Lecture 22

CSE 331

Oct 24, 2022

Project deadlines coming up

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

Click to add notes

Please be in touch w/ your group

note @304

stop following 84 views

Actions

Please respond to your project group mates

Please do respond back if a group project member reaches out to you to get started on the project. Just FYI, ***I always reserve the right to kick you out of your group (which means a 0 for you) in case you are unresponsive to repeated requests from your group members.***

I understand some of you might be busy now-- it is fine with me if your group decide as a whole how the work will be divided (so e.g. someone does less work on the initial problems and someone does more work on the later problems). As long as the group agrees, I do not care about the details.

But please do respond back in a timely fashion: not doing so is you not doing your part in a group project.

project

Edit good note 0

Updated 6 days ago by Atri Rudra

1-on-1 meetings

note @324 | stop following | 67 views

Actions

Meetings to discuss CSE 331 performance

By tonight, I will email those who have a D+ or below in their mid-term grade (for more details on the grade see @323) to setup a one-on-one meeting to talk with me but I figured I should post the information about meeting times now rather than later tonight.

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 **15 mins** meetings. Please note that **these are NOT walk-ins**: if no one signs up for a slot, I will NOT be on zoom then. If you want to come and talk with me, **please EMAIL me with ALL the slots below that work for you**. (Private posts on piazza will not work; please email me!) Slots will be assigned on a first-come-first-serve basis. *Also I might only be able to confirm your time after 11pm on the day before your scheduled slot.*

Note: These are my current availabilities-- some of the slots might be used up in some other non-CSE 331 meetings. So please send multiple choices for when you can meet.

To make things easier, **ALL meeting will be on zoom** ([illegible])

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

- **Thursday (Oct 20):** ~~9:00am, 9:15am, 9:30am, 9:45am, 10:00am, 10:15am, 10:30am, 10:45am, 11:00am, 11:15am, 11:30am, 11:45am, 12:00pm, 12:15pm~~, 12:30pm, 12:45pm, 1:00pm, 1:15pm, 3:15pm, 3:30pm, 3:45pm, 4:00pm, 4:15pm, 4:30pm, 4:45pm
- **Friday (Oct 21):** 9:00am, 9:15am, 9:30am, 9:45am, ~~10:00am~~, 1:30pm, 1:45pm, 4:00pm, ~~4:15pm~~, 4:30pm, 4:45pm
- **Monday (Oct 24):** 9:00am, 9:15am, 9:30am, 9:45am, 10:00am, 3:00pm, 3:15pm, ~~3:30pm~~, 3:45pm, 6:00pm, 6:15pm
- **Friday (Oct 28):** 9:00am, 9:15am, 9:30am, 9:45am, 10:00am, ~~1:30pm~~, 1:45pm, 3:00pm, 3:15pm, 3:30pm, 3:45pm

(Apologies but my schedule for the rest of this week and next is bit of a mess. If none of the times above work for you but you still want to meet, please email me and we can set up a time for the week of Oct 31)

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 during my usual office hours.

Again, **please email me your (at least top 3) choices along with preference for in-person or virtual** (again note the ALL slots are virtual).

mid-term | grading

Edit | good note | 0

Updated 9 seconds ago by Atri Rudra

Guest lecture on Wed

My 4pm
OH
canceled

Trevor
will run
late night
OH

A. Erdem Sarıyüce

One time amnesty for AI violation

note @346 · stop following · 1 view

Actions

One time amnesty option

If for whatever reason you did not follow the HW policies for any of homeworks 1-5 (e.g. you looked at sources that yoou should not have, or collaborated in a group of size >3 including yourself, etc.), we're giving an option for one time amnesty. What this means is you can withdraw any question(s) where you might have violated any HW policy (and get a 0 for the corresponding question(s)) without any academic integrity violation charges.

