Lecture 21

CSE 331

Oct 23, 2023

Time pressure on mid-term(s)

Few clarifications (mid-term + project)

Few clarifications/comments:

- I generally mention this once the mid-terms are graded but since this has been mentioned to me by more than one of you:
  - Yes, I know the mid-terms had time pressure (as I mentioned in the mid-term posts). Specifically, unless you referred away everything you could, you might have been short of time.
  - However, there will not besuch a time pressure on the final exam. The way I create final exams is that a student who is top of everything would require <= 1min/point for a total of <= 100 minutes. And everyone gets 150 minutes in total (so 50% more than the ideally prepared student).
    - I have *never* heard of complaints about time in my final exams for the last 10+ years.
- Couple of clarifications on the project:
  - I would highly recommend that y'all focus more of your energies on trying to figure out good heuristics for the coding project and then submit to see how it performs.
    - Note I mentioned heuristics since we do not know of optimal solutions for these problems
    - While reading through the code that we provided could be useful do not spend too much time on it. This is not a programming course: at least in my *personal* opinion, to complete the coding problem successfully you do not really need to understand much of the provided code (except the functionality of some of the functions-- for which you need to understand the mathematical definitions in the problem).
  - If you submit code for problem 2, do submit it to problems 3-5.
    - It is possible that even if your code does not work very well for problem 2, it *might* work pretty well for one of the later problems.
    - Of course the above is not guaranteed but you have nothing to lose by doing the other submissions.

project   mid-term

Edit   good note | 0   Updated 7 minutes ago by Atri Rudra

Some other stuff coming up

Actions ▾

What's next?

Now that the mid-terms are done, hope y'all take some time to decompress! Some of you might have questions on how you're doing in the course, how you did in the mid-term exams and perhaps some of you think you'd like to come and chat with me.

I just wanted to give y'all some heads up on this:

- Our goal is to be able to finishing grading (both the) mid-terms by middle of this week (so by Wed or so).
  - Your TAs also have mid-terms so we appreciate your patience as they grade your mid-terms!
  - Once that is done, as with the HWs, I'll release the stats as well as the grading rubric. The usual re-grade policy will apply.
- Grading of HW 4 will not start until your TAs have graded the mid-terms.
  - Our aim will be to provide feedback on HW 4 *before* HW 5 is due (HW 5 goes out this Tue) but apologies in case we are not able to do this! (Again, please see comment above on your TAs having mid-terms.)
- Once the mid-terms are graded I'll assign temporary letter grades to y'all (based on your scores of HWs1-3, Quiz 1 and mid-terms) just so that y'all get a sense of where you stand in the course currently.
  - I'll put up a piazza post with the details once the temp. letter grades have been assigned.
  - Note that this will *not* be the same as the mid-semester grade that I submitted to HUB @367.
- Those who have a D+ or below in their temporary letter grade, I'll send email asking you to setup a one-on-one meeting (<=10 mins).
  - Even if you have a better grade than D+ but want to chat about your performance, you can also sign up (but those with D+ or below will get preference for a slot)
  - I'll put up a piazza post with details once I finalize the meeting slots.

mid-term   grading

Edit   good note   0

Updated 3 minutes ago by Atri Rudra

Project deadlines coming up

Tue, Oct 31		**(HW 5 in)**
Wed, Nov 1	Multiplying large integers ▶^{F22} ▶^{F21} ▶^{F19} ▶^{F18} ▶^{F17} x^2	[KT, Sec 5.5] *Reading Assignment:* Unraveling the mystery behind the identity
Fri, Nov 3	Closest Pair of Points ▶^{F22} ▶^{F21} ▶^{F19} ▶^{F18} ▶^{F17} x^2	[KT, Sec 5.4] (**Project (Problems 1 & 2 `Coding`) in**)
Mon, Nov 6	Kickass Property Lemma ▶^{F22} ▶^{F21} ▶^{F19} ▶^{F18} ▶^{F17} x^2	[KT, Sec 5.4] (**Project (Problems 1 & 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

