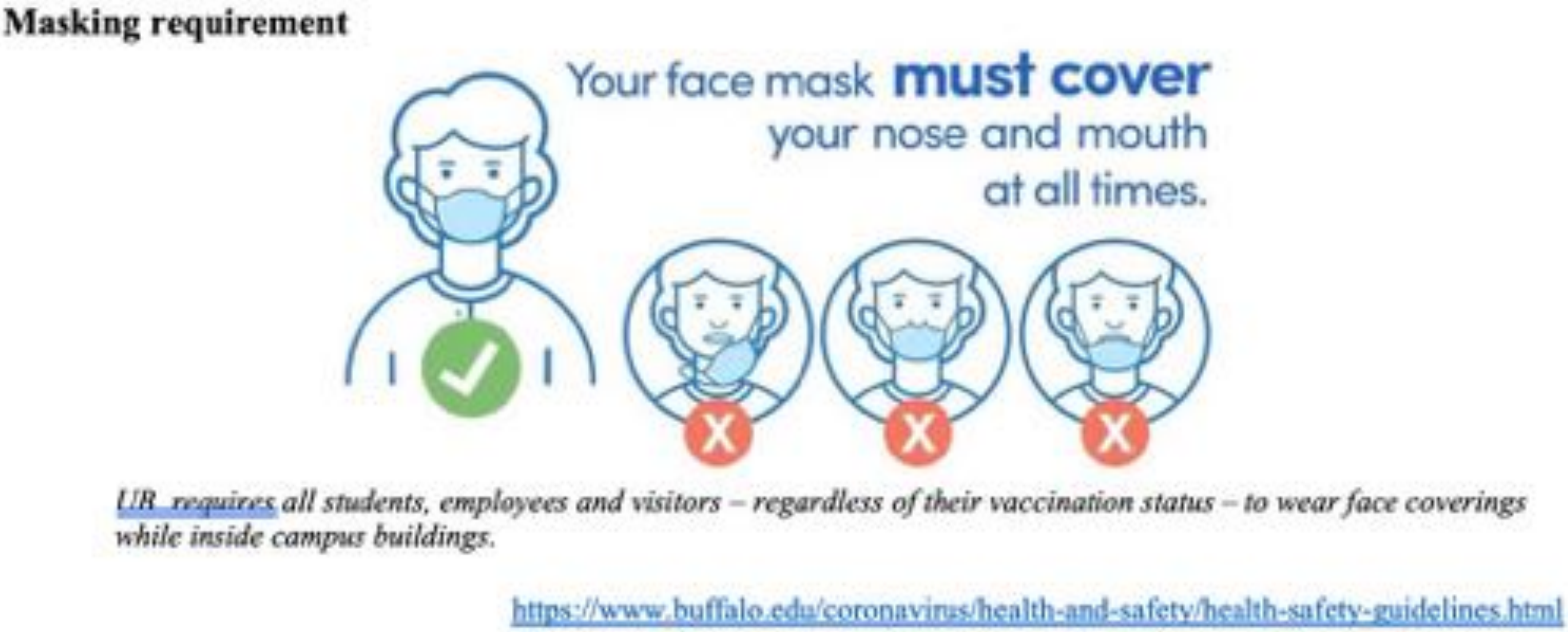
Lecture 14

CSE 331

Oct 1, 2021

Please have a face mask on



Masking requirement

Your face mask **must cover** your nose and mouth at all times.

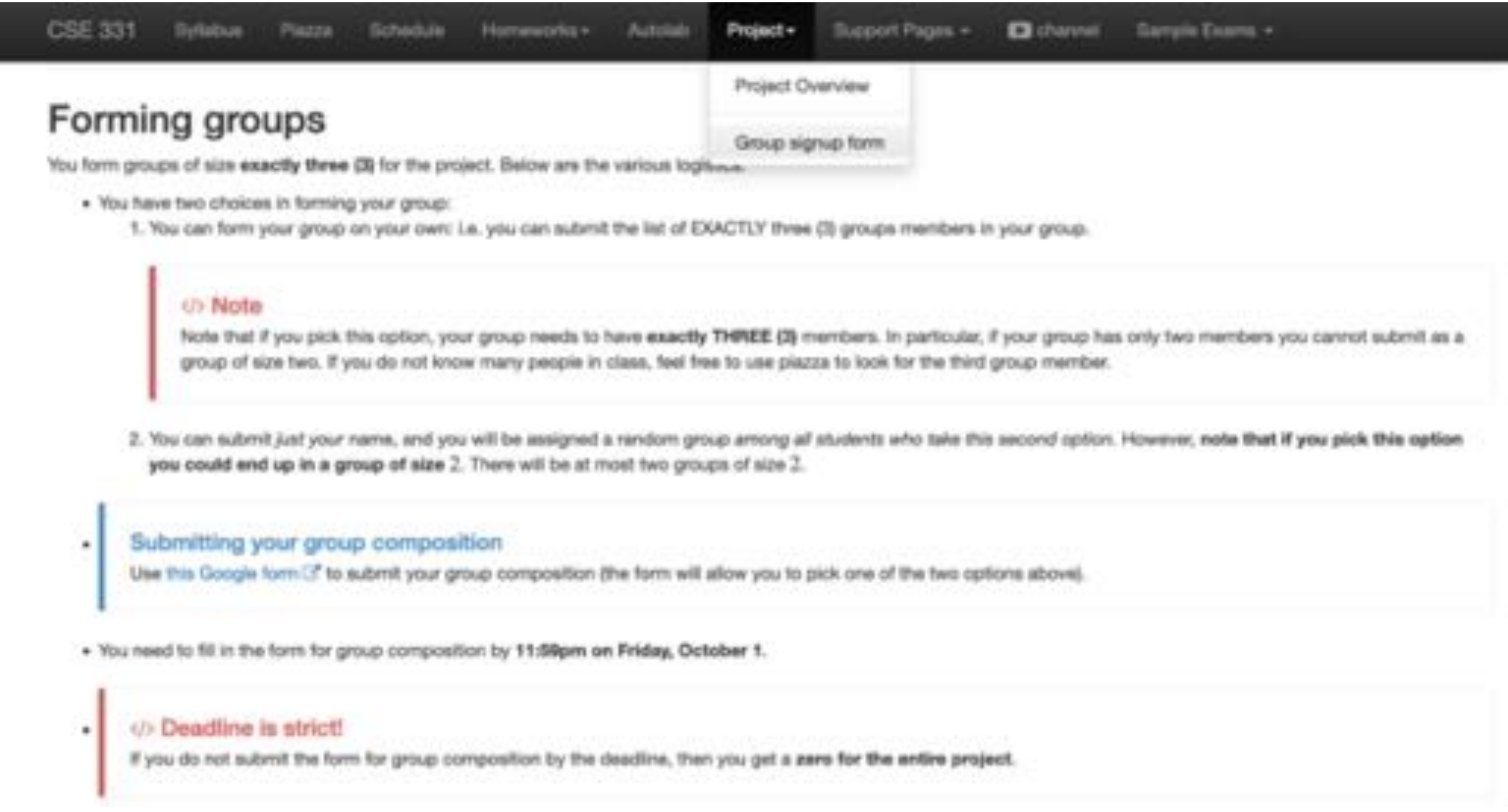
UR requires all students, employees and visitors – regardless of their vaccination status – to wear face coverings while inside campus buildings.

https://www.buffalo.edu/coronavirus/health-and-safety/health-safety-guidelines.html

If you need it, ask for help

Project groups due TONIGHT!

Deadline: Friday, Oct 1, 11:59pm



Forming groups

You form groups of size **exactly three (3)** for the project. Below are the various logistics.

- You have two choices in forming your group:
  1. You can form your group on your own: i.e. you can submit the list of EXACTLY three (3) groups members in your group.

     > **</> Note**
     >
     > Note that if you pick this option, your group needs to have **exactly THREE (3)** members. In particular, if your group has only two members you cannot submit as a group of size two. If you do not know many people in class, feel free to use piazza to look for the third group member.

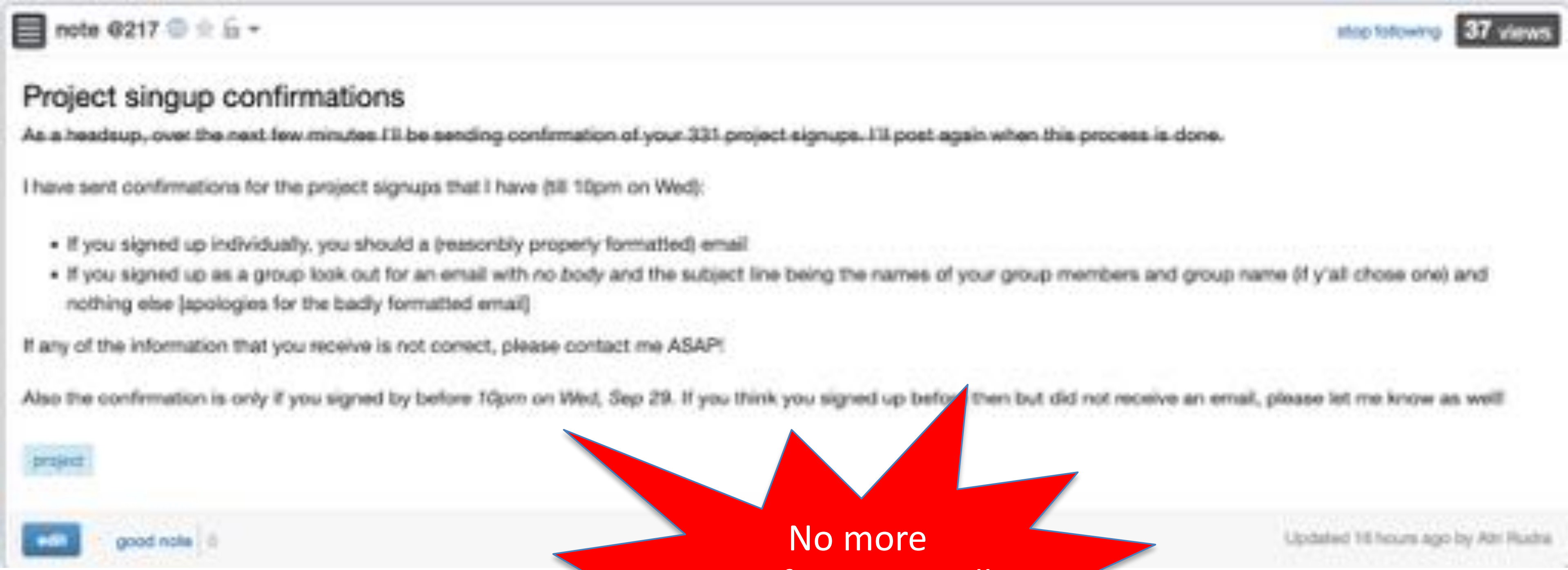
  2. You can submit *just your name*, and you will be assigned a random group *among all students who take this second option*. However, **note that if you pick this option you could end up in a group of size** 2. There will be at most two groups of size 2.