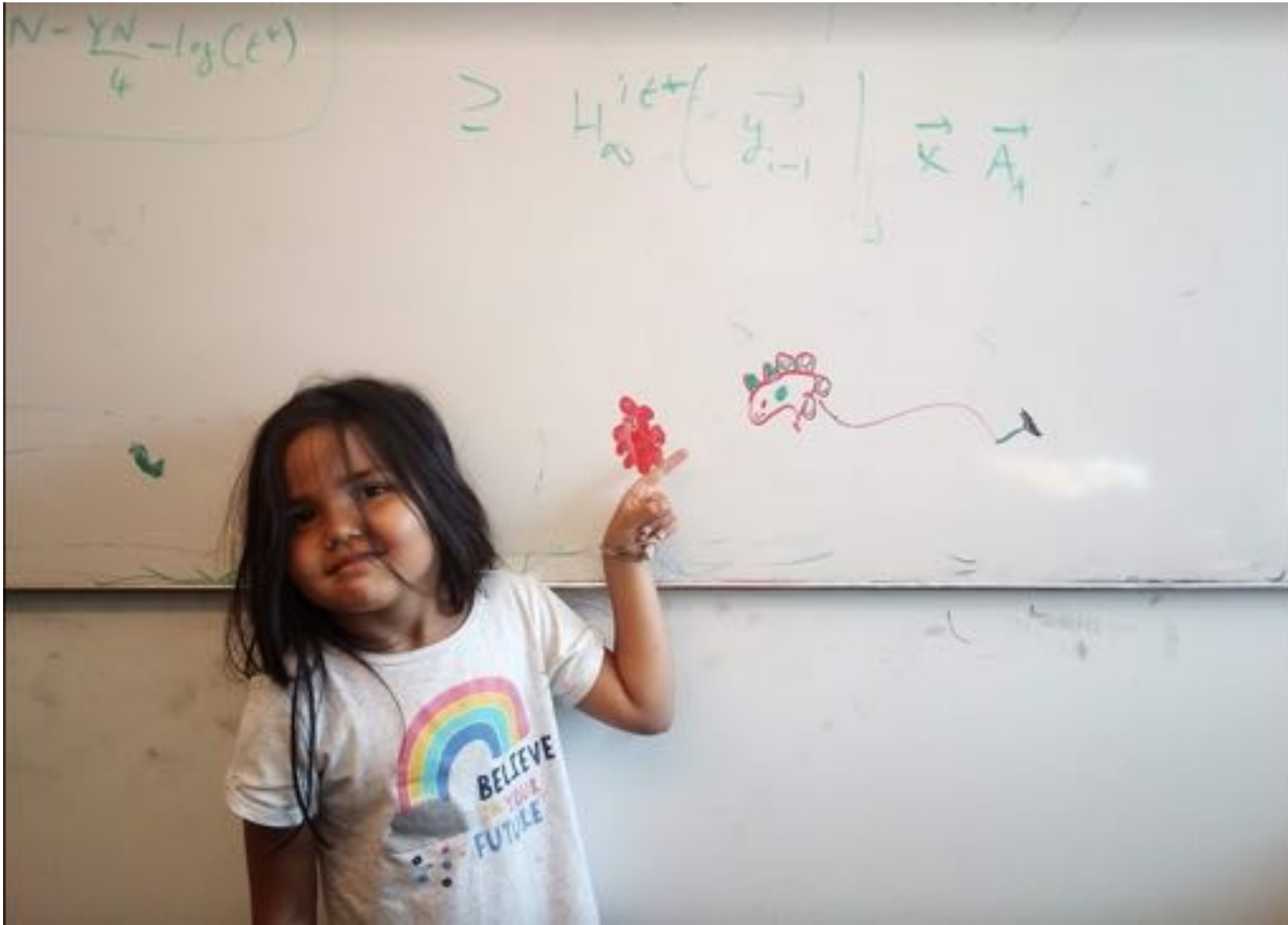
To avail of this option, **please email Atri by Thursday 5pm** letting him know which question(s) you would like to withdraw. You do NOT have to give any reason for the withdrawing the questions.

This will be the only option during the semester. If you get caught violating HW policies iin HWs 1-5 after Th 5pm (or in any future HWs), then I will start academic integrity violation procedure against you.

homework1 homework2 homework3 homework4 homework5 grading

Edit · good note | 0 · Updated 31 seconds ago by Atri Rudra

Questions/Comments?



