pm, 1:50pm, 2pm, 2:10pm, 2:20pm, 2:30pm, 2:40pm, 2:50pm, 3:00pm, 3:10pm, 3:20pm, 3:30pm, 3:40pm, 3:50pm, 4:00pm, ~~4:10pm~~, 4:20pm, 4:30pm, 4:40pm, ~~4:50pm~~, 5:00pm
- **Friday (Nov 4):** 10am, 10:10am, 10:20am, 10:30am, 10:40am, 10:50am

#pin

mid-term grading

edit good note 0

Updated Just now by Ari Rudra

Mini project video due ~1.5 weeks

note

stop following 90 views

Actions

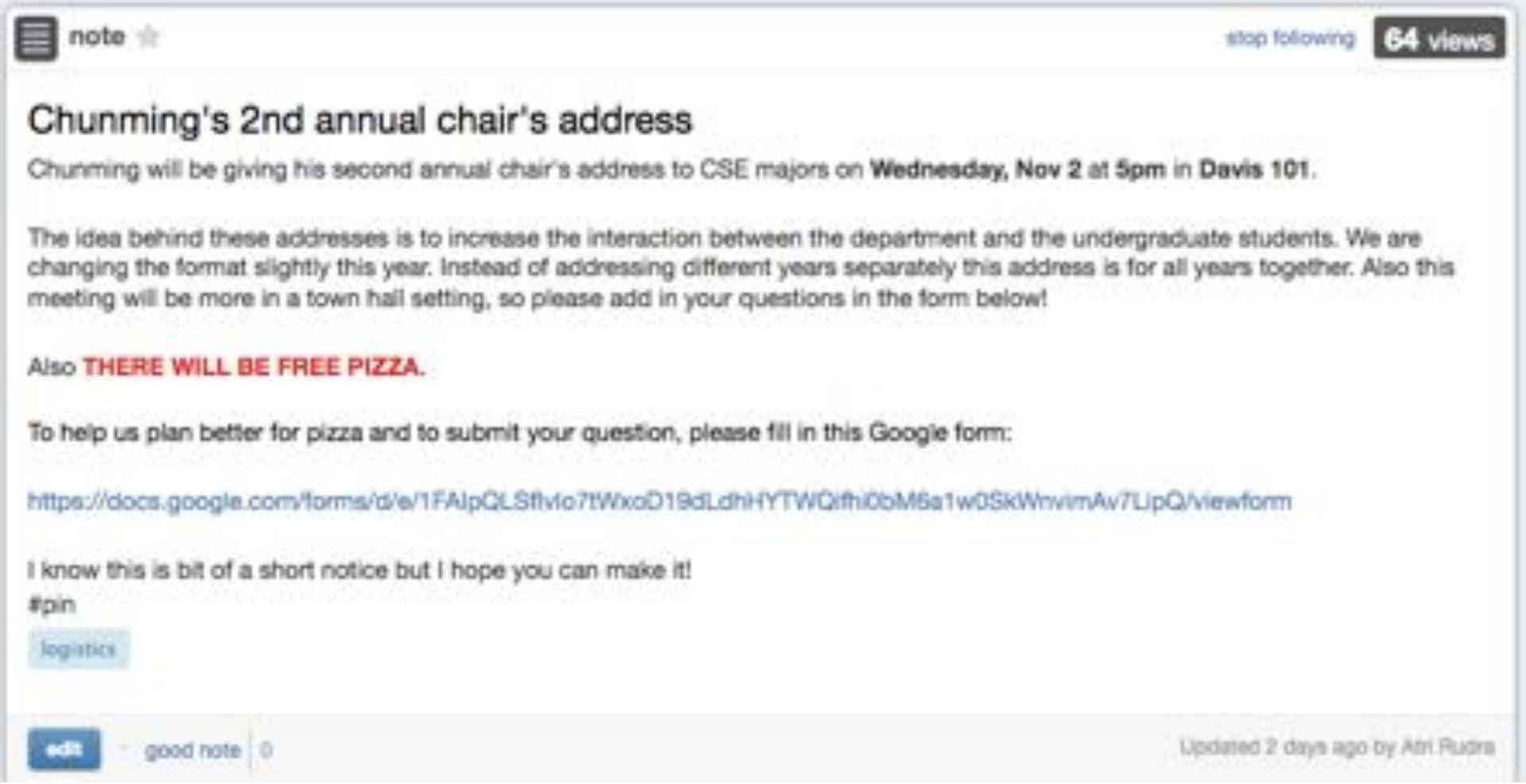
Mini project video

Sorry for the delay in posting this information. For the basics, please see the mini-project page.

