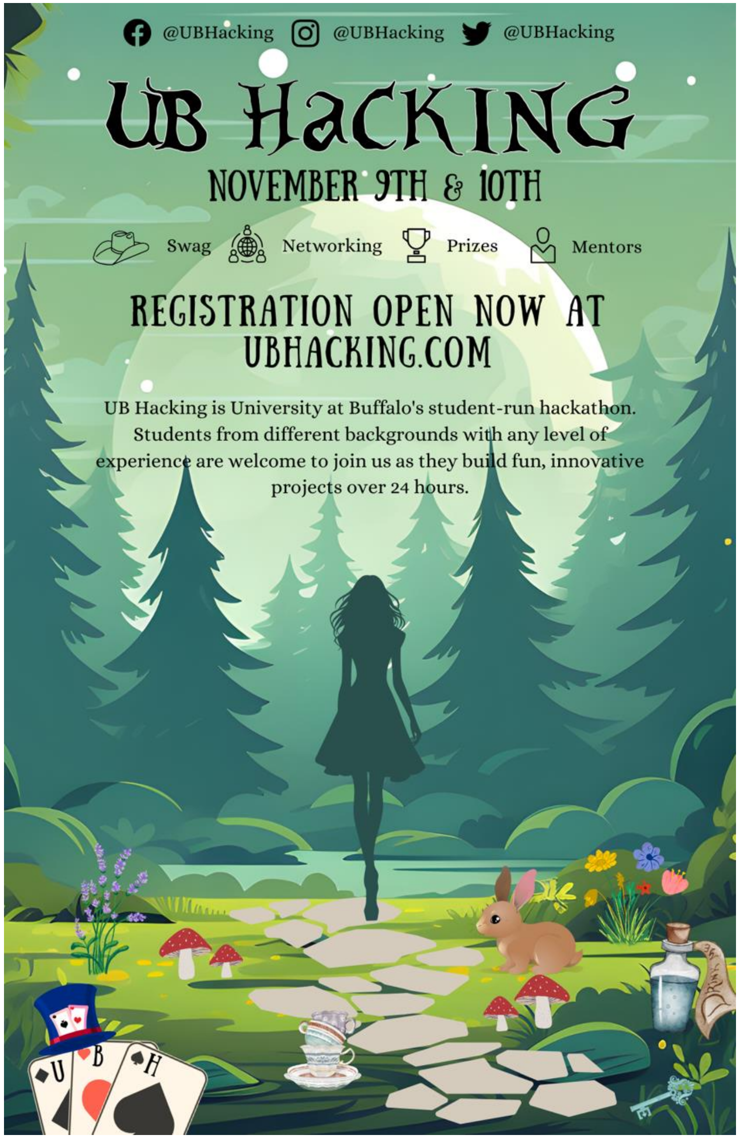
Lecture 22

CSE 331

Oct 23, 2024

UB Hacking!