Kruskal's Algorithm

Joseph B. Kruskal

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

$T = \emptyset$

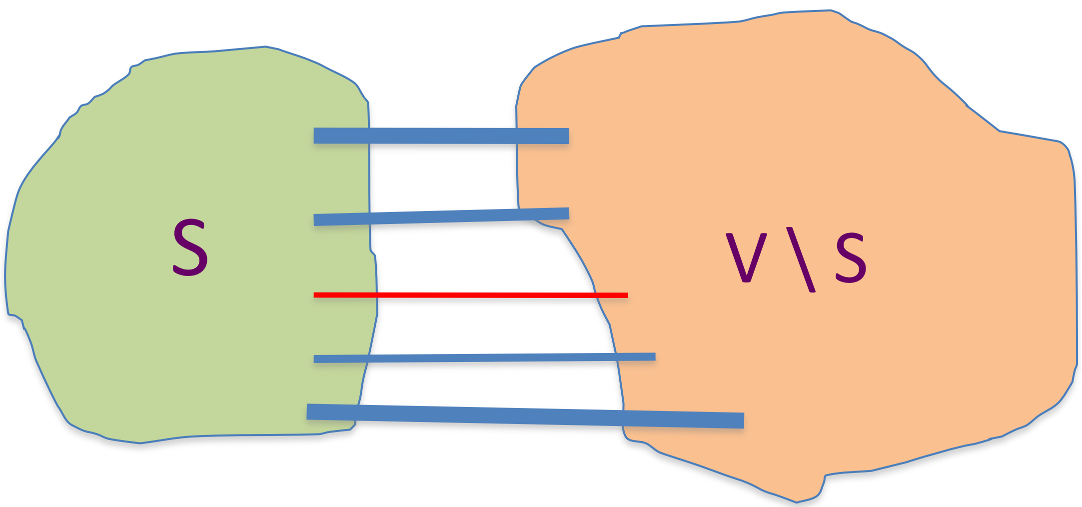
Sort edges in increasing order of their cost