- > **Submitting your group composition**
  >
  > Use this Google form to submit your group composition (the form will allow you to pick one of the two options above).

- You need to fill in the form for group composition by **11:59pm on Friday, October 1.**

- > **</> Deadline is strict!**
  >
  > If you do not submit the form for group composition by the deadline, then you get a **zero for the entire project.**

About ~10 have not signed up



Project singup confirmations

~~As a headsup, over the next few minutes I'll be sending confirmation of your 331 project signups. I'll post again when this process is done.~~

I have sent confirmations for the project signups that I have (till 10pm on Wed):

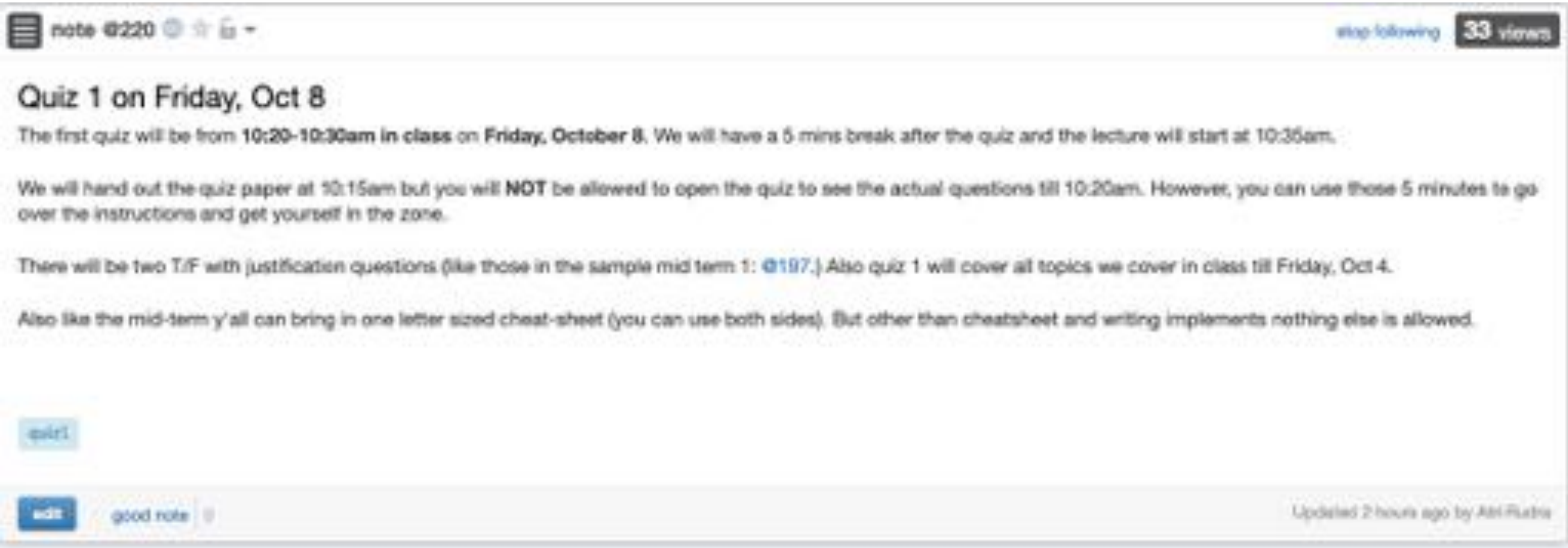
- If you signed up individually, you should a (reasonbly properly formatted) email
- If you signed up as a group look out for an email with *no body* and the subject line being the names of your group members and group name (if y'all chose one) and nothing else [apologies for the badly formatted email]

If any of the information that you receive is not correct, please contact me ASAP!

Also the confirmation is only if you signed by before 10pm on Wed, Sep 29. If you think you signed up befo then but did not receive an email, please let me know as well!

No more confirmations till AFTER the deadline

Quiz 1 in a week



Quiz 1 on Friday, Oct 8

The first quiz will be from **10:20-10:30am in class** on **Friday, October 8**. We will have a 5 mins break after the quiz and the lecture will start at 10:35am.

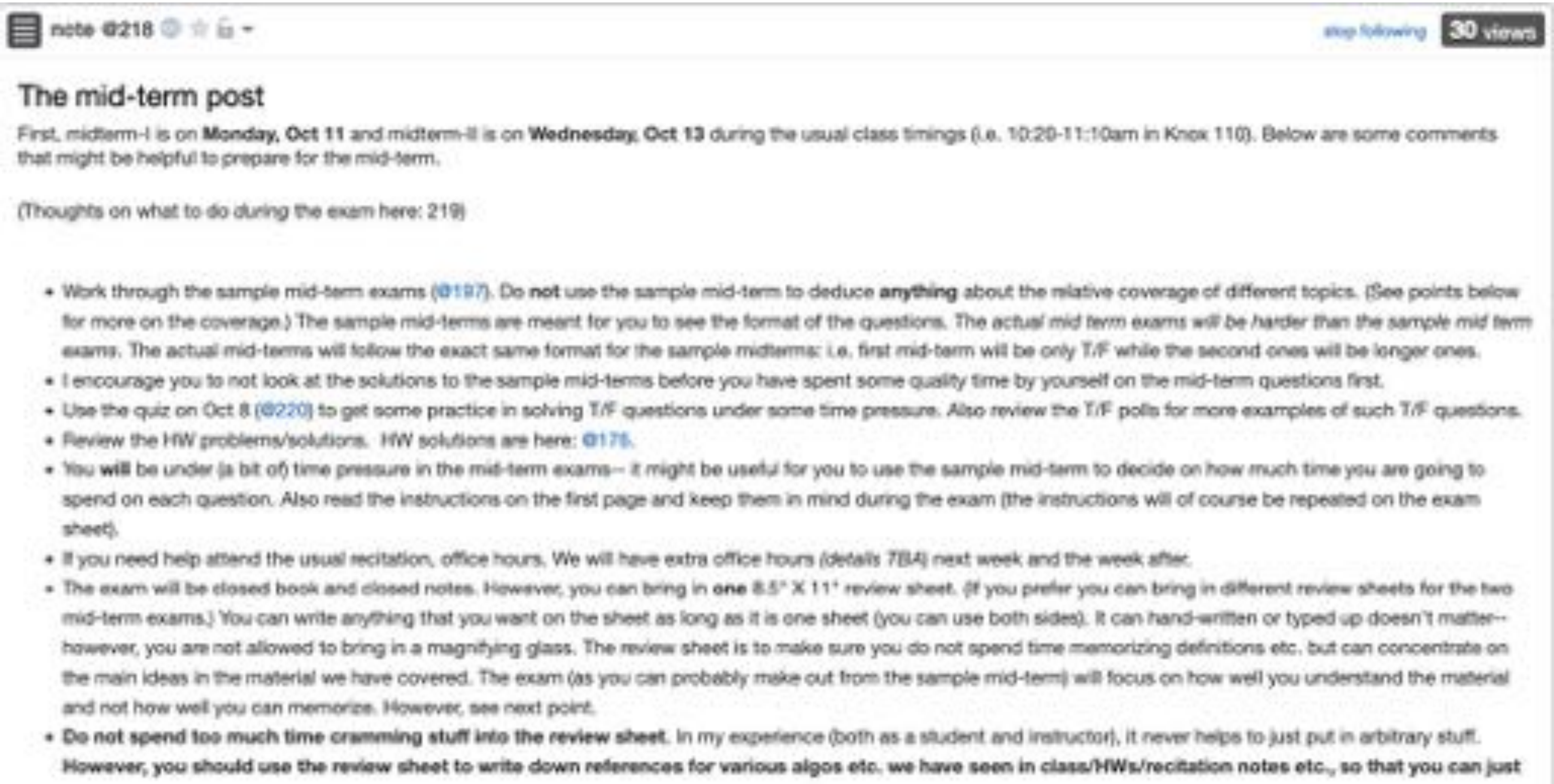
We will hand out the quiz paper at 10:15am but you will **NOT** be allowed to open the quiz to see the actual questions till 10:20am. However, you can use those 5 minutes to go over the instructions and get yourself in the zone.

There will be two T/F with justification questions (like those in the sample mid term 1: @197.) Also quiz 1 will cover all topics we cover in class till Friday, Oct 4.

Also like the mid-term y'all can bring in one letter sized cheat-sheet (you can use both sides). But other than cheatsheet and writing implements nothing else is allowed.