@UBHacking @UBHacking @UBHacking

UB HACKING

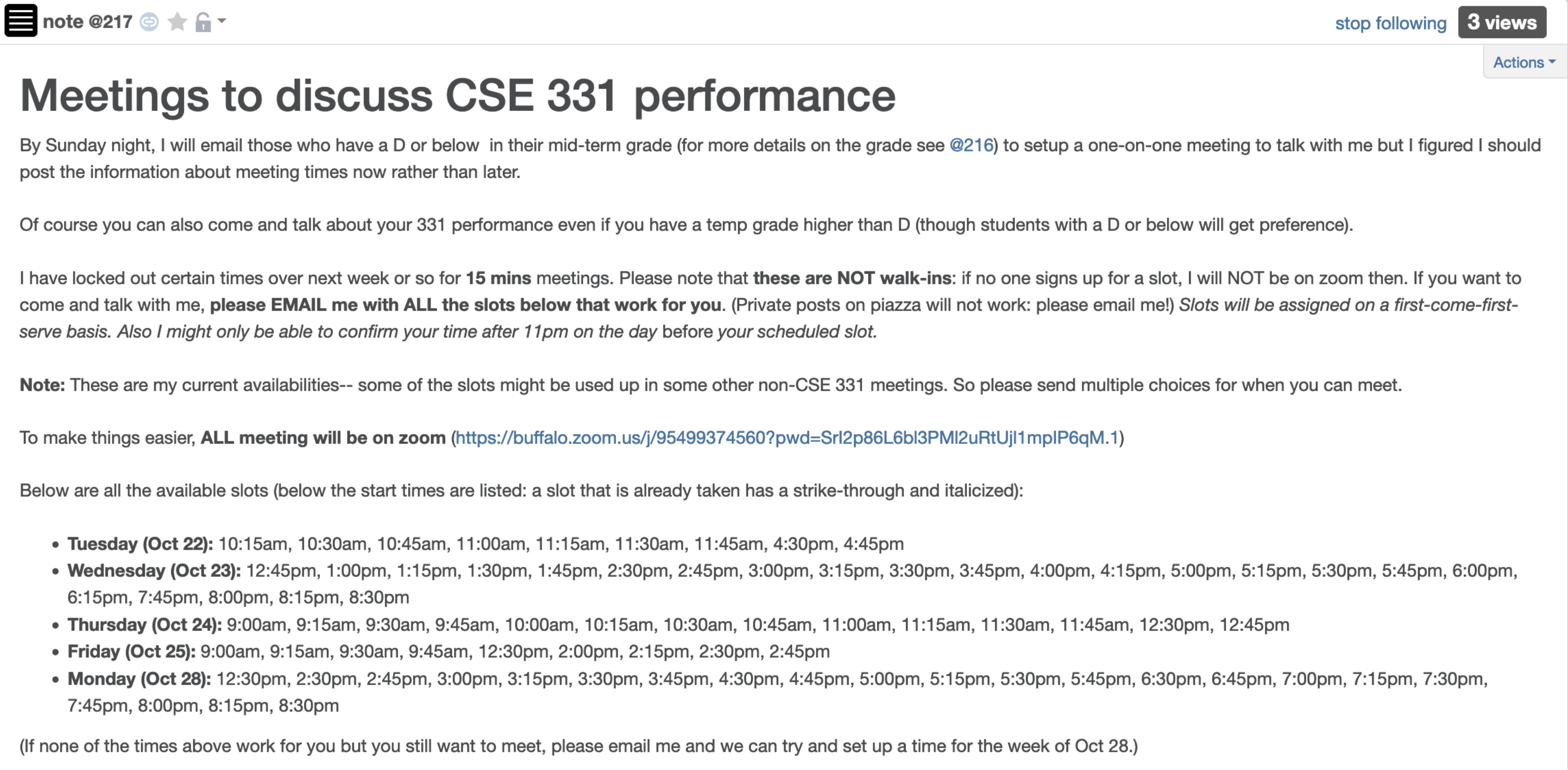
NOVEMBER 9TH & 10TH

Swag Networking Prizes Mentors

REGISTRATION OPEN NOW AT UBHACKING.COM

UB Hacking is University at Buffalo's student-run hackathon. Students from different backgrounds with any level of experience are welcome to join us as they build fun, innovative projects over 24 hours.

Details on 1-on-1 meetings



Meetings to discuss CSE 331 performance

By Sunday night, I will email those who have a D or below in their mid-term grade (for more details on the grade see @216) 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.

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.