Questions/Comments?
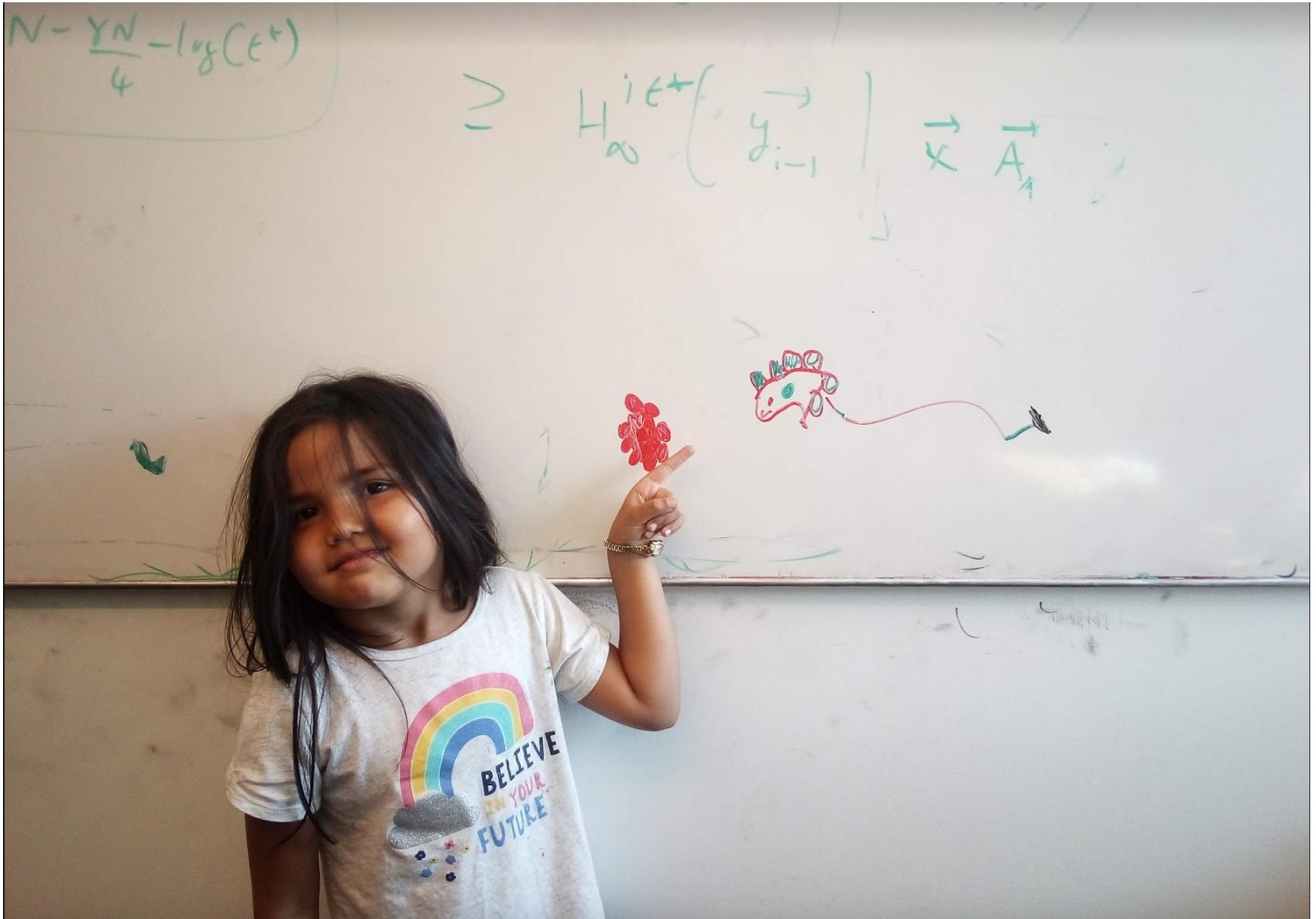
Make broadband more available

Cattaraugus County

Population: 79518
Median Income: $41,368.88
Access to any cable technology: 67.5%
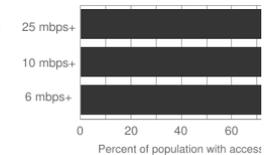Access to two or more wireline providers: 61.2%

BOTH
technical and
societal issues
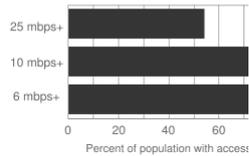
Say you are
tasked to come
up with the
infrastructure

Erie County

Population: 913295
Median Income: $49,817.67
Access to any cable technology: 98.9%
Access to two or more wireline providers: 96.8%

Building a fiber network

Lay down fibers to connect n locations

All n locations should be connected

Laying down a fiber costs money

What is the cheapest way to lay down the fibers?