Updated 2 hours ago by Atri Rudra

Mid-term post



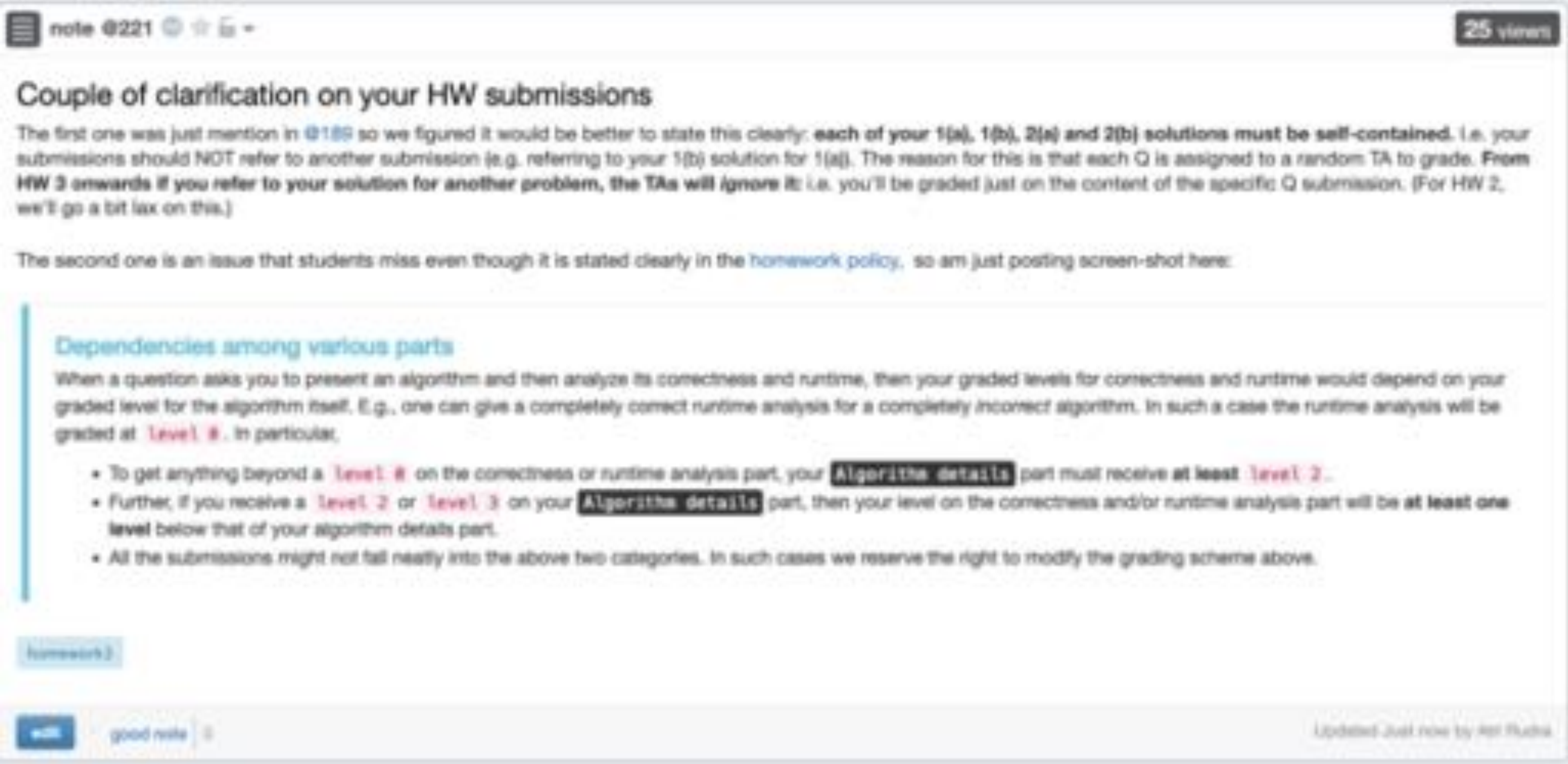
The mid-term post

First, midterm-I is on **Monday, Oct 11** and midterm-II is on **Wednesday, Oct 13** during the usual class timings (i.e. 10:20-11:10am in Knox 110). Below are some comments that might be helpful to prepare for the mid-term.

(Thoughts on what to do during the exam here: 219)

- Work through the sample mid-term exams (@197). Do **not** use the sample mid-term to deduce **anything** about the relative coverage of different topics. (See points below for more on the coverage.) The sample mid-terms are meant for you to see the format of the questions. *The actual mid term exams will be harder than the sample mid term exams.* The actual mid-terms will follow the exact same format for the sample midterms: i.e. first mid-term will be only T/F while the second ones will be longer ones.
- I encourage you to not look at the solutions to the sample mid-terms before you have spent some quality time by yourself on the mid-term questions first.
- Use the quiz on Oct 8 (@220) to get some practice in solving T/F questions under some time pressure. Also review the T/F polls for more examples of such T/F questions.
- Review the HW problems/solutions. HW solutions are here: @175.
- You **will** be under (a bit of) time pressure in the mid-term exams-- it might be useful for you to use the sample mid-term to decide on how much time you are going to spend on each question. Also read the instructions on the first page and keep them in mind during the exam (the instructions will of course be repeated on the exam sheet).
- If you need help attend the usual recitation, office hours. We will have extra office hours (*details TBA*) next week and the week after.
- The exam will be closed book and closed notes. However, you can bring in **one** 8.5" X 11" review sheet. (If you prefer you can bring in different review sheets for the two mid-term exams.) You can write anything that you want on the sheet as long as it is one sheet (you can use both sides). It can hand-written or typed up doesn't matter-- however, you are not allowed to bring in a magnifying glass. The review sheet is to make sure you do not spend time memorizing definitions etc. but can concentrate on the main ideas in the material we have covered. The exam (as you can probably make out from the sample mid-term) will focus on how well you understand the material and not how well you can memorize. However, see next point.
- **Do not spend too much time cramming stuff into the review sheet**. In my experience (both as a student and instructor), it never helps to just put in arbitrary stuff. **However, you should use the review sheet to write down references for various algos etc. we have seen in class/HWs/recitation notes etc., so that you can just**

Clarifications on your HW



Couple of clarification on your HW submissions

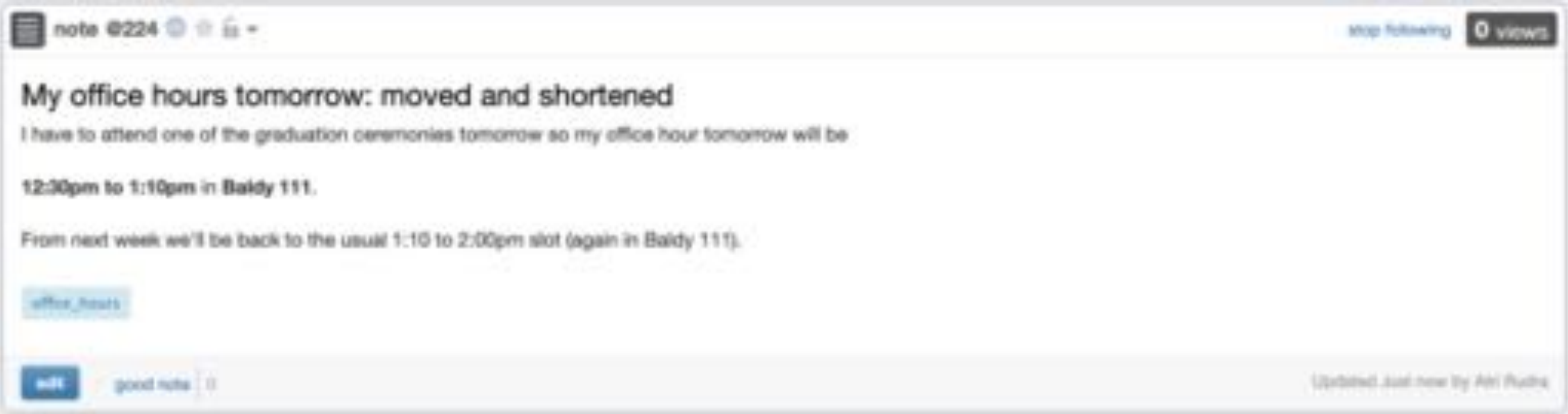
The first one was just mention in @189 so we figured it would be better to state this clearly: **each of your 1(a), 1(b), 2(a) and 2(b) solutions must be self-contained.** i.e. your submissions should NOT refer to another submission (e.g. referring to your 1(b) solution for 1(a)). The reason for this is that each Q is assigned to a random TA to grade. **From HW 3 onwards if you refer to your solution for another problem, the TAs will *ignore* it:** i.e. you'll be graded just on the content of the specific Q submission. (For HW 2, we'll go a bit lax on this.)

The second one is an issue that students miss even though it is stated clearly in the homework policy, so am just posting screen-shot here:

> ### Dependencies among various parts
>
> When a question asks you to present an algorithm and then analyze its correctness and runtime, then your graded levels for correctness and runtime would depend on your graded level for the algorithm itself. E.g., one can give a completely correct runtime analysis for a completely *incorrect* algorithm. In such a case the runtime analysis will be graded at `level 0`. In particular,
>
> - To get anything beyond a `level 0` on the correctness or runtime analysis part, your `Algorithm details` part must receive **at least** `level 2`.
> - Further, if you receive a `level 2` or `level 3` on your `Algorithm details` part, then your level on the correctness and/or runtime analysis part will be **at least one level** below that of your algorithm details part.
> - All the submissions might not fall neatly into the above two categories. In such cases we reserve the right to modify the grading scheme above.

Updated Just now by Atri Rudra

My office hour today



My office hours tomorrow: moved and shortened

I have to attend one of the graduation ceremonies tomorrow so my office hour tomorrow will be

12:30pm to 1:10pm in **Baldy 111**.

From next week we'll be back to the usual 1:10 to 2:00pm slot (again in Baldy 111).

Questions?

Breadth First Search (BFS)

Build layers of vertices connected to s

$L_0 = \{s\}$

Assume $L_0,..,L_j$ have been constructed

L_{j+1} set of vertices not chosen yet but are connected to L_j

Stop when new layer is empty

Use linked lists

Use CC[v] array

Rest of Today’s agenda

Quick run time analysis for BFS

Quick run time analysis for DFS (and Queue version of BFS)

Helping you schedule your activities for the day

$O(m+n)$ BFS Implementation

```
BFS($s$)

CC[$s$] = T and CC[$w$] = F for every $w \neq s$
Set $i = 0$
Set $L_0 = \{s\}$
While $L_i$ is not empty
    $L_{i+1} = \emptyset$
    For every $u$ in $L_i$
        For every edge $(u,w)$
        If CC[$w$] = F then
            CC[$w$] = T
            Add $w$ to $L_{i+1}$
    $i$++
```

Array

Input graph as Adjacency list

Linked List

Version in KT also computes a BFS tree

All the layers as one

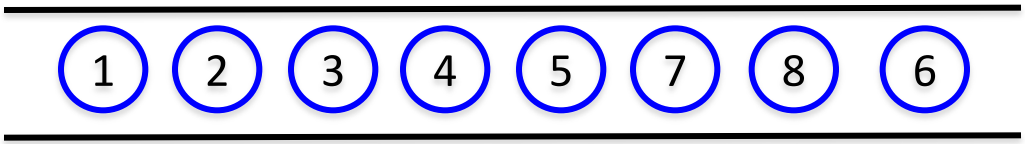
BFS(s)

