Lecture 15

CSE 331

Oct 2, 2023

Upcoming quiz/exams

Quiz 1 This FRIDAY

Mid-term 1 Wednesday Oct 18

Mid-term 2 Fri two days after Mid-term 1

Bit more on Quiz 1

note @223 stop following **0 views**

Actions

Quiz 1 on Friday, Oct 6

The first quiz will be from **11:00-11:10am in class** on **Friday, October 6**. We will have a 5 mins break after the quiz and the lecture will start at 11:15am.

We will hand out the quiz paper at 10:55am but you will **NOT** be allowed to open the quiz to see the actual questions till 11:00am. 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 T/F polls.) I will post sample mid-terms by Monday night so that you'll be able to see the formatting of such T/F questions.

Also quiz 1 will cover all topics we cover in class until Monday, Oct 2.

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.

quiz1

Edit good note | 0 Updated 2 minutes ago by Atri Rudra

Project Groups formed

note @230 | stop following | 61 views

Actions

Random groups formed + remaining 3 groups

~~Over the next hour or so, I'll be sending email confirmation about the following:~~

I have sent email confirmation to the following groups:

- Random groups
- Groups of size 3 that registered by the deadline

Like in @174 ~~the email will of the following format~~:

- Be on the lookout 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 or with Random group #x in case you asked to be signed up for a random group) and nothing else [apologies for the badly formatted email]

~~I'll post again once I'm done sending out all the information-- so please do not email me BEFORE I post again that I'm done :)~~

If you submitted the form before the deadline but you have not received any email about a groups, please email me ASAP!

The total number of students who signed up to be assigned a random group was divisible by 3 so all random groups are of size 3!

Note that if you already got a confirmation email about your group last week then you will NOT get another confirmation email.

project

Next few weeks are gonna be busy

Mon, Oct 2	Interval Scheduling Problem F22 F21 F19 F18 F17 x^2	[KT, Sec 4.1]
Tue, Oct 3		(**HW 3 in**)
Wed, Oct 4	Greedy Algorithm for Interval Scheduling F22 F21 F19 F18 F17 x^2	[KT, Sec 4.1] (**Project out**) *Reading Assignment:* [KT, Sec 4.1, 4.2]
Fri, Oct 6	Shortest Path Problem F22 F19 F18 F17 x^2	[KT, Sec 4.4] (**Quiz 1**) *Reading Assignment:* Care package on minimizing maximum lateness
Mon, Oct 9	**No class**	Fall break!
Tue, Oct 10		(**HW 4 out**)
Wed, Oct 11	Dijkstra's algorithm F22 F21 F19 F18 F17 x^2	[KT, Sec 4.4]
Fri, Oct 13	Correctness of Dijkstra's Algorithm F22 F21 F19 F18 F17 x^2	[KT, Sec 4.4] *Reading Assignment:* [KT, Sec 4.4]
Mon, Oct 16	Minimum Spanning Tree F22 F21 F19 F18 F17 x^2	[KT, Sec 4.5] *Reading Assignment:* [KT, Sec 4.5]
Tue, Oct 17		(**HW 4 in**)
Wed, Oct 18	**Mid-term exam: I**	
Fri, Oct 20	**Mid-term exam: II**	

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)

Array

Input graph as Adjacency list

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

Set i = 0

Set L_0= {s}

While L_i is not empty

Linked List

$L_{i+1} = \emptyset$

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++

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≠ 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