Today's agenda

Minimum Spanning Tree (MST) Problem

Greedy algorithm(s) for MST problem

On to the board…

Minimum Spanning Tree Problem

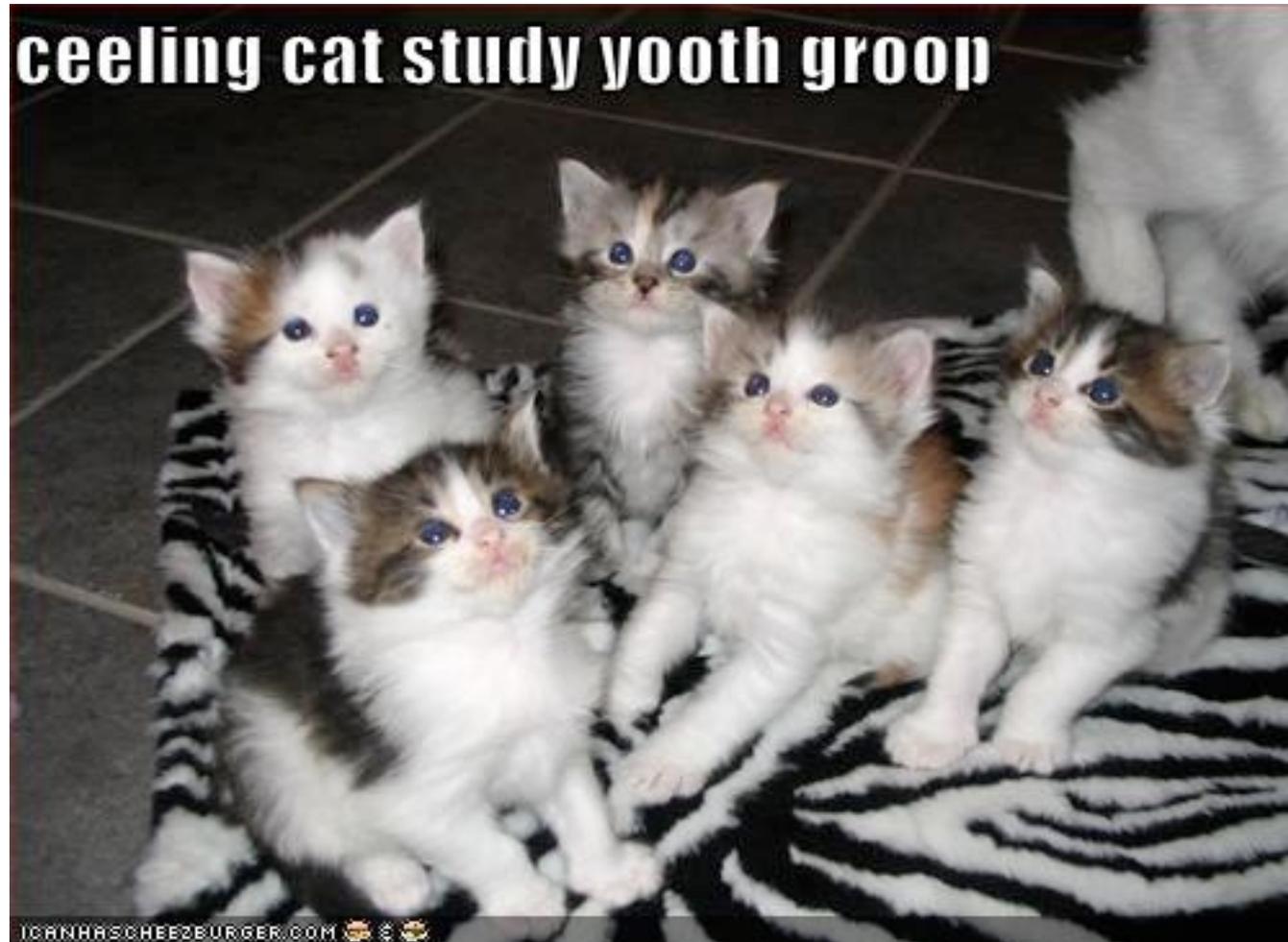**Input**: Undirected, connected $G = (V,E)$, edge costs c_e

Output: Subset $E' \subseteq E$, s.t. $T = (V,E')$ is connected
$C(T)$ is minimized

If all $c_e > 0$, then T is indeed a tree

Rest of today's agenda

Greedy algorithm(s) for MST problem

Discuss: Greedy algorithm!