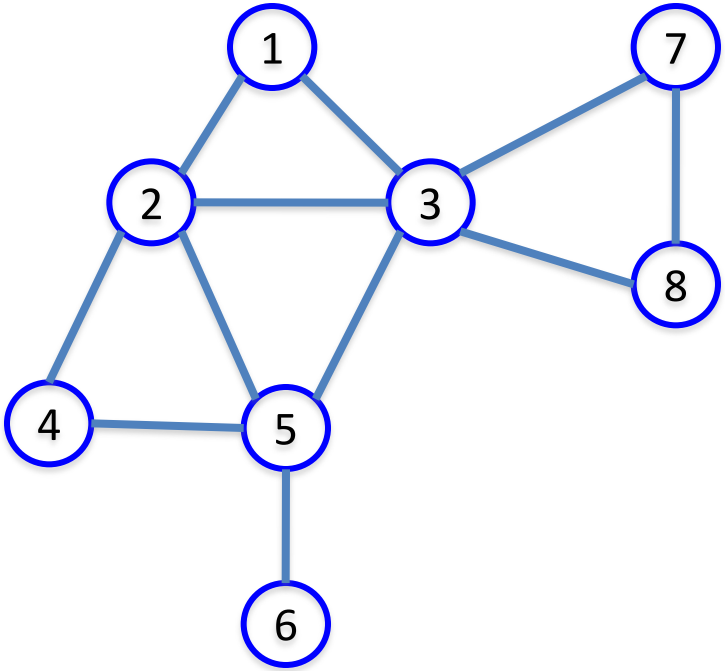
```
CC[s] = T and CC[w] = F for every $w \neq s$
Set i = 0
Set $L_0$ = {s}
While $L_i$ is not empty
        $L_{i+1} = \emptyset$
        For every u in $L_i$
                For every edge (u,w)
                If CC[w] = F then
                        CC[w] = T
                        Add w to $L_{i+1}$
        i++
```

All layers are considered in first-in-first-out order

Can combine all layers into one queue: all the children of a node are added to the end of the queue

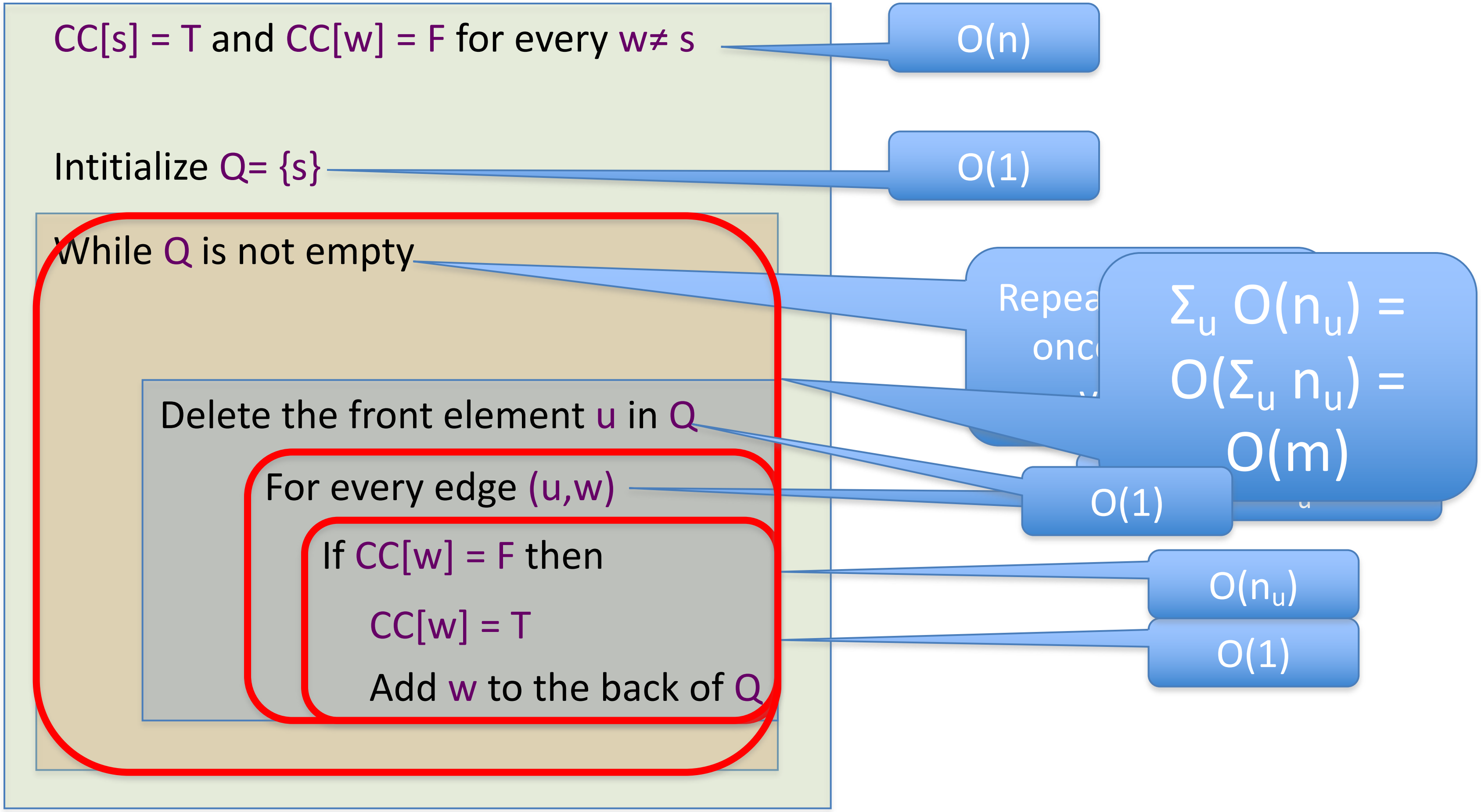
An illustration





Queue O(m+n) implementation

BFS(s)



```
CC[s] = T and CC[w] = F for every w≠ s

Intitialize Q= {s}

While Q is not empty

    Delete the front element u in Q

        For every edge (u,w)

            If CC[w] = F then

                CC[w] = T

                Add w to the back of Q
```

O(n)

O(1)

Repea

onc

Σ_u O(n_u) = O(Σ_u n_u) = O(m)

O(1)

O(n_u)

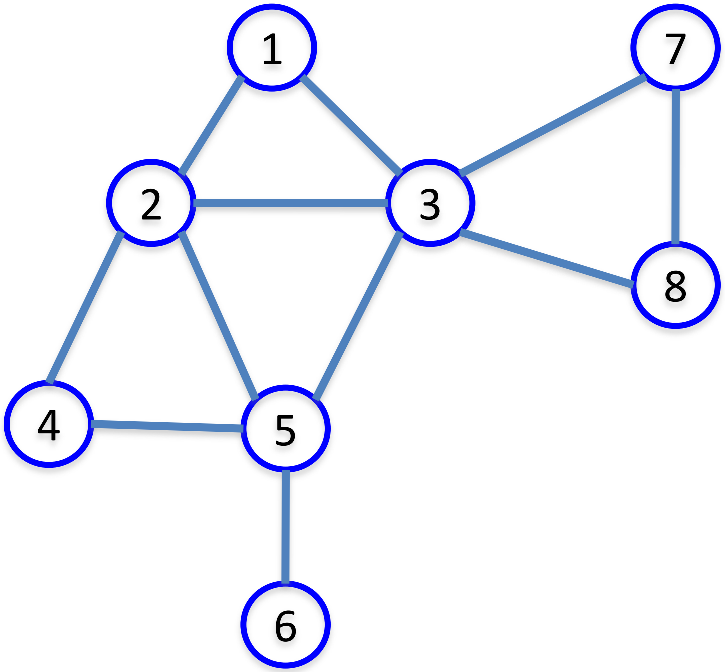
O(1)

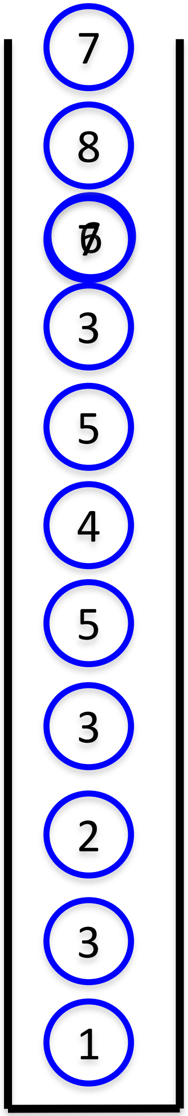
Questions/Comments?

Implementing DFS in $O(m+n)$ time

Same as BFS except stack instead of a queue

A DFS run using an explicit stack





DFS stack implementation

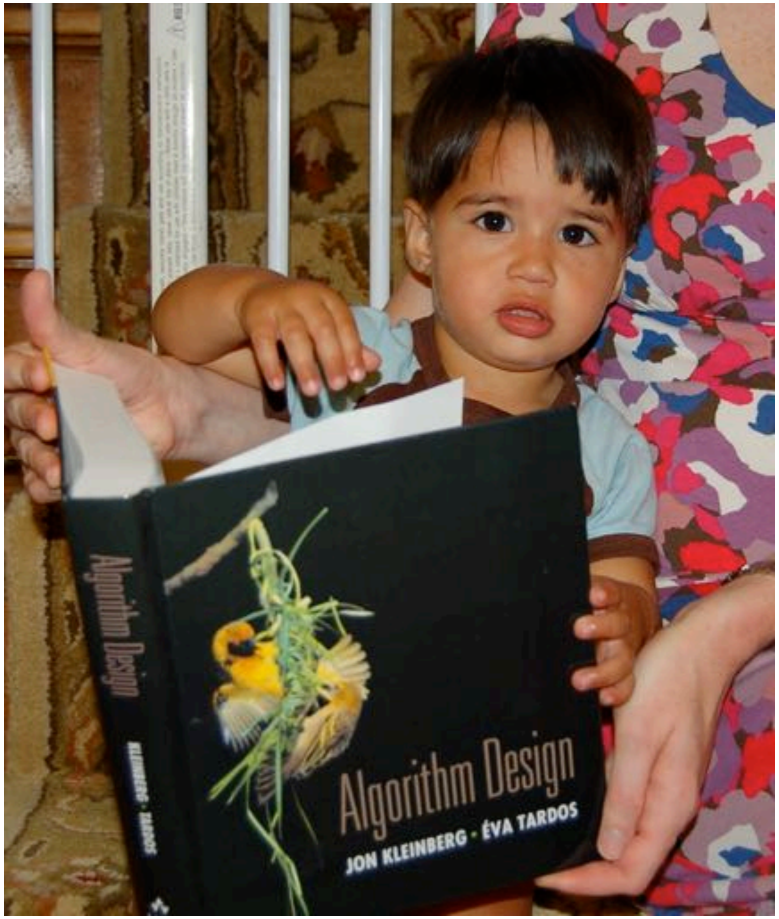
```
DFS(s)
    CC[s] = T and CC[w] = F for every w≠ s

    Intitialize Ŝ = {s}
    While Ŝ is not empty

        Pop the top element u in Ŝ
            For every edge (u,w)
                If CC[w] = F then
                    CC[w] = T
                    Push w to the top of Ŝ
```

Same O(m+n) run time analysis as for BFS

Questions/Comments?