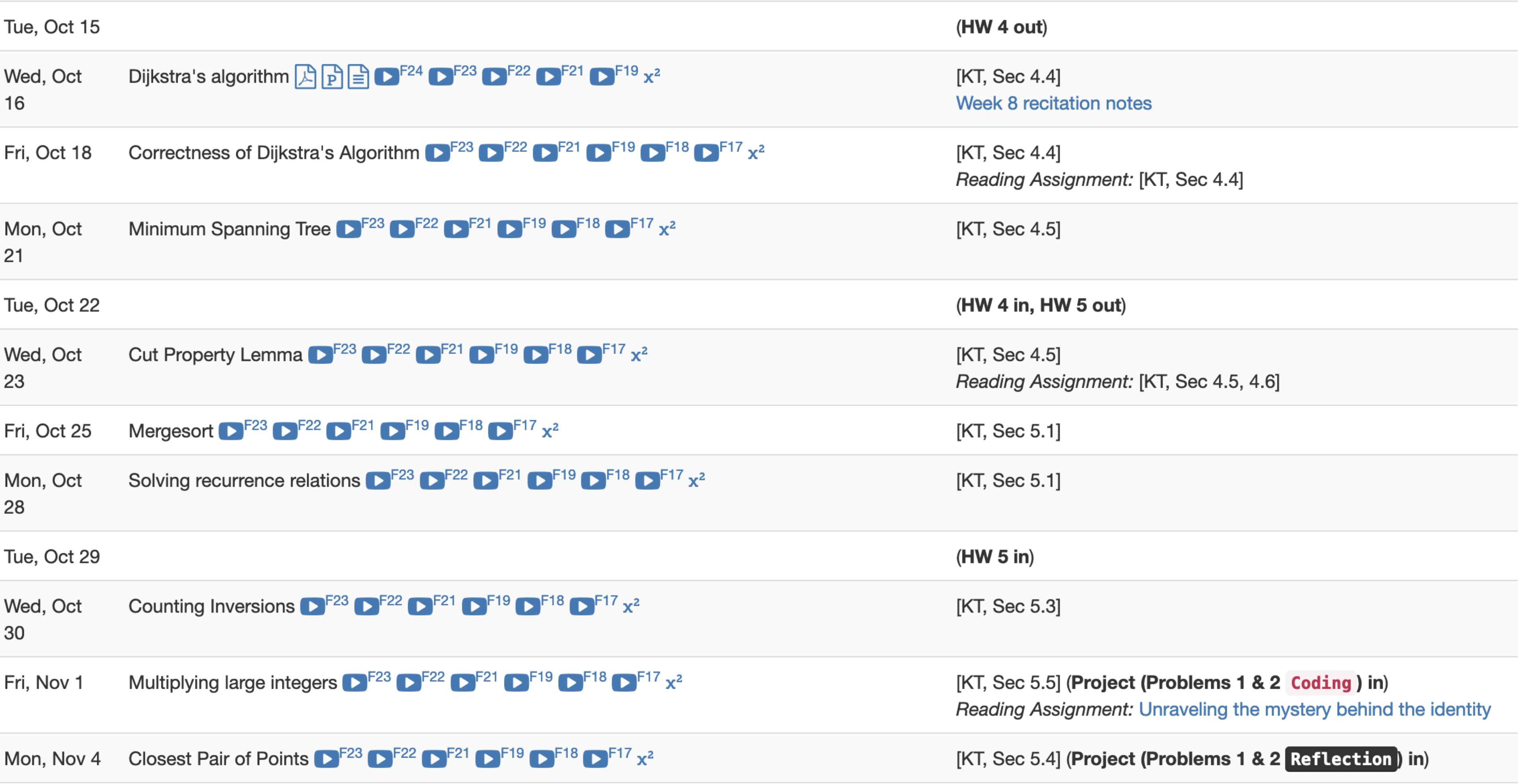
To make things easier, **ALL meeting will be on zoom** (https://buffalo.zoom.us/j/95499374560?pwd=Srl2p86L6bl3PMl2uRtUjl1mplP6qM.1)

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

- **Tuesday (Oct 22):** 10:15am, 10:30am, 10:45am, 11:00am, 11:15am, 11:30am, 11:45am, 4:30pm, 4:45pm
- **Wednesday (Oct 23):** 12:45pm, 1:00pm, 1:15pm, 1:30pm, 1:45pm, 2:30pm, 2:45pm, 3:00pm, 3:15pm, 3:30pm, 3:45pm, 4:00pm, 4:15pm, 5:00pm, 5:15pm, 5:30pm, 5:45pm, 6:00pm, 6:15pm, 7:45pm, 8:00pm, 8:15pm, 8:30pm
- **Thursday (Oct 24):** 9:00am, 9:15am, 9:30am, 9:45am, 10:00am, 10:15am, 10:30am, 10:45am, 11:00am, 11:15am, 11:30am, 11:45am, 12:30pm, 12:45pm
- **Friday (Oct 25):** 9:00am, 9:15am, 9:30am, 9:45am, 12:30pm, 2:00pm, 2:15pm, 2:30pm, 2:45pm
- **Monday (Oct 28):** 12:30pm, 2:30pm, 2:45pm, 3:00pm, 3:15pm, 3:30pm, 3:45pm, 4:30pm, 4:45pm, 5:00pm, 5:15pm, 5:30pm, 5:45pm, 6:30pm, 6:45pm, 7:00pm, 7:15pm, 7:30pm, 7:45pm, 8:00pm, 8:15pm, 8:30pm