Below are the main logistics. **IT IS IMPORTANT TO READ THESE CAREFULLY SINCE NOT FOLLOWING INSTRUCTION COULD LEAD TO LOSS OF ALL POINTS.**

- The deadline is **Monday, November 14, 11:59pm**. You can start submitting on Autolab anytime from now till the deadline.
- You will need to need to form your group on Autolab again for this submission. See @304 for instructions on how to do it.
  - ***Very* important**: Please make sure you submit your group's submission **after** the group has been formed. If this is not done, the **entire group will get a zero**.
    - No excuses on this-- make sure you do this group formation well in advance. If you cannot reach one of your group members at the last moment, then that is your problem.
- You will need to submit a **PDF** with the following information:
  - Link to the your group's video on Youtube
    - The video has to be for **AT MOST FIVE (5) MINS.** While grading anything beyond the 5 min mark will be completely ignored. Of course a shorter video is fine!
  - If you would prefer your groups video to be not listed on this page, please add in an explicit sentence saying so. By default, all videos will be linked to on the above page.
  - If you submit in a format other than PDF then your group will get a zero. Also make sure to preview the submitted PDF to double-check that Autolab can actually read your submitted file.
- The grading rubric is similar to the pitch but there are some small differences:

Chunming's address TODAY



note

stop following 64 views

Chunming's 2nd annual chair's address

Chunming will be giving his second annual chair's address to CSE majors on **Wednesday, Nov 2** at **5pm** in **Davis 101**.

The idea behind these addresses is to increase the interaction between the department and the undergraduate students. We are changing the format slightly this year. Instead of addressing different years separately this address is for all years together. Also this meeting will be more in a town hall setting, so please add in your questions in the form below!

Also **THERE WILL BE FREE PIZZA.**

To help us plan better for pizza and to submit your question, please fill in this Google form:

https://docs.google.com/forms/d/e/1FAIpQLSfIvIo7tWxoD19dLdhHYTWQifhi0bM6a1w0SkWnvimAv7LipQ/viewform