Reading Assignment

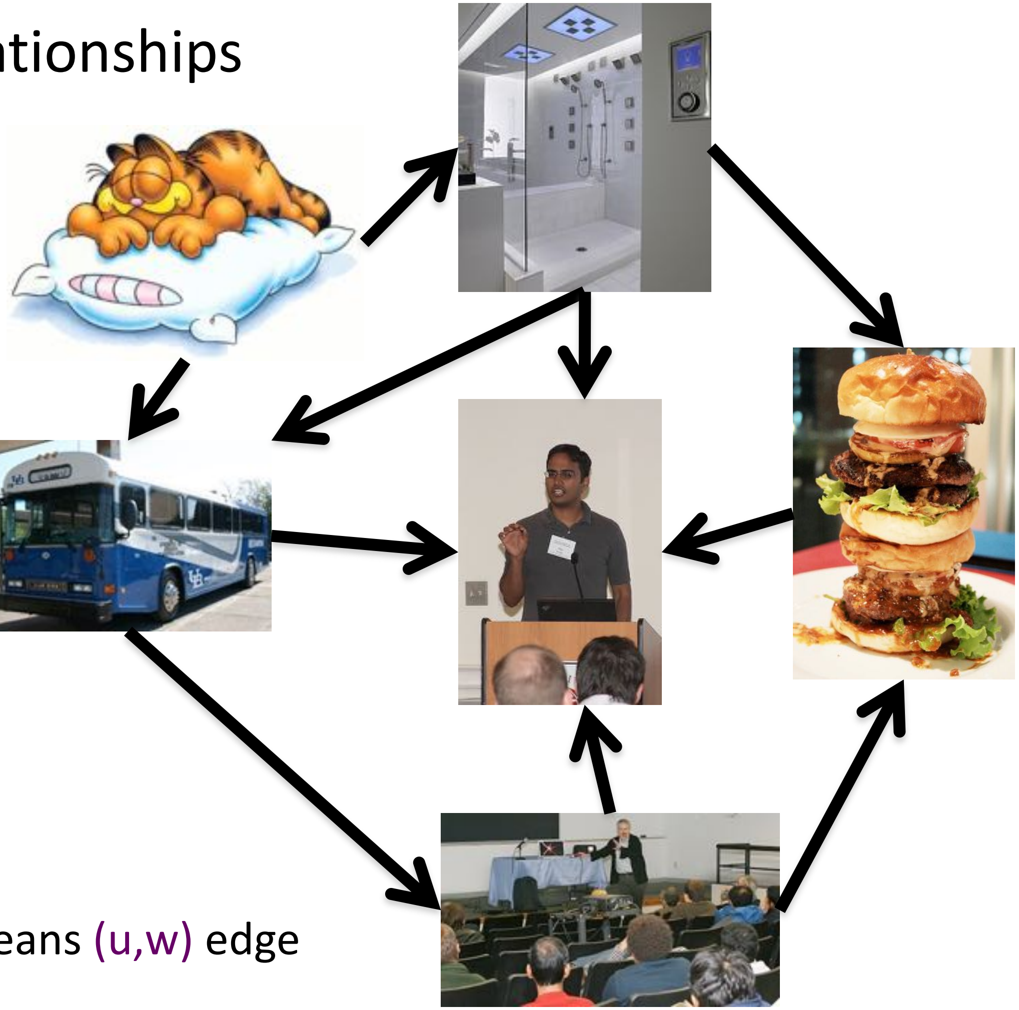
Sec 3.3, 3.4, 3.5 and 3.6 of [KT]



Directed graphs

Model asymmetric relationships

Precedence relationships

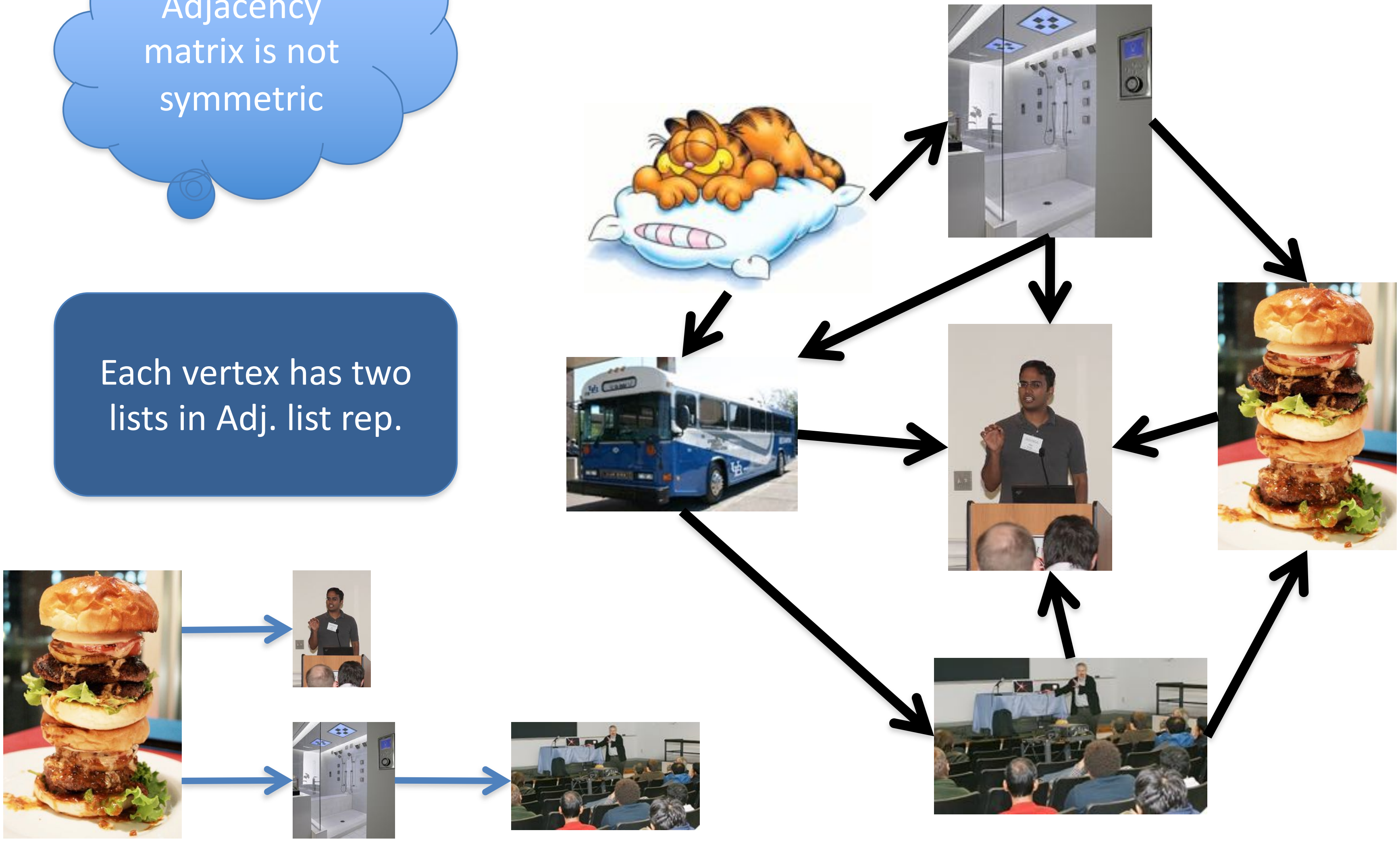


u needs to be done before w means (u,w) edge

Directed graphs

Adjacency matrix is not symmetric

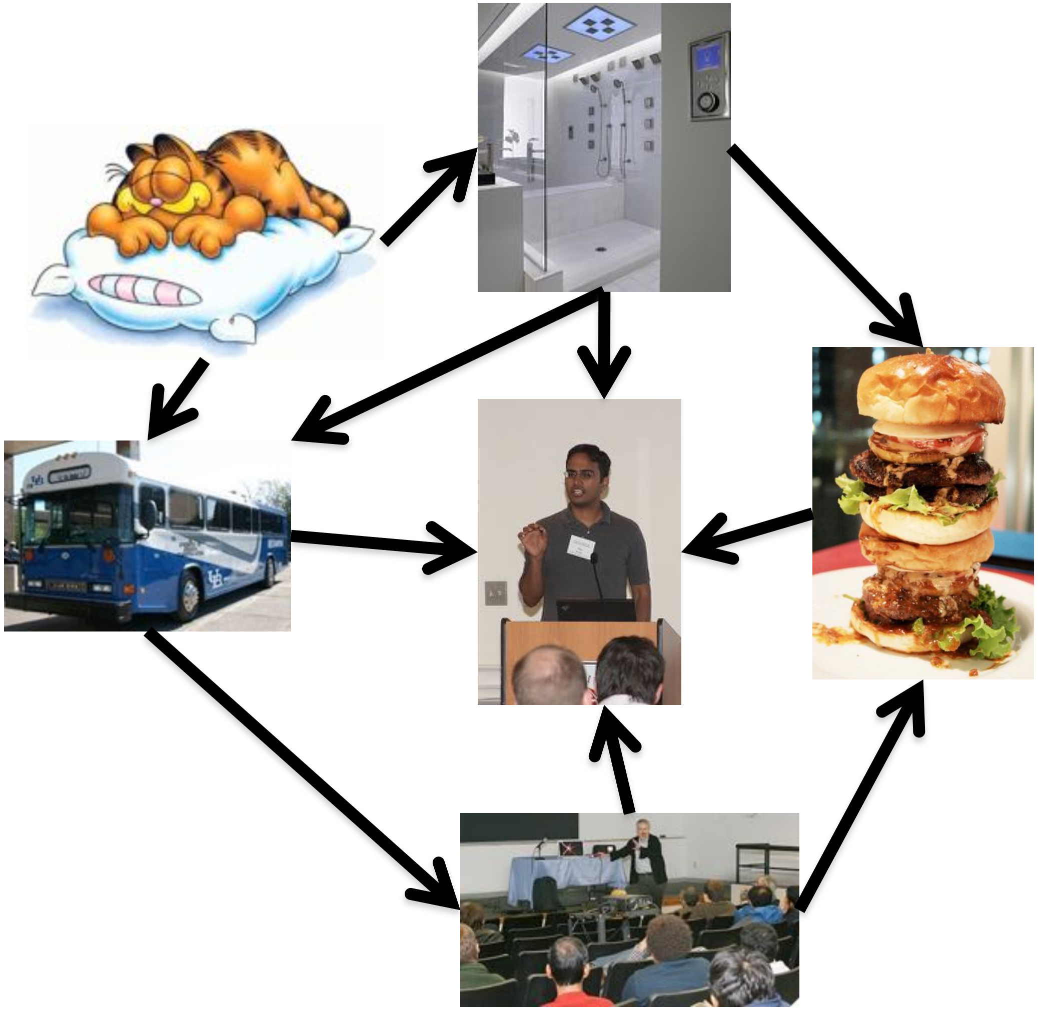
Each vertex has two lists in Adj. list rep.