Kruskal's Algorithm

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

T = ∅

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

Joseph B. Kruskal
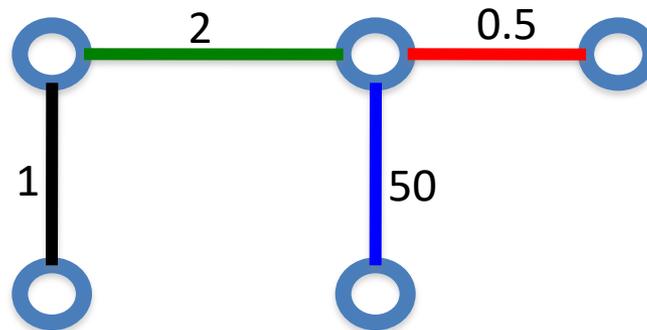
Prim's algorithm

Similar to Dijkstra's algorithm

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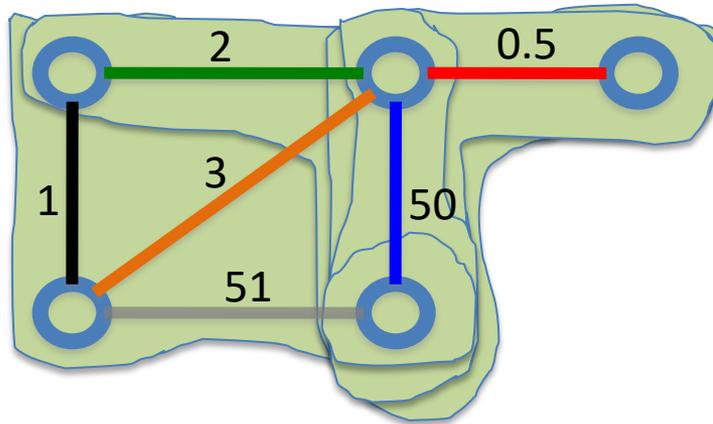
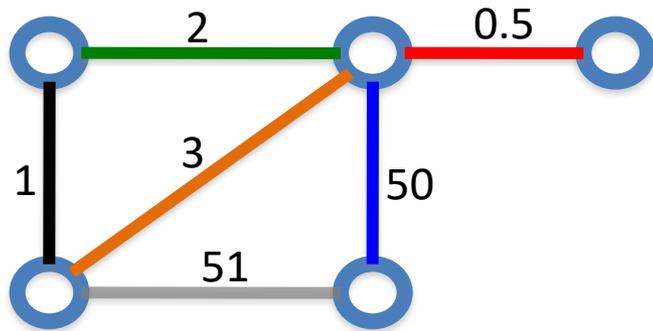Input: $G=(V,E)$, $c_e > 0$ for every e in E

$S = \{s\}$, $T = \emptyset$

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
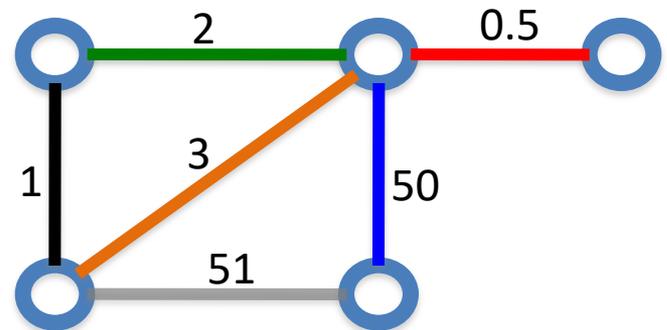
    Add w to S, e to T

Reverse-Delete Algorithm

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

T = E

Sort edges in decreasing order of their cost

Consider edges in sorted order

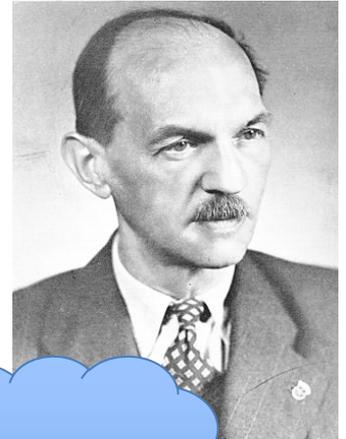If an edge can be removed T without disconnecting T then remove it
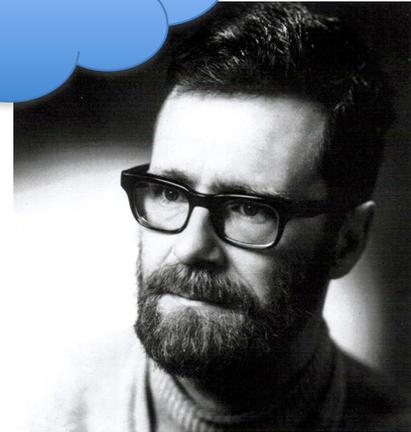
(Old) History of MST algorithms

1920: Otakar Borůvka

1930: Vojtěch Jarník

Same algo!

1956: Kruskal

1957: Prim

1959: Dijkstra