Consider edges in sorted order

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

Cut Property Lemma for MSTs

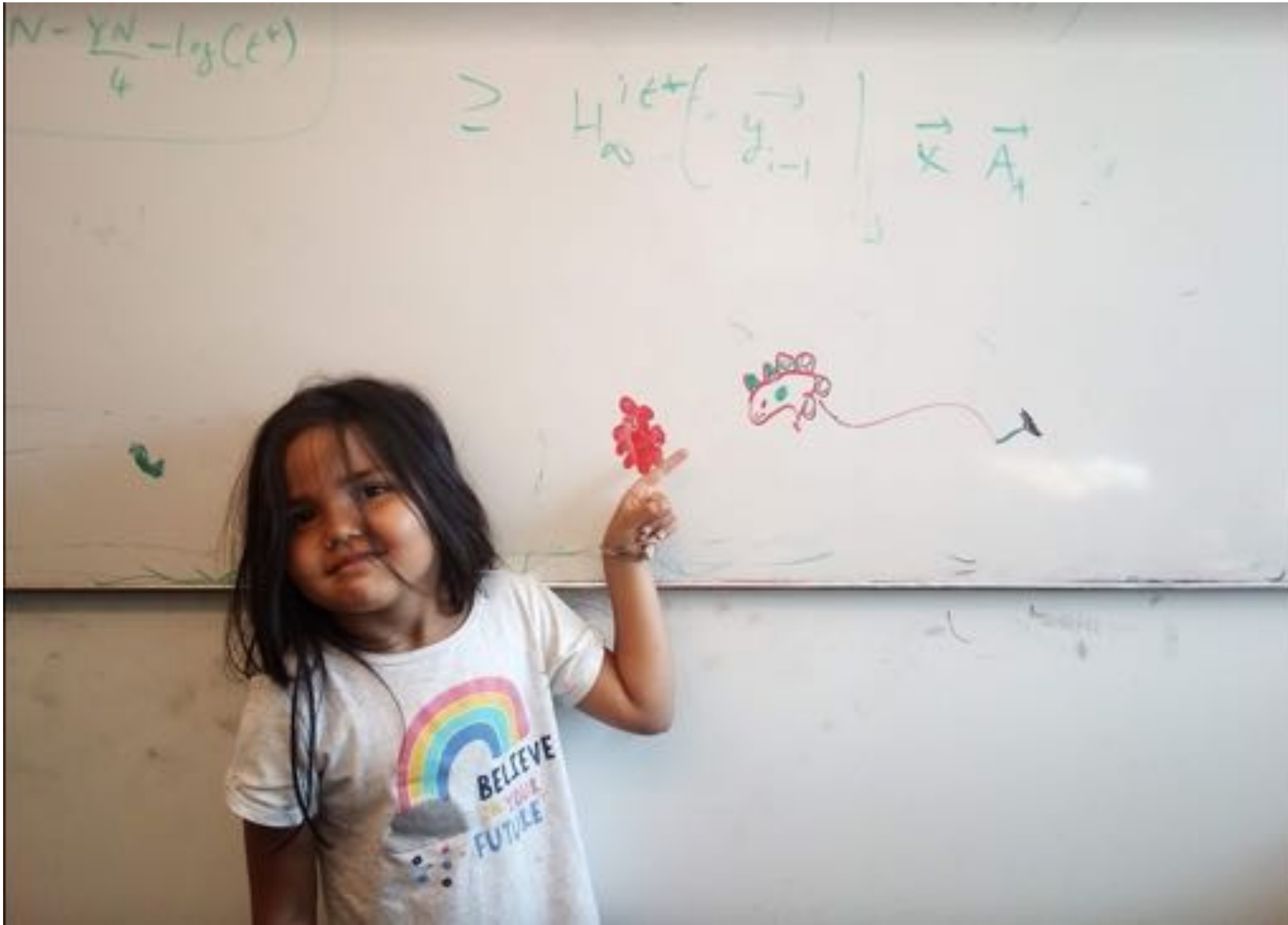
Condition: S and V\S are non-empty



Cheapest crossing edge is in **all** MSTs

Assumption: All edge costs are distinct

Questions/Comments?