Directed Acyclic Graph (DAG)

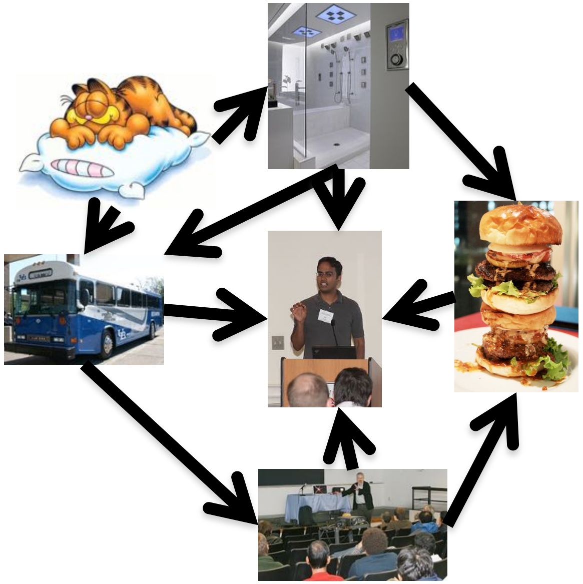
No directed cycles

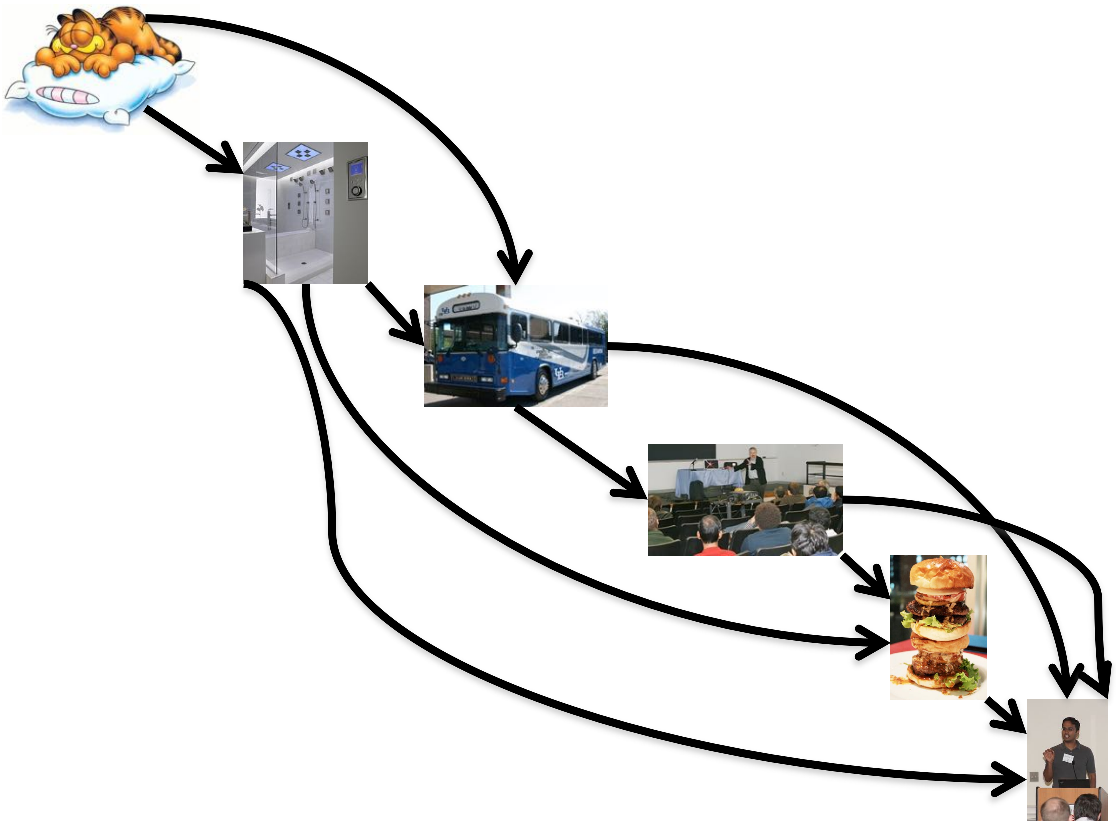
Precedence relationships are consistent



Topological Sorting of a DAG

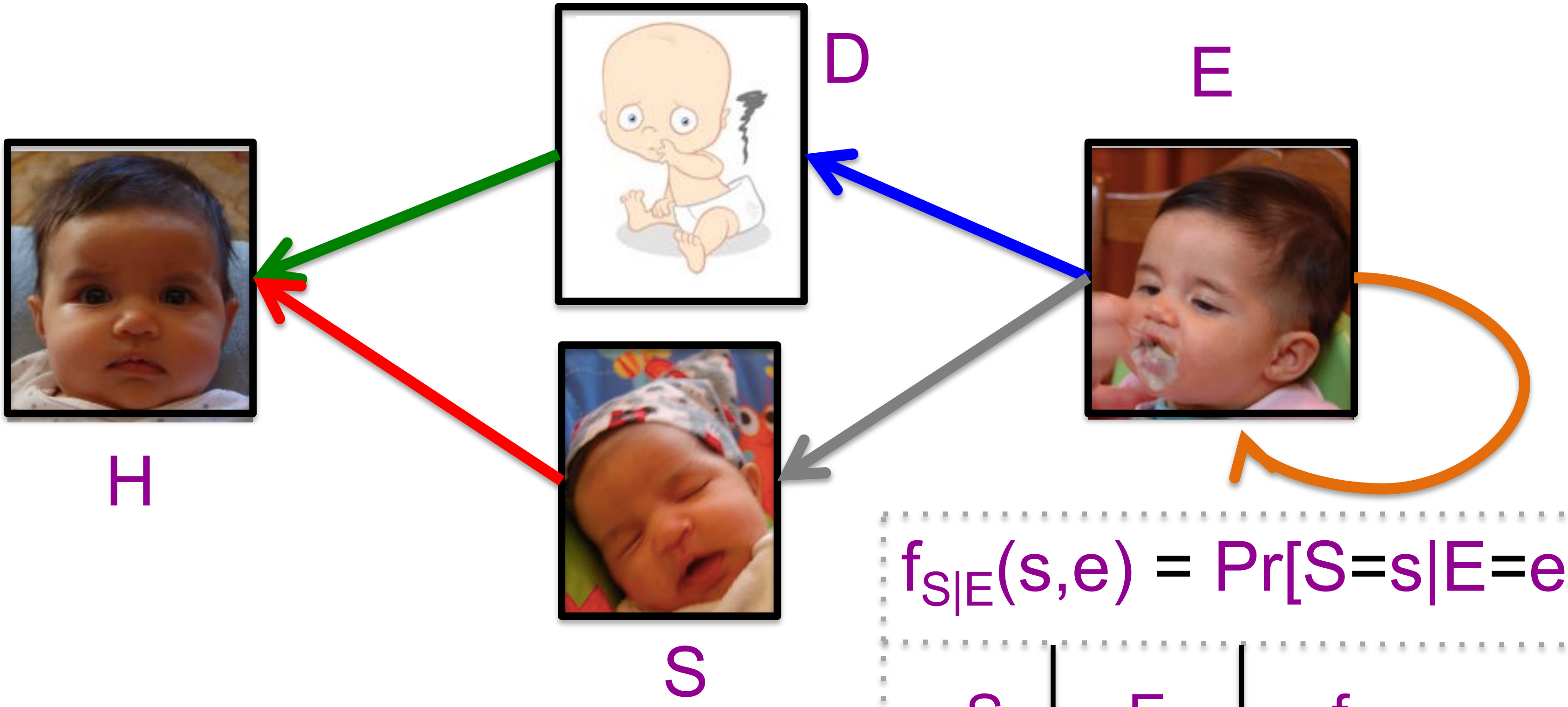
Order the vertices so that all edges go "forward"