I know this is bit of a short notice but I hope you can make it!
#pin

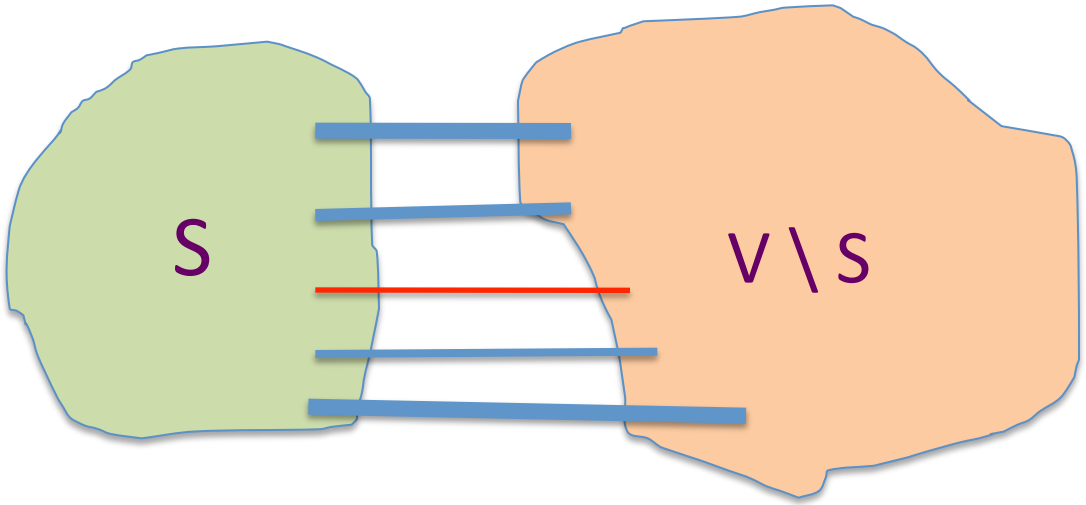
logistics

edit good note 0

Updated 2 days ago by Atri Rudra

Cut Property Lemma for MSTs

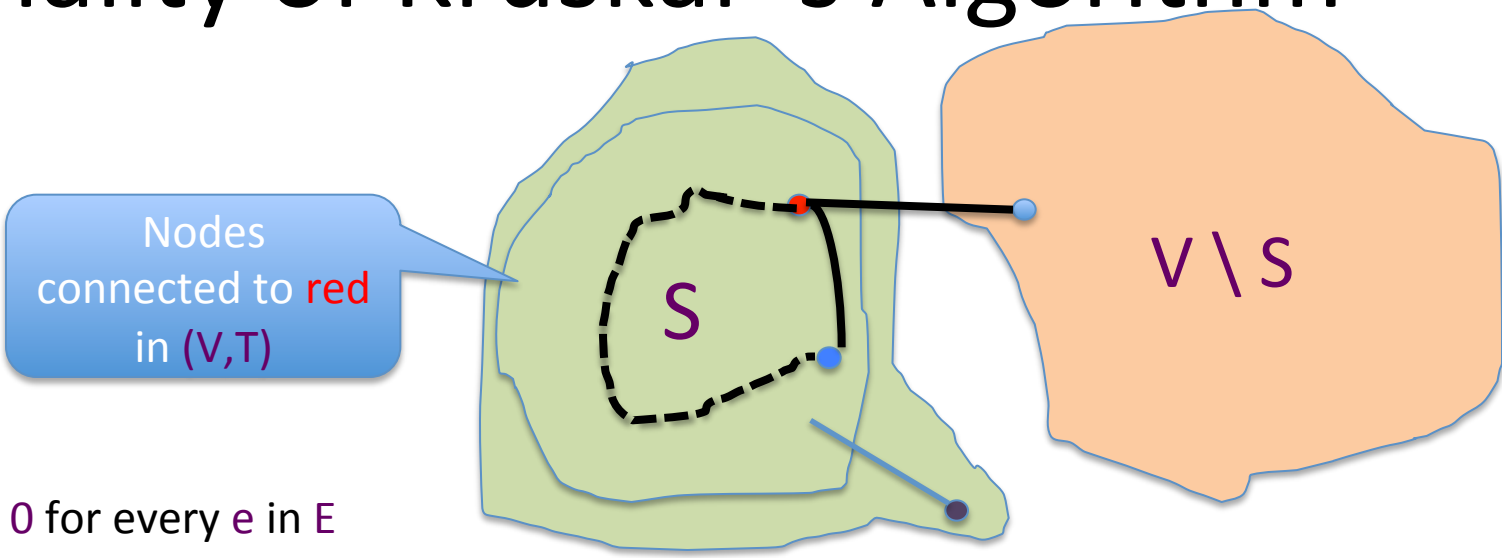
Condition: S and V\S are non-empty



Cheapest crossing edge is in **all** MSTs

Assumption: All edge costs are distinct

Optimality of Kruskal's Algorithm



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

T = Ø

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

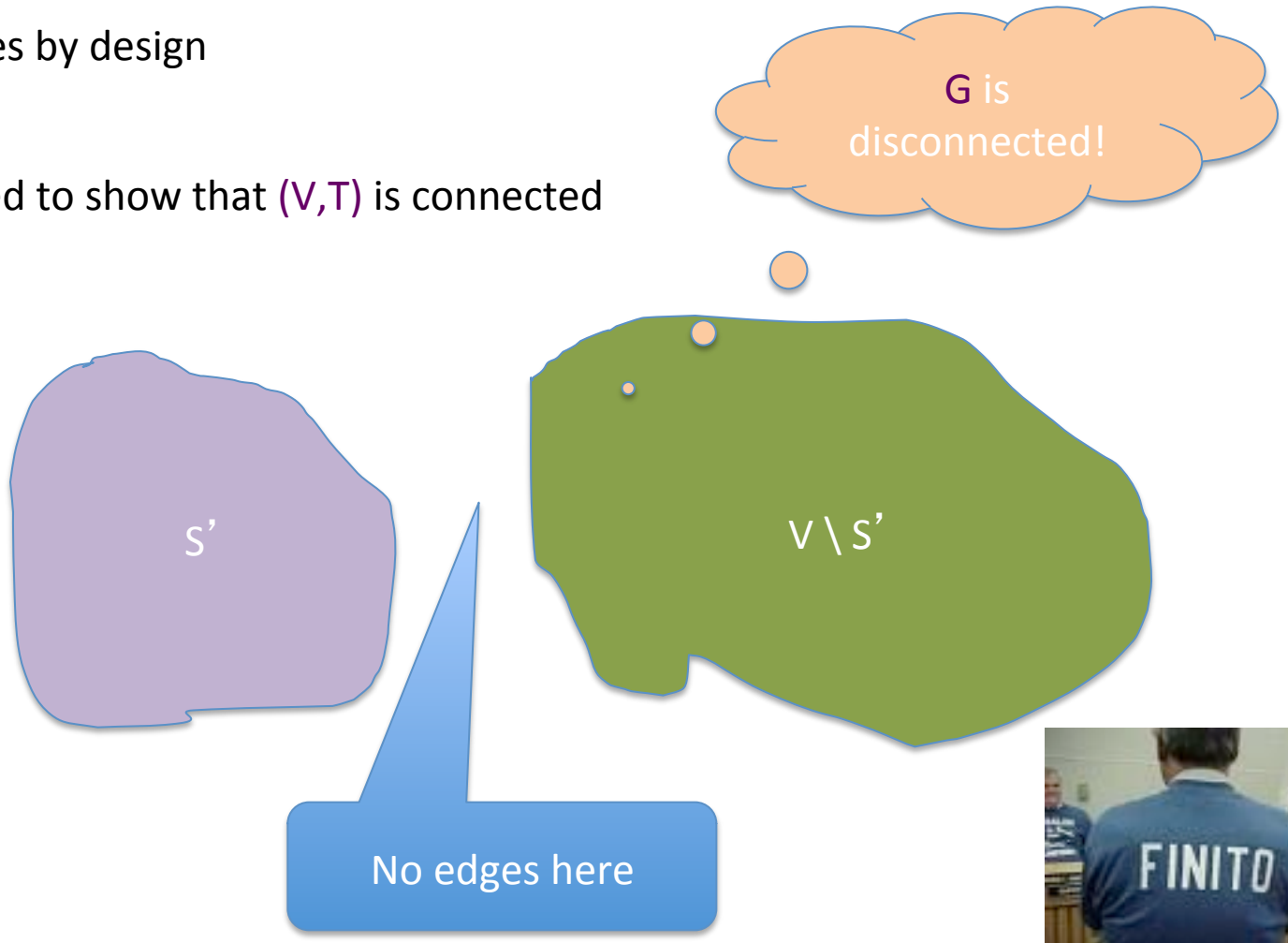