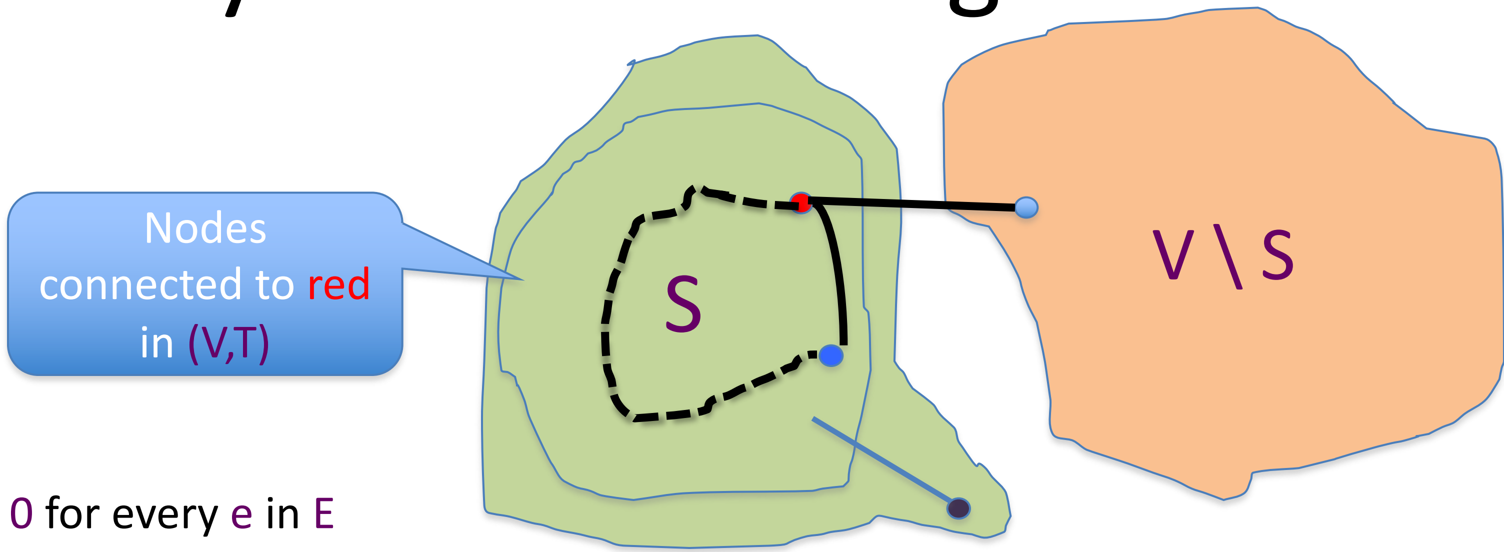
Today’s agenda

Optimality of Kruskal’s algorithm

Remove distinct edge weights assumption

Quick runtime analysis of Prim’s+Kruskal’s

Optimality of Kruskal’s Algorithm



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

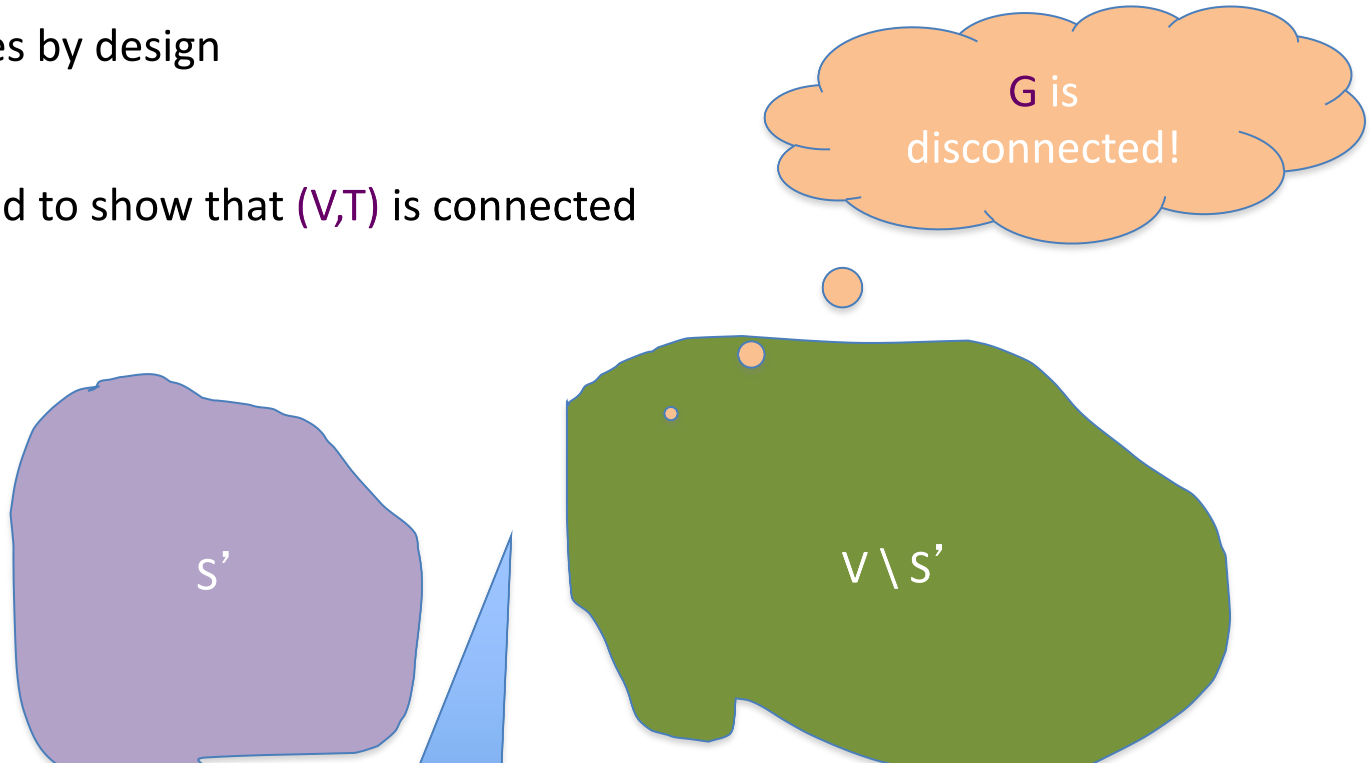
V\S is non-empty

First crossing edge considered

Is (V,T) a spanning tree?