Probabilistic Graphical Models (PGMs)

http://ginaskokopelli.com/wp-content/uploads/2013/01/DiaperDealsLogo.jpg

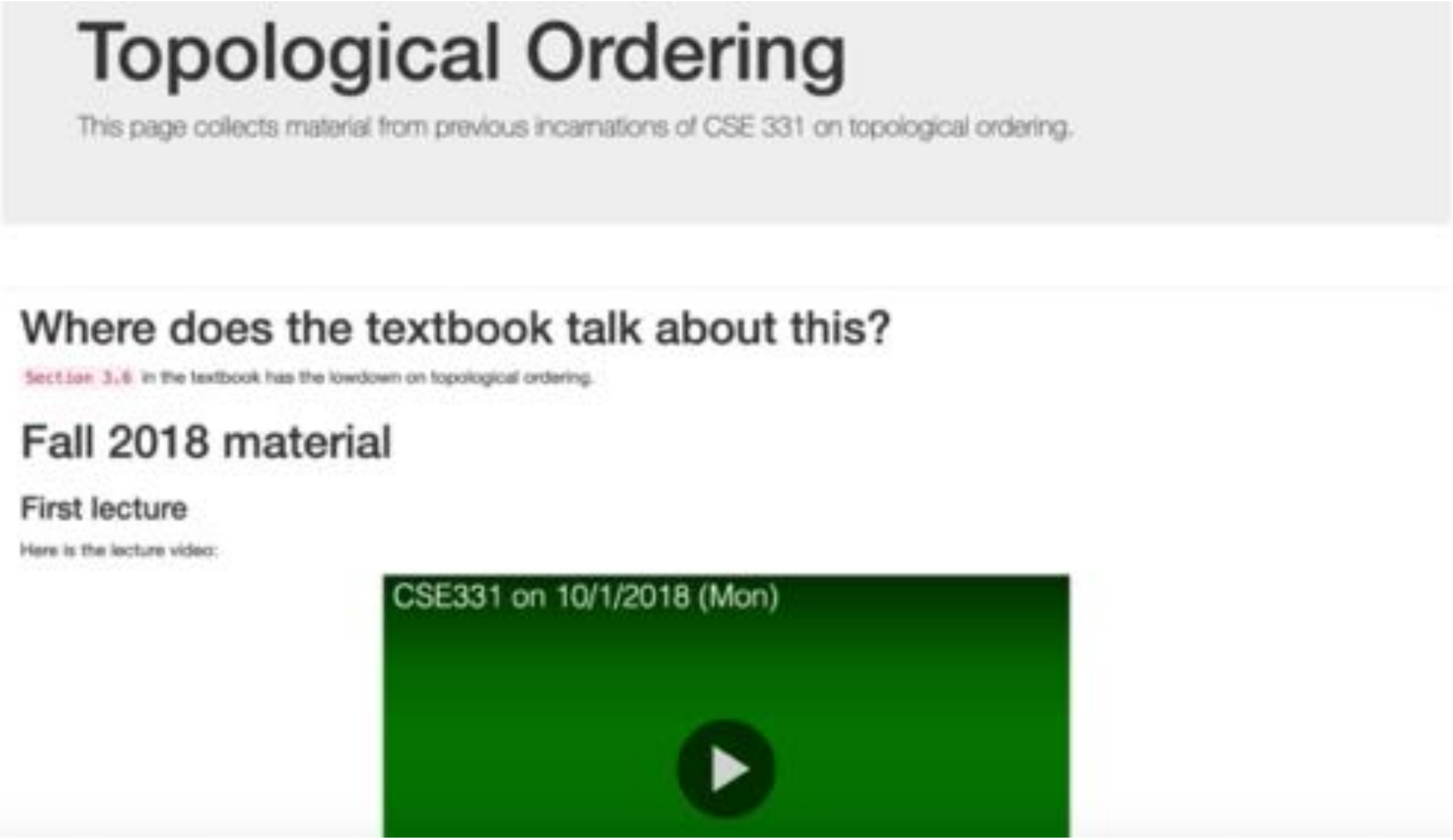


$f_{S|E}(s,e) = \Pr[S{=}s|E{=}e]$

S	E	$f_{S\|E}$
1	1	0.8
1	0	0.3
0	1	0.2
0	0	0.7

$$\varphi\,(h) = \sum_{d,s,e} f_{H|D,S}(h,d,s) \times f_{S|E}(s,e) \times f_{D|E}(d,e) \times f_E(e)$$

More details on Topological sort



Topological Ordering

This page collects material from previous incarnations of CSE 331 on topological ordering.

Where does the textbook talk about this?

`Section 3.6` in the textbook has the lowdown on topological ordering.

Fall 2018 material

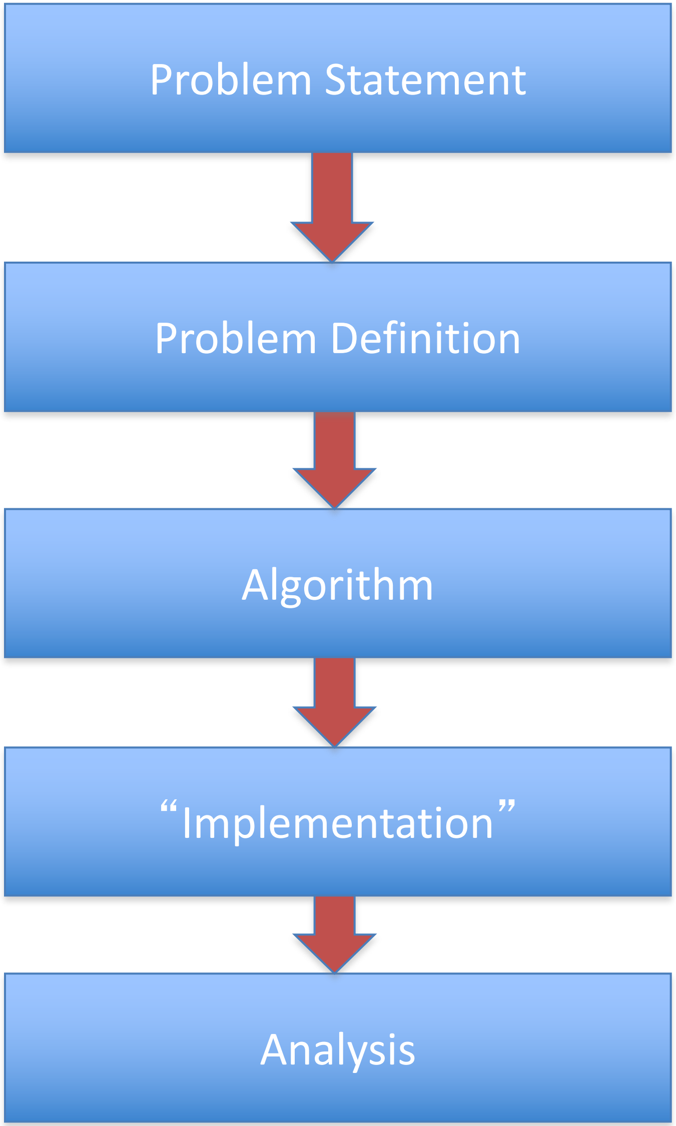
First lecture

Here is the lecture video:

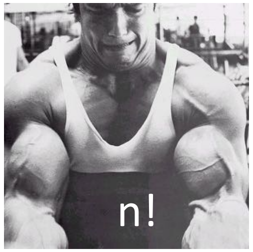
Questions/Comments?

Mid-term material until here

Main Steps in Algorithm Design



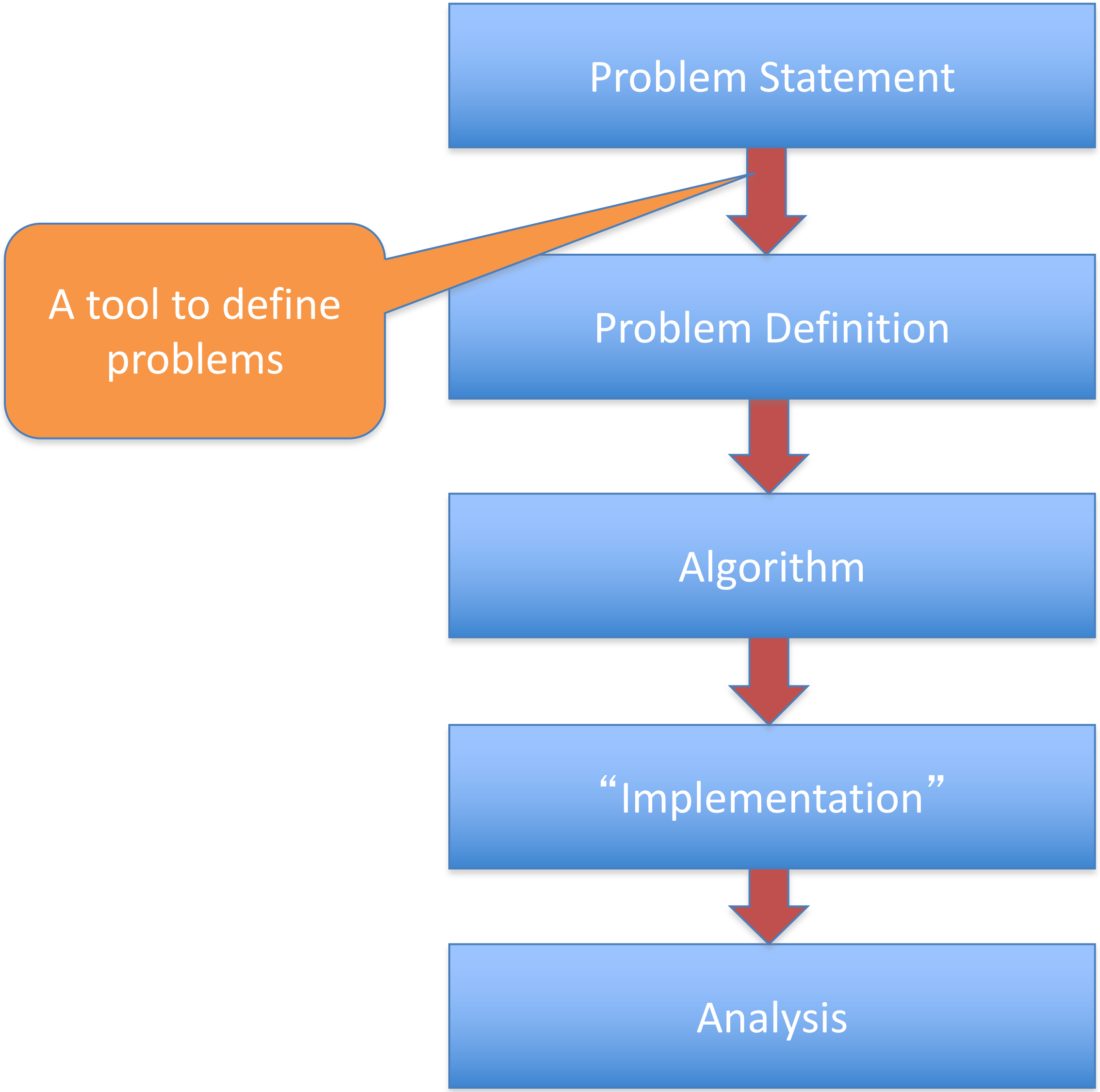
NRMP
National Resident Matching Program



Data Structures

Correctness+Runtime Analysis

Where do graphs fit in?



