Lecture 20

CSE 331

Oct 19, 2022

Project deadlines coming up

Fri, Oct 28	Counting Inversions F21 F19 F18 F17 x²	[KT, Sec 5.3] (**Project (Problem 1** `Coding` **) in**)
Mon, Oct 31	Multiplying large integers F21 F19 F18 F17 x²	[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 x²	[KT, Sec 5.4]
Fri, Nov 4	Kickass Property Lemma F21 F19 F18 F17 x²	[KT, Sec 5.4] (**Project (Problem 2** `Coding` **) in**)
Mon, Nov 7	Weighted Interval Scheduling F21 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.

Mid-terms graded

note @319

stop following 3 views

Actions

Mid term 2 graded

Mid-term 2 has now been graded and the scores and feedback released on Autolab.

(Please see the re-grade policy as well as the grading rubric below before contacting us with questions on grading.)

Here are the stats:

Mid-term 2

Problem	Mean	Median	StdDev	Max	Min
2(a) Algo Idea	10.5	14.5	5.7	15.0	0.0
2(a) Runtime	3.1	4.0	1.9	5.0	0.0

Mid-term temp grades

Will be assigned hopefully by tonight

Some other stuff coming up

note @282 ☆ 🔒 ▾

stop following | 0 views

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:

- (As a tangent, note that HW 4 is already out: @274)
- Our goal is to be able to finishing grading (both the) mid-terms by early to mid next week.
  - 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.
- 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 need to submit to HUB by tomorrow (mainly because the mid terms will not be graded by this Friday, which is when the mid-semester grades are due).
- 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 11 seconds ago by Atri Rudra

HW 5 is out

Homework 5

Due by **11:30pm, Tuesday, October 25, 2022**.

Make sure you follow all the homework policies.

All submissions should be done via Autolab.

Check the week 8 recitation notes for this homework.

Question 1 (Computing Set Intersection on a Network) [50 points]