No cycles by design

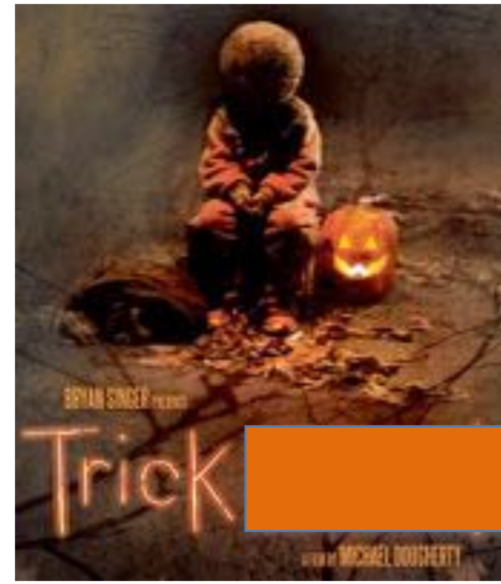
Just need to show that (V,T) is connected



Removing distinct cost assumption

Change all edge weights by very small amounts

Make sure that all edge weights are distinct

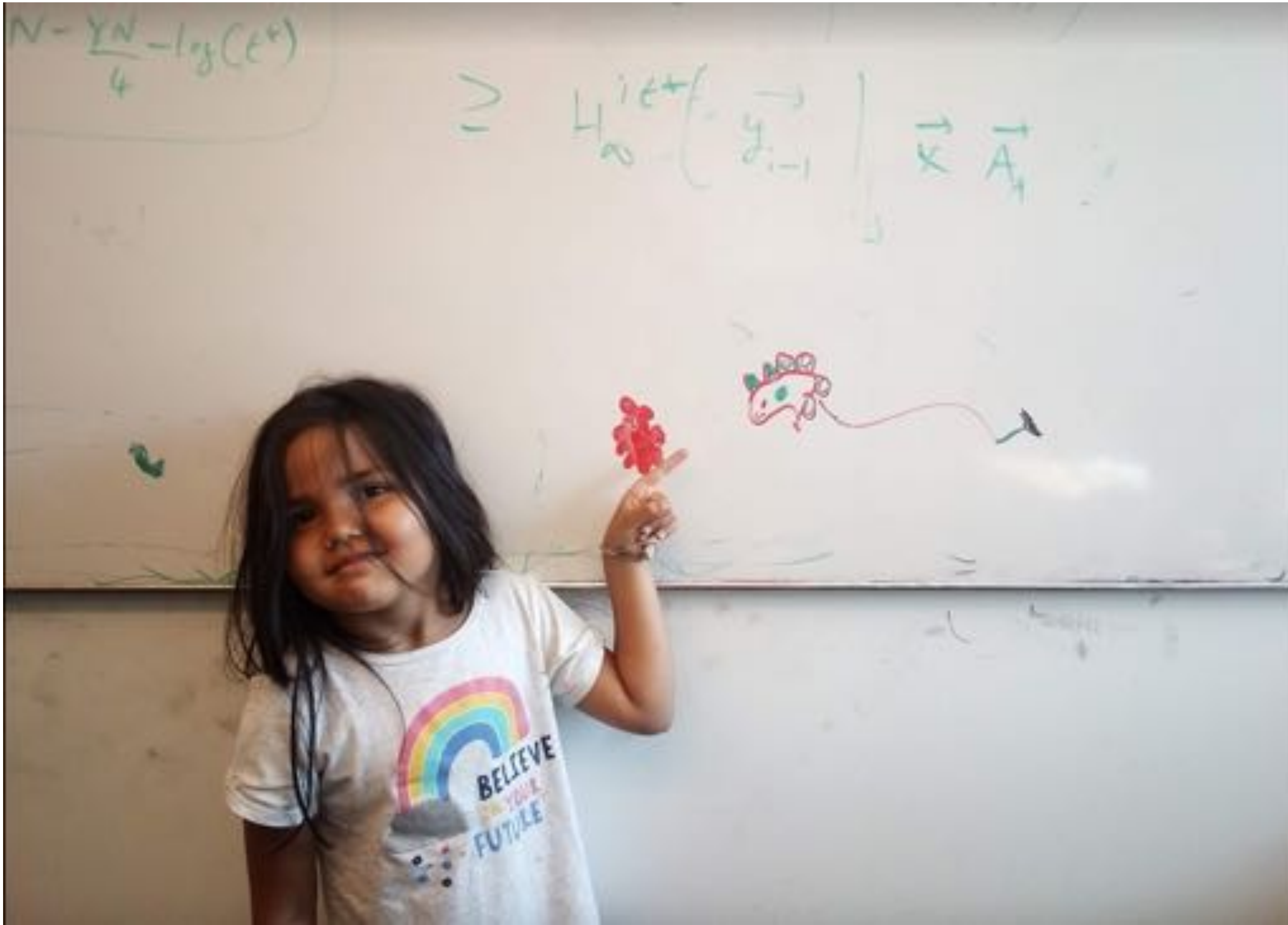


MST for "perturbed" weights is the same as for original

Changes have to be small enough so that this holds

EXERCISE: Figure out how to change costs

Questions/Comments?