Problem Statement

A tool to define problems

Problem Definition

Algorithm

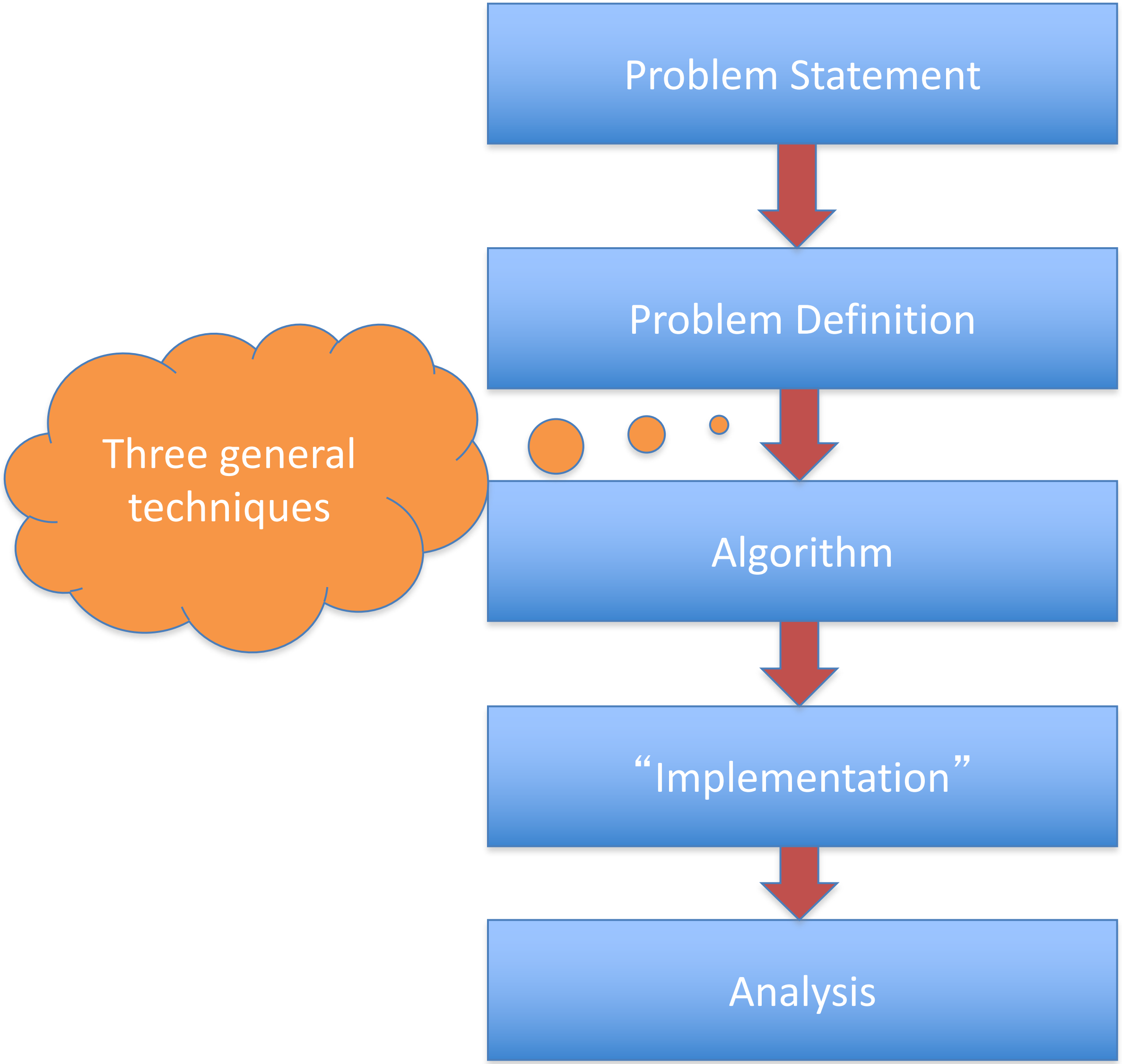
“Implementation”

Data Structures

Analysis

Correctness+Runtime Analysis

Rest of the course*



Problem Statement

Problem Definition

Three general techniques

Algorithm

“Implementation”

Data Structures

Analysis

Correctness+Runtime Analysis

Greedy algorithms

Build the final solution piece by piece

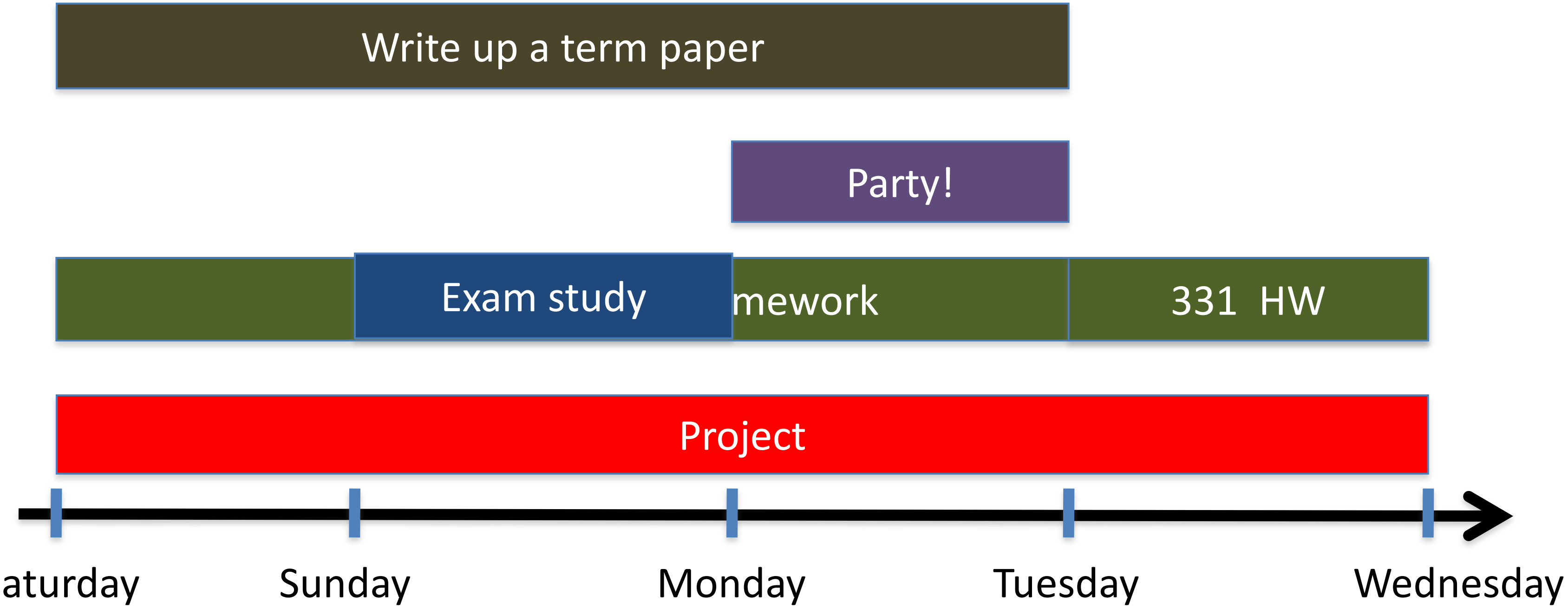
Being short sighted on each piece

Never undo a decision

Know when you see it

End of Semester blues

Can only do one thing at any day: what is the maximum number of tasks that you can do?



Write up a term paper

Party!

Exam study

mework

331 HW

Project

Saturday

Sunday

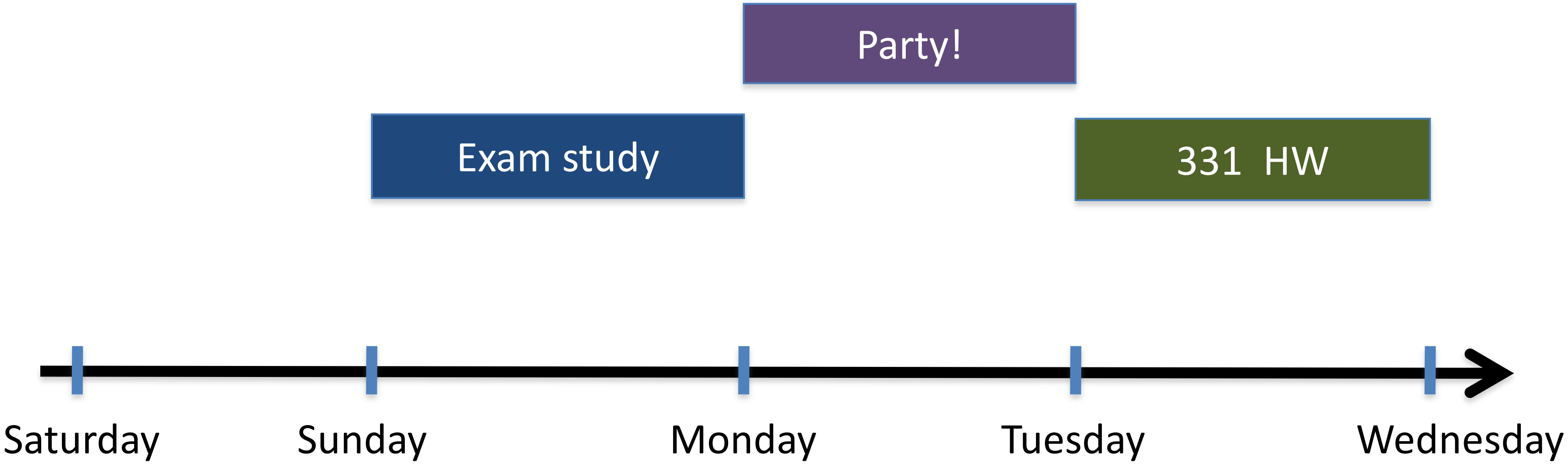
Monday

Tuesday

Wednesday

The optimal solution

Can only do one thing at any day: what is the maximum number of tasks that you can do?



Interval Scheduling Problem

Input: n intervals $[s(i), f(i))$ for $1 \leq i \leq n$

$\{ s(i), \ldots, f(i)-1 \}$

Output: A *schedule* S of the n intervals

No two intervals in S conflict

$|S|$ is maximized