The Problem

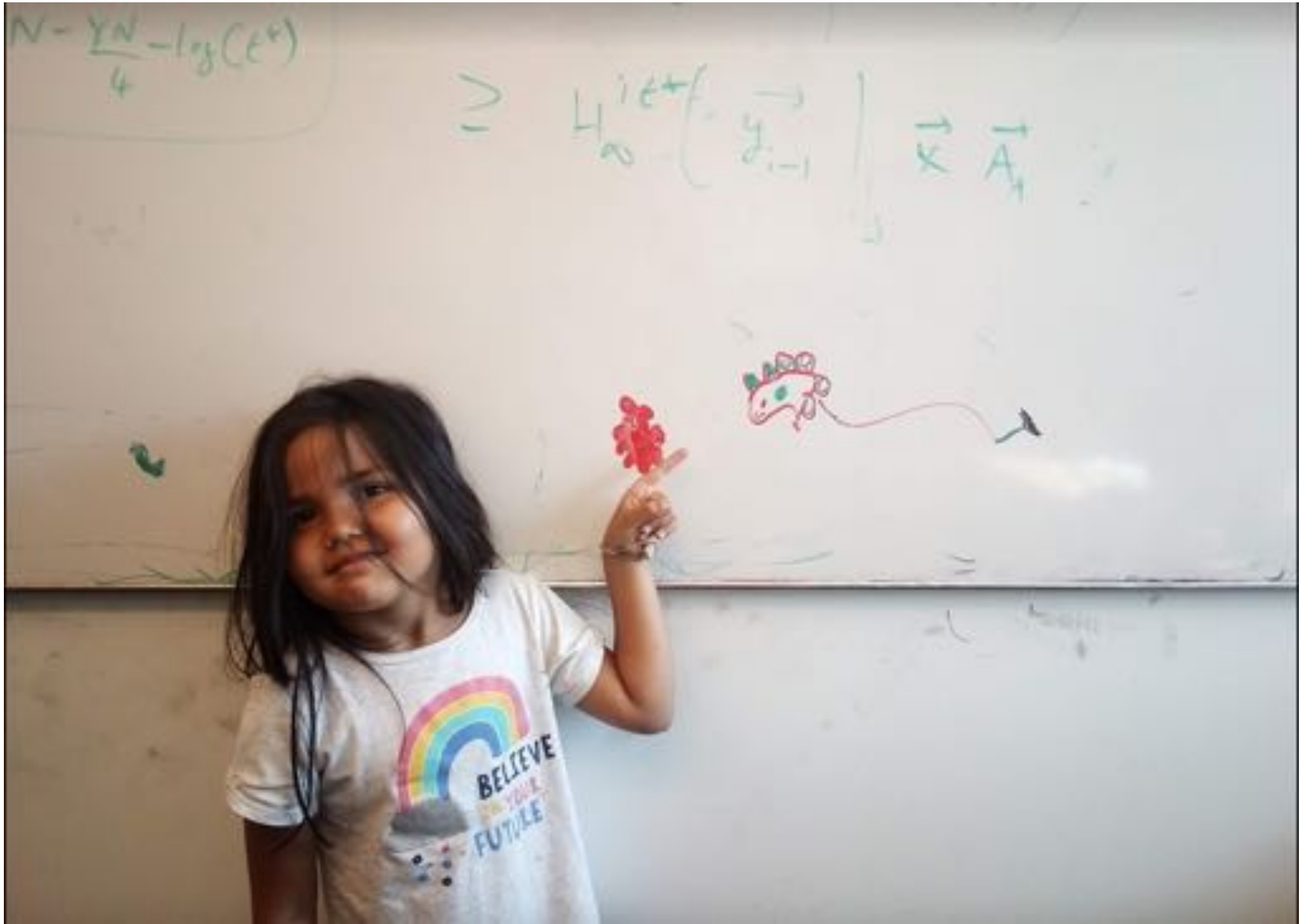
In this problem, we will take a break from trying to minimize the runtime of the algorithm and focus on an important resource in distributed computing: the total number of bits communicated over a network by the algorithm.

Given a graph $G = (V, E)$, which is the underlying *network topology*, we want to compute the intersection of $n = |V|$ sets over the network G. More precisely, every node $u \in V$, gets a set $S_u \subseteq [M]$ for some integer $M \geq 1$. (Note that M has *nothing* to do with the number of edges in G.) Further we are given a special node $t \in V$. The goal of this problem is to design an algorithm such that when the algorithm terminates, the node t knows the intersection of all sets:

$$\cap_{u \in V} S_u.$$

Moreover, we want to design such an algorithm that minimizes the total communication over G.

Questions/Comments?



Make broadband more available

BOTH technical and societal issues

Say you are tasked to come up with the infrastructure

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

Joseph B. Kruskal

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

$T = \emptyset$

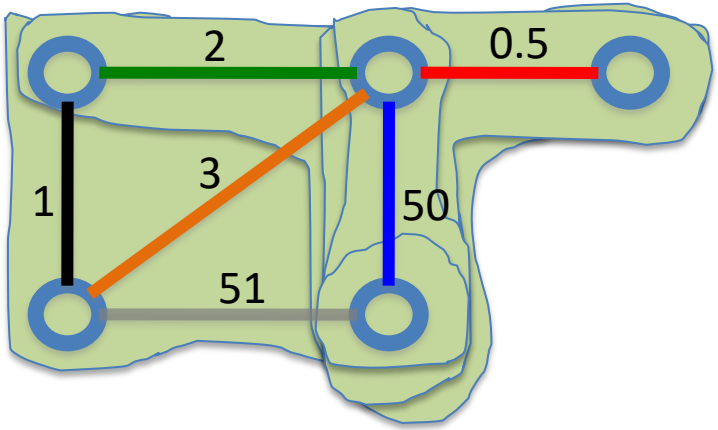
Sort edges in increasing order of their cost