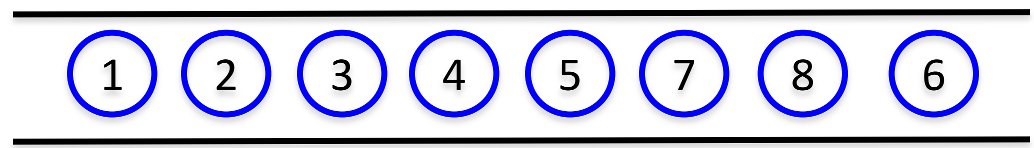
Add w to L_{i+1}

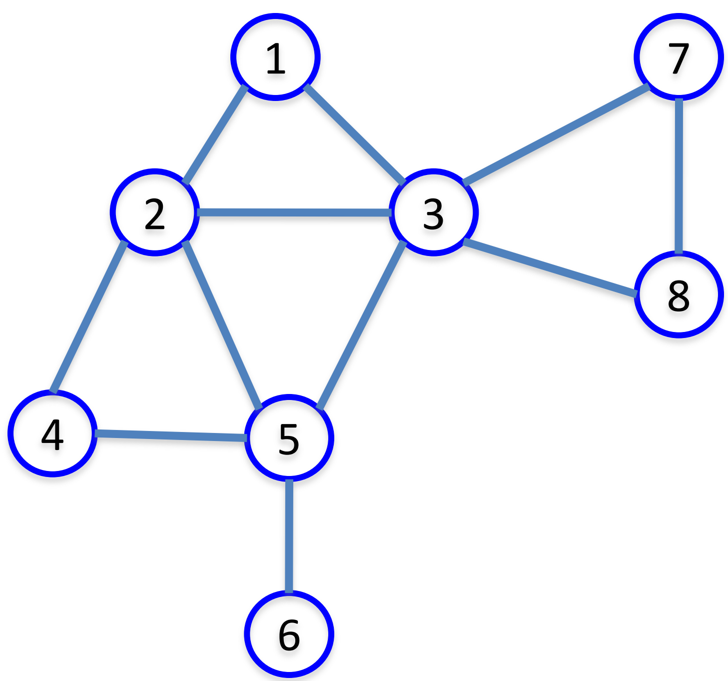
i++

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 — $O(n)$

Intitialize Q= {s} — $O(1)$

While Q is not empty — Repea… onc…

    Delete the front element u in Q — $O(1)$

        For every edge (u,w) — $O(1)$

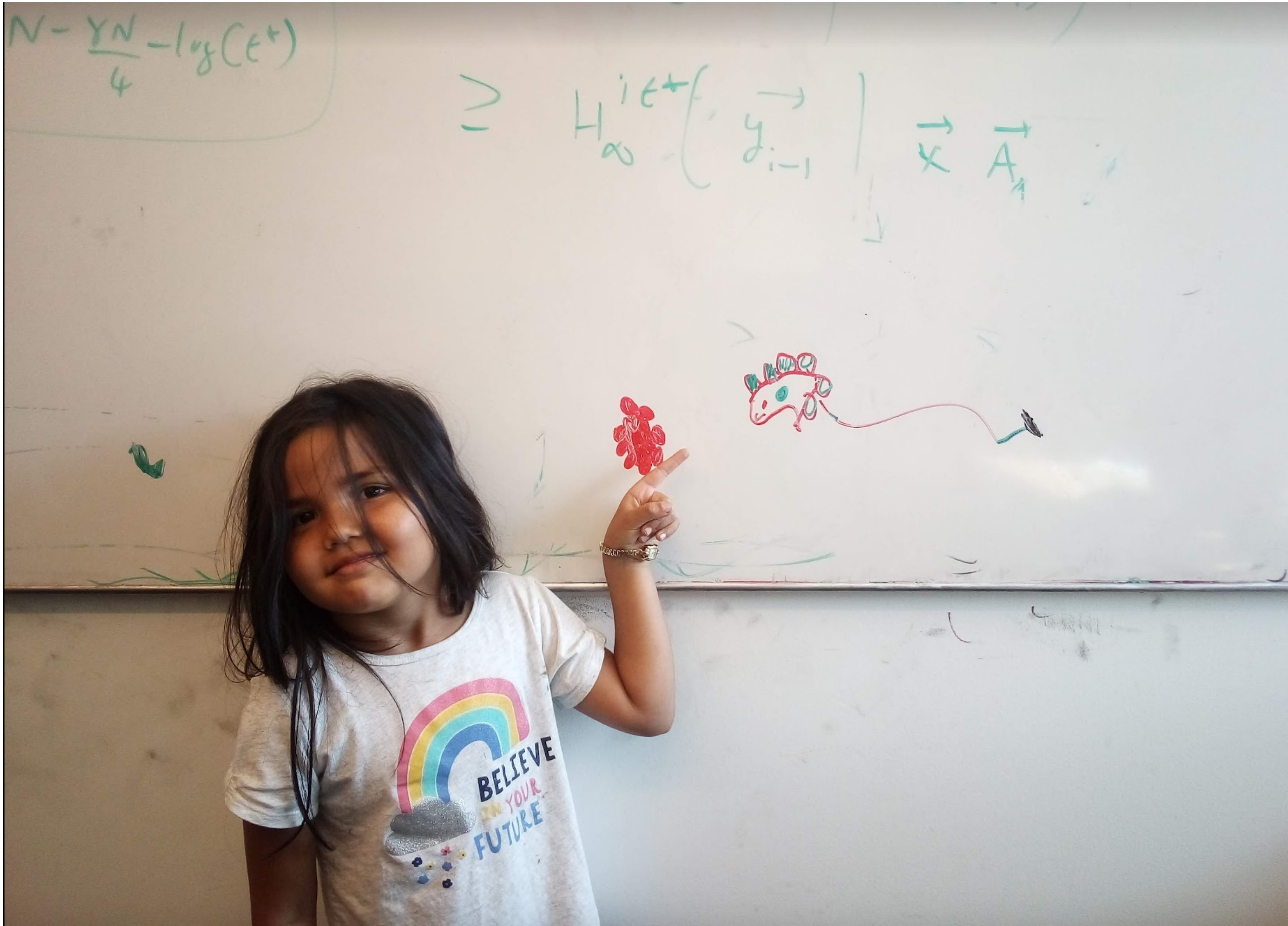
            If CC[w] = F then — $O(n_u)$