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

$O(m^2)$ time overall

$T = \emptyset$

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

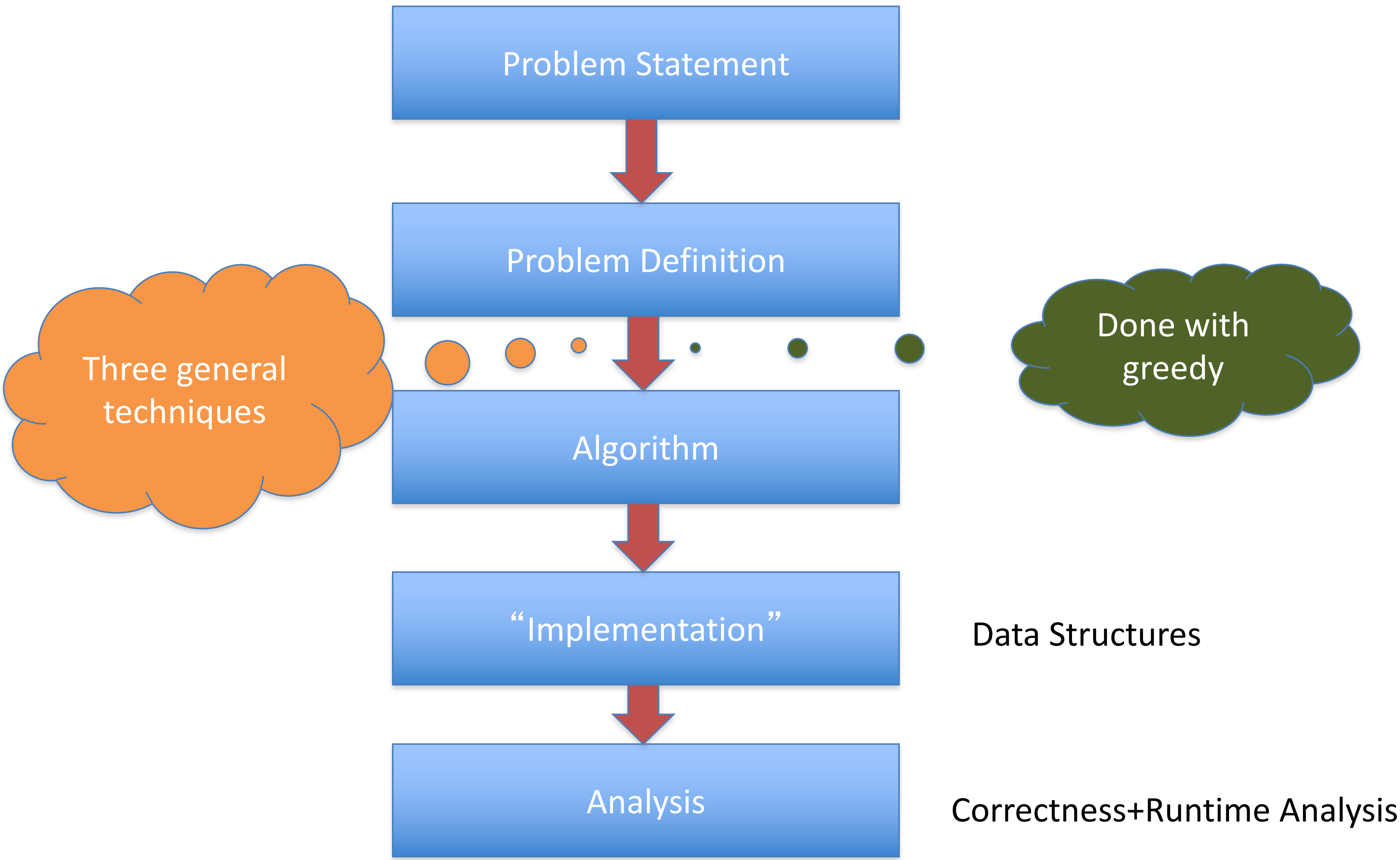
Can be verified in $O(m+n)$ time

Joseph B. Kruskal

Reading Assignment

Sec 4.5, 4.6 of [KT]

High Level view of the course



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(n²) 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	[illegible]
3	[illegible]
2	[illegible]
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)

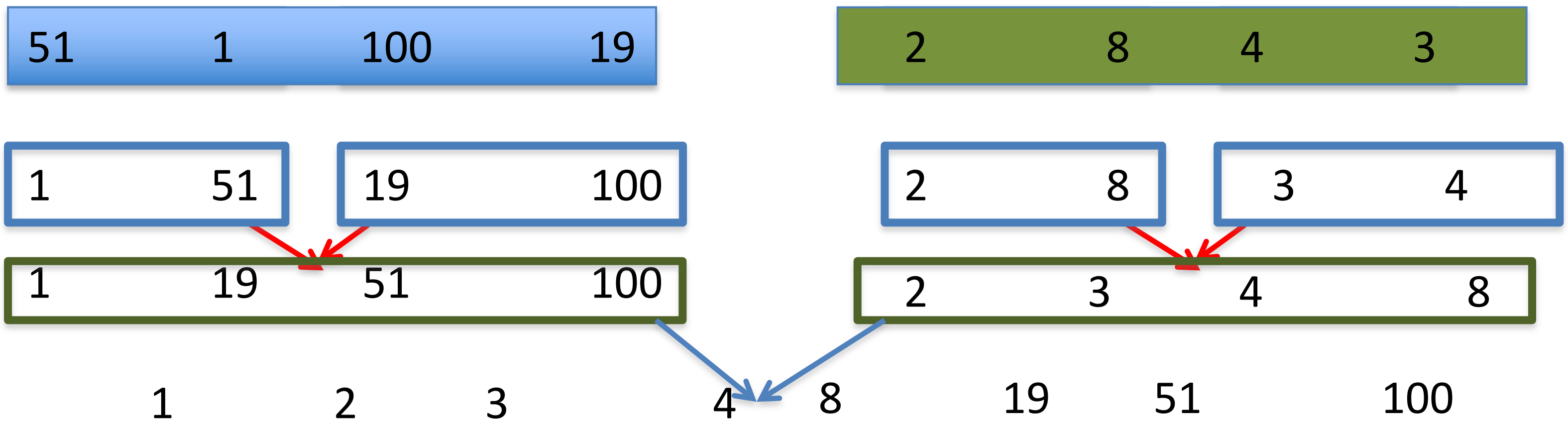
If $n = 1$ **return** the order a_1
If $n = 2$ **return** the order $\min(a_1,a_2); \max(a_1,a_2)$

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

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

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

An example run



MergeSort(a, n)

If $n = 1$ **return** the order a_1

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

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

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

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

Runtime analysis on the board…