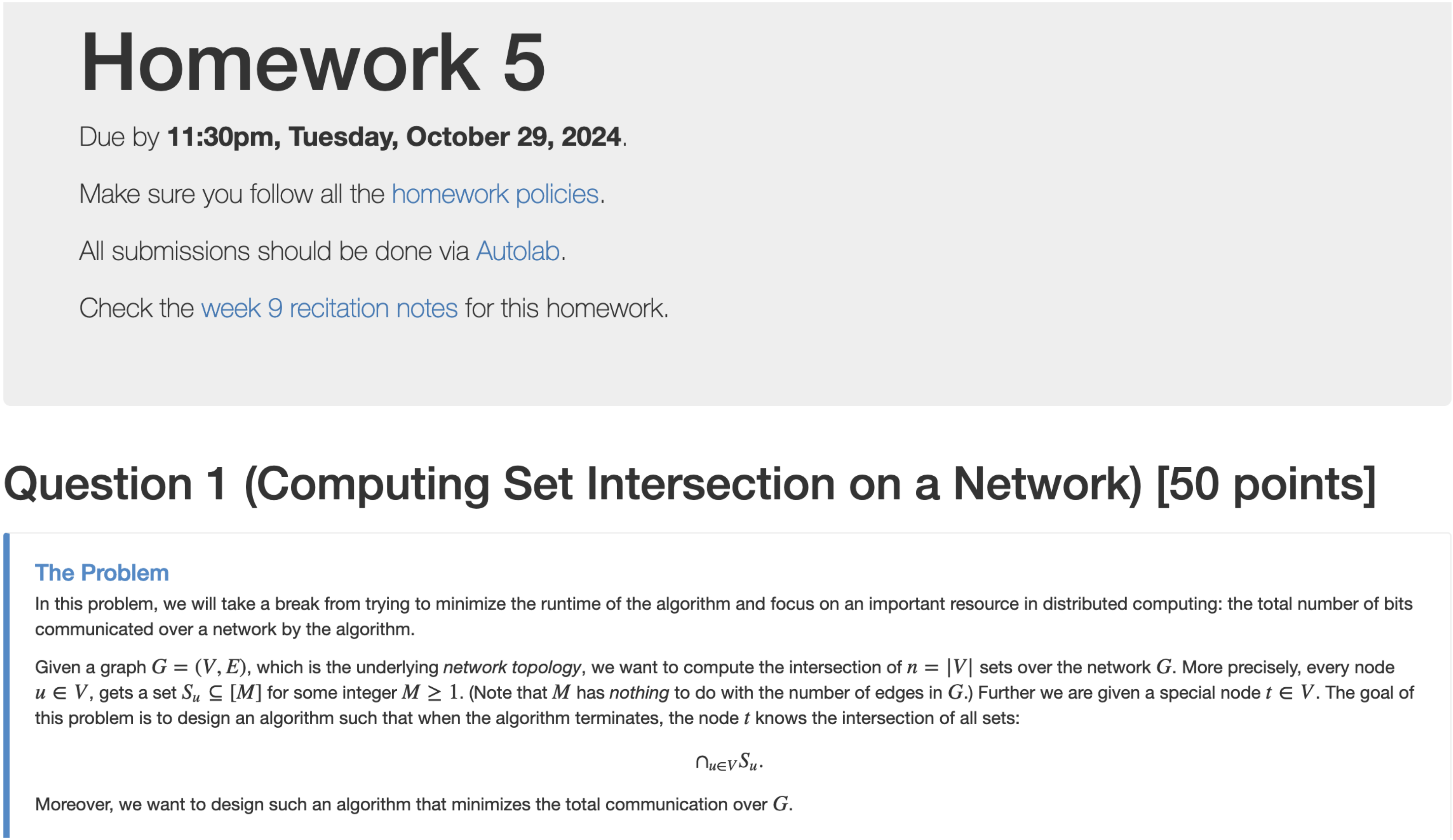
(If none of the times above work for you but you still want to meet, please email me and we can try and set up a time for the week of Oct 28.)