V\S is non-empty

First crossing edge considered

Is (V,T) a spanning tree?

No cycles by design

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



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 = Ø

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)

Joseph B. Kruskal

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

$O(m^2)$ time overall

T = Ø

Sort edges in increasing order of their cost

Consider edges in sorted order

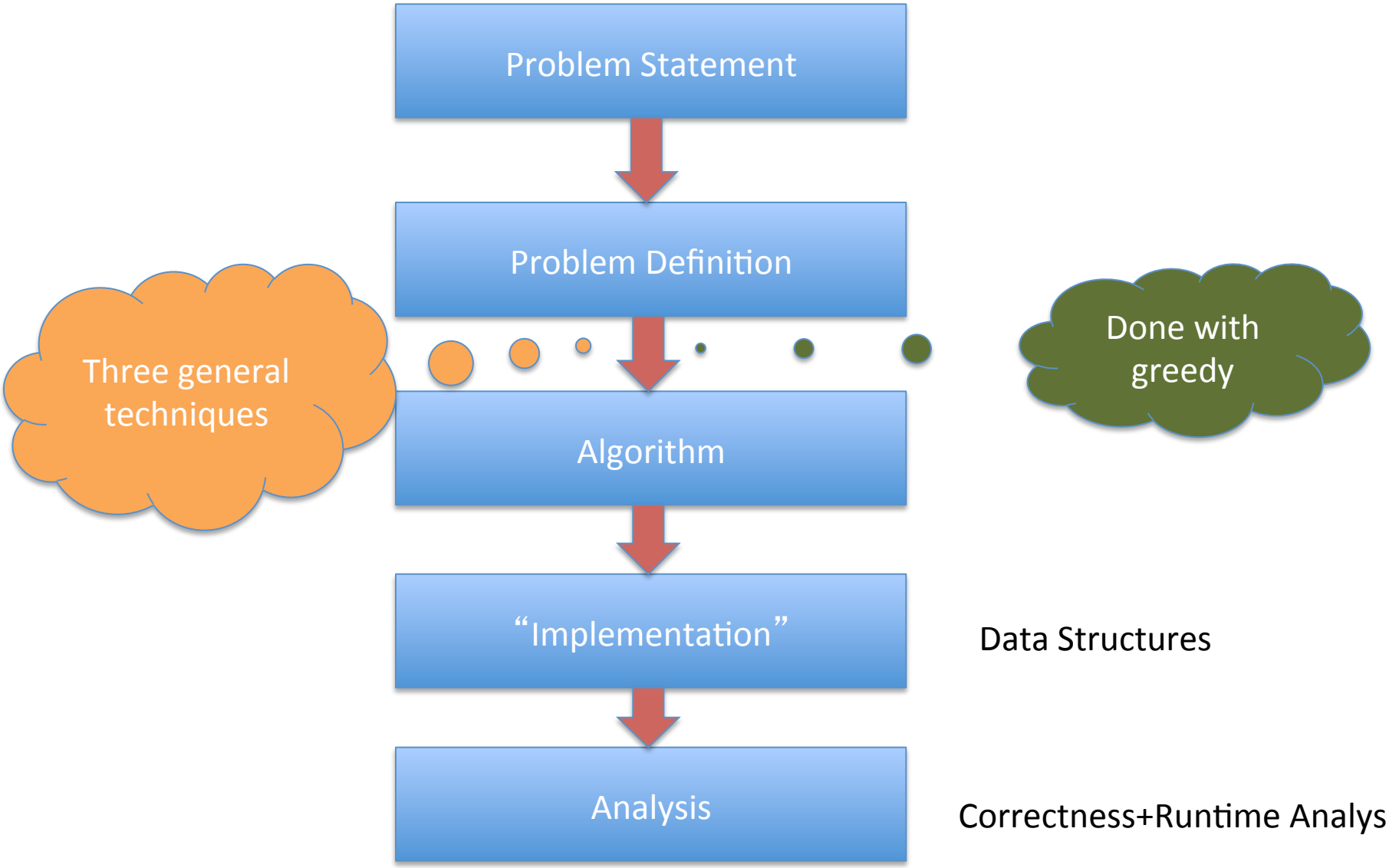
If an edge can be added to T without adding a cycle then add it to T