Consider edges in sorted order

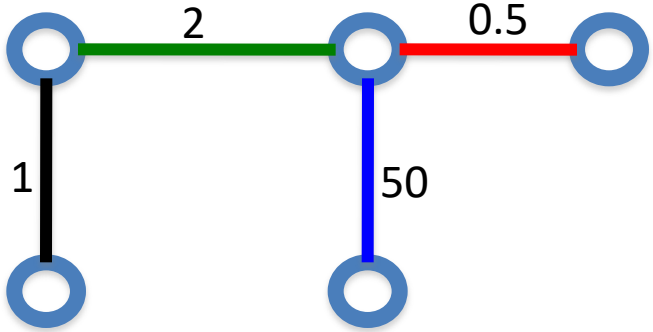
If an edge can be added to T without adding a cycle then add it to T

Prim's algorithm

Similar to Dijkstra's algorithm



Robert Prim



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

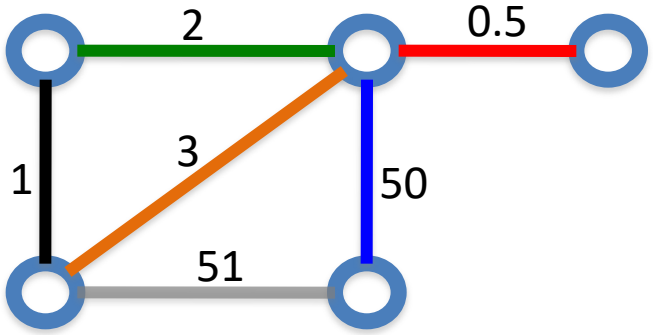
S = {s}, T = Ø

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

Reverse-Delete Algorithm



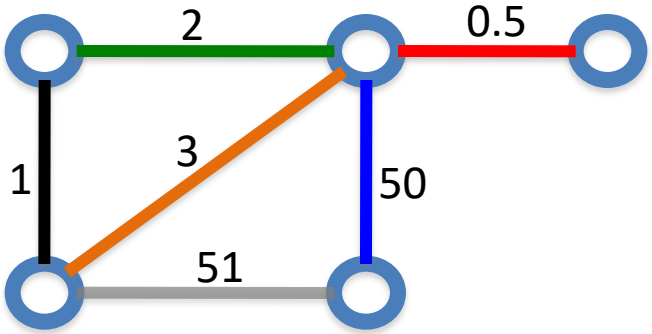
Input: G=(V,E), $c_e > 0$ for every e in E

T = E

Sort edges in decreasing order of their cost

Consider edges in sorted order

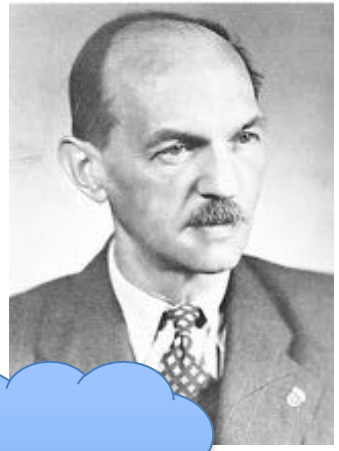
If an edge can be removed T without disconnecting T then remove it



(Old) History of MST algorithms

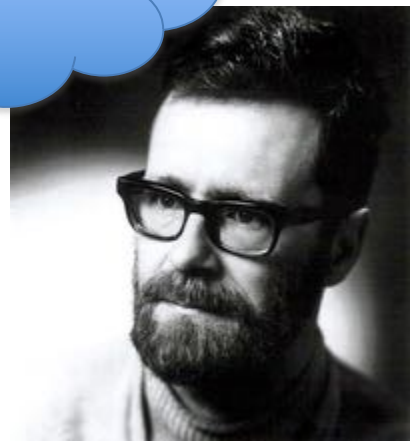
1920: Otakar Borůvka

1930: Vojtěch Jarník



Same algo!

1956: Kruskal



1957: Prim

1959: Dijkstra