Project deadlines coming up



Tue, Oct 15		**(HW 4 out)**
Wed, Oct 16	Dijkstra's algorithm 📄 📄 📄 ▶F24 ▶F23 ▶F22 ▶F21 ▶F19 x²	[KT, Sec 4.4] Week 8 recitation notes
Fri, Oct 18	Correctness of Dijkstra's Algorithm ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 4.4] *Reading Assignment:* [KT, Sec 4.4]
Mon, Oct 21	Minimum Spanning Tree ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 4.5]
Tue, Oct 22		**(HW 4 in, HW 5 out)**
Wed, Oct 23	Cut Property Lemma ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 4.5] *Reading Assignment:* [KT, Sec 4.5, 4.6]
Fri, Oct 25	Mergesort ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 5.1]
Mon, Oct 28	Solving recurrence relations ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 5.1]
Tue, Oct 29		**(HW 5 in)**
Wed, Oct 30	Counting Inversions ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 5.3]
Fri, Nov 1	Multiplying large integers ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 5.5] **(Project (Problems 1 & 2 `Coding`) in)** *Reading Assignment:* Unraveling the mystery behind the identity
Mon, Nov 4	Closest Pair of Points ▶F23 ▶F22 ▶F21 ▶F19 ▶F18 ▶F17 x²	[KT, Sec 5.4] **(Project (Problems 1 & 2 `Reflection`) in)**

HW 5 out



Homework 5

Due by **11:30pm, Tuesday, October 29, 2024**.

Make sure you follow all the homework policies.

All submissions should be done via Autolab.

Check the week 9 recitation notes for this homework.

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

The Problem

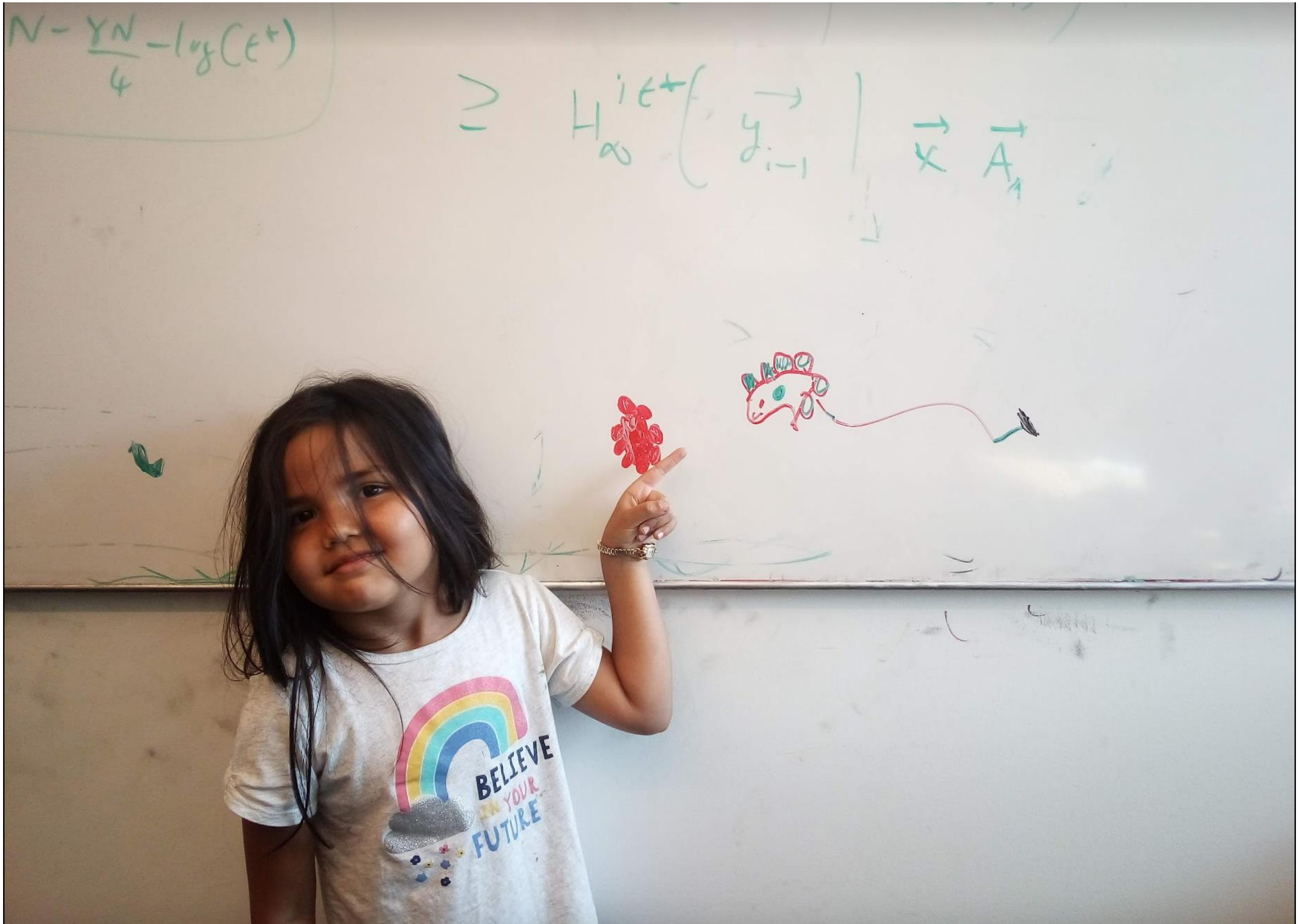
In this problem, we will take a break from trying to minimize the runtime of the algorithm and focus on an important resource in distributed computing: the total number of bits communicated over a network by the algorithm.