Can be verified in O(m+n) time

Reading Assignment

Sec 4.5, 4.6 of [KT]

High Level view of the course



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: $a_1, a_2, \ldots, a_n$

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

$O(n^2)$ overall

Make sure that all the processed numbers are sorted

$b_1 = a_1$

for $i = 2 \ldots n$

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(log n)

O(n)

a	b
4	4
3	3
2	2
1	1

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

Sort each half recursively

Merge the two sorted halves into one sorted output

Unless n=2

How fast can sorted arrays be merged?

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

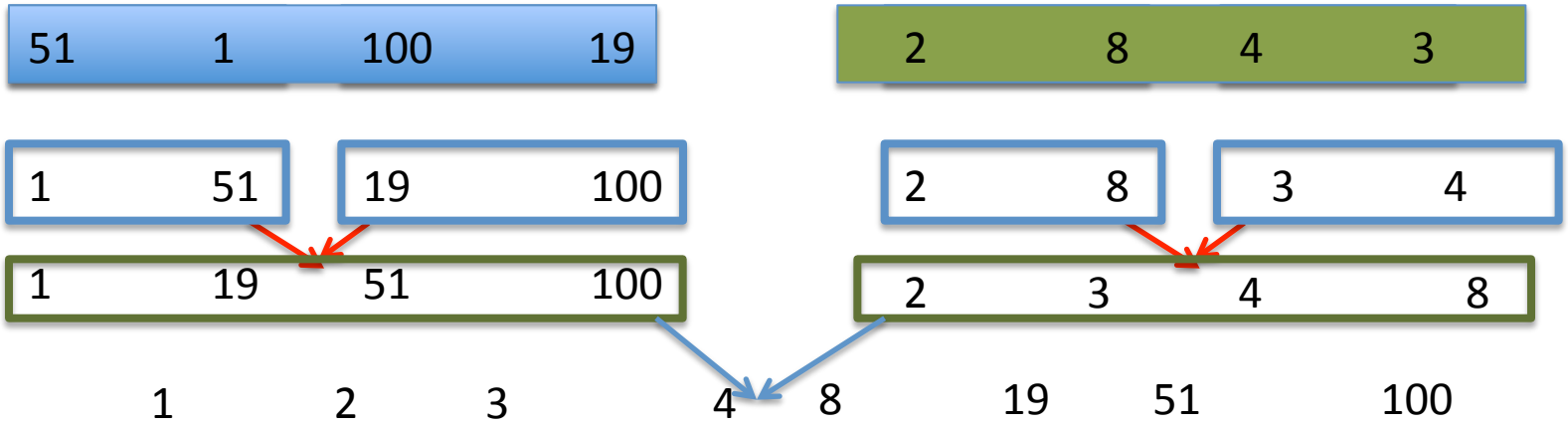
If n = 2 **return** the order $\min(a_1,a_2)$; $\max(a_1,a_2)$

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

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

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

An example run



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,..., a_n/2

  a_R = a_n/2+1,..., 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