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?



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

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

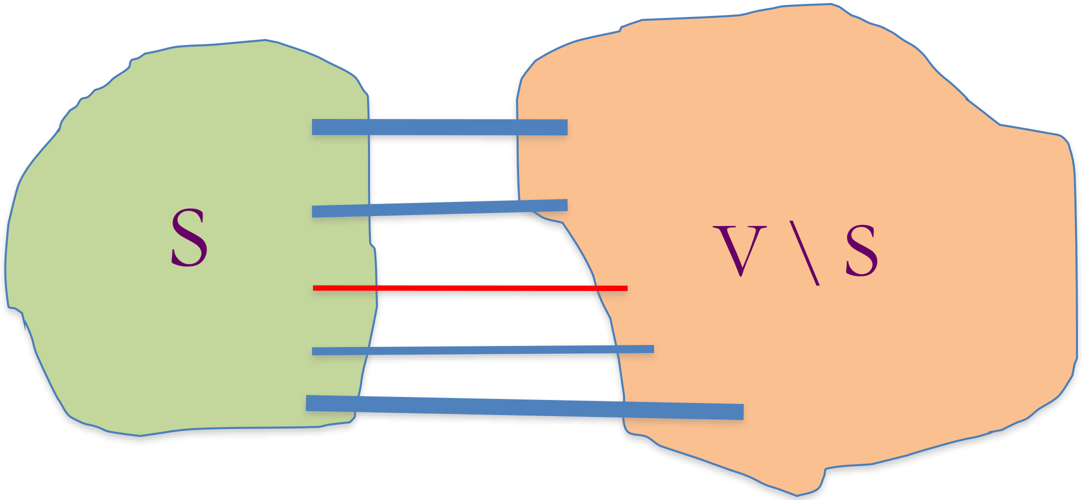
Robert Prim

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

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

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 $T$
```

Cut Property Lemma for MSTs

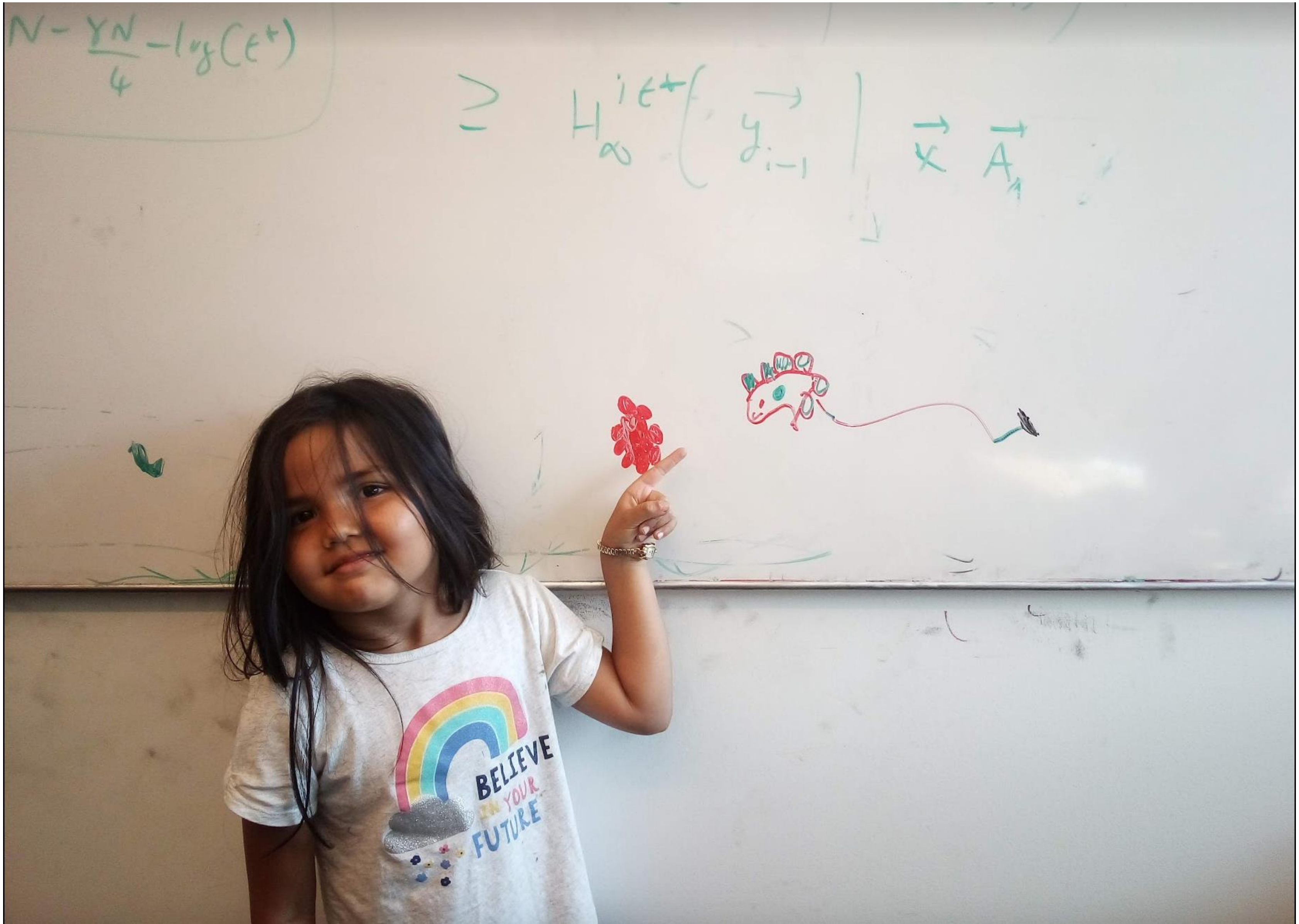
Condition: S and V\S are non-empty



Cheapest crossing edge is in **all** MSTs

Assumption: All edge costs are distinct

Questions/Comments?


$N - \frac{YN}{4} - \log(\epsilon^+)$
$\geq H_\infty^{i\epsilon^+}(\vec{y}_{i-1} \mid \vec{x}\ \vec{A}_1$
BELIEVE IN YOUR FUTURE

Today's agenda

Optimality of Prim's algorithm

Prove Cut Property Lemma

Optimality of Kruskal's algorithm

Remove distinct edge weights assumption

On to the board…

$\frac{r}{k} \leq D^{kt}(f) \leq D^{1w}(f) \leq r \log(\frac{n}{r})$
x_1
x_2
x_3
x_4
x_5