                CC[w] = T — $O(1)$

                Add w to the back of Q

$\Sigma_u O(n_u) = O(\Sigma_u n_u) = O(m)$

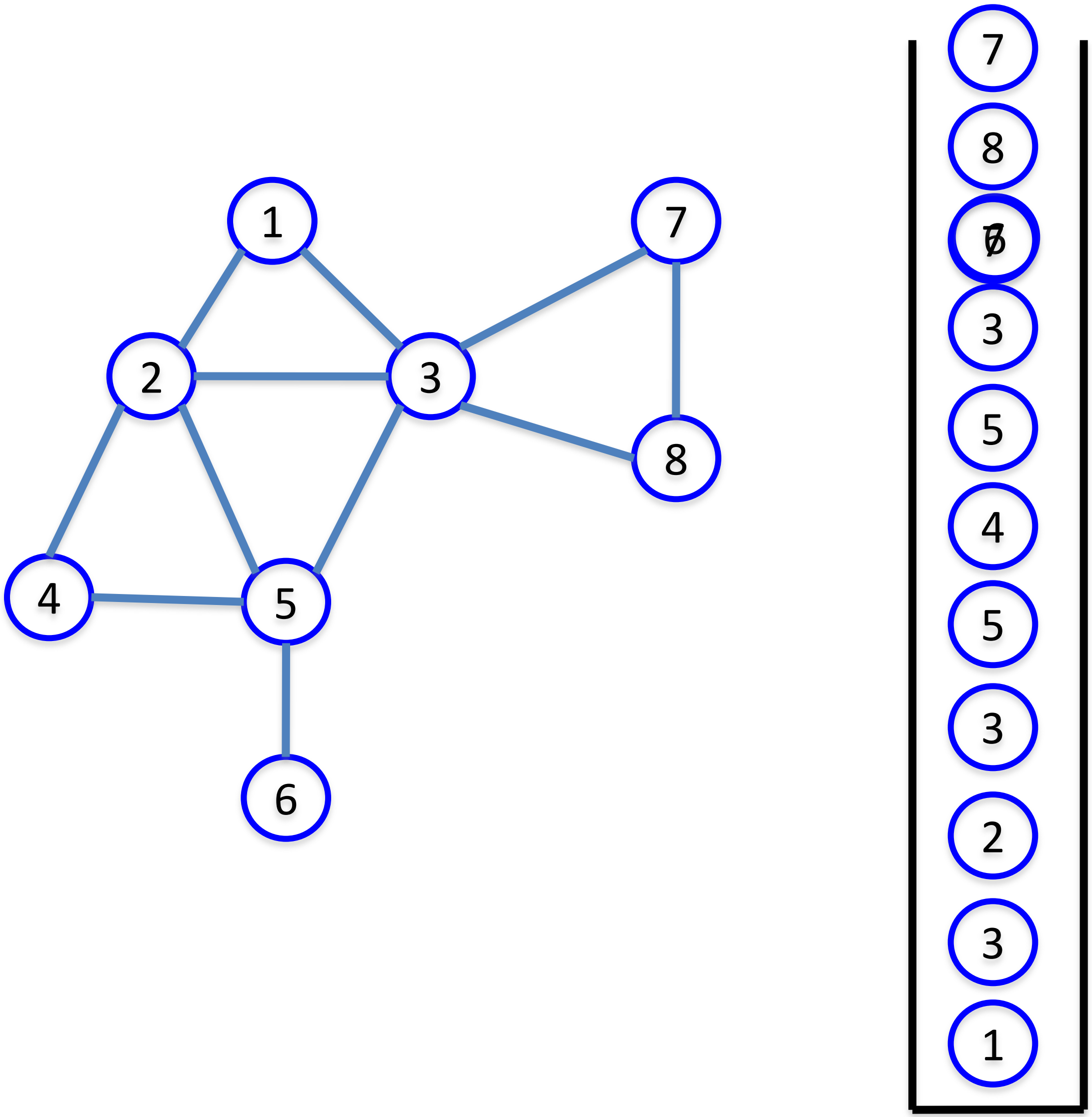
Questions/Comments?



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

If CC[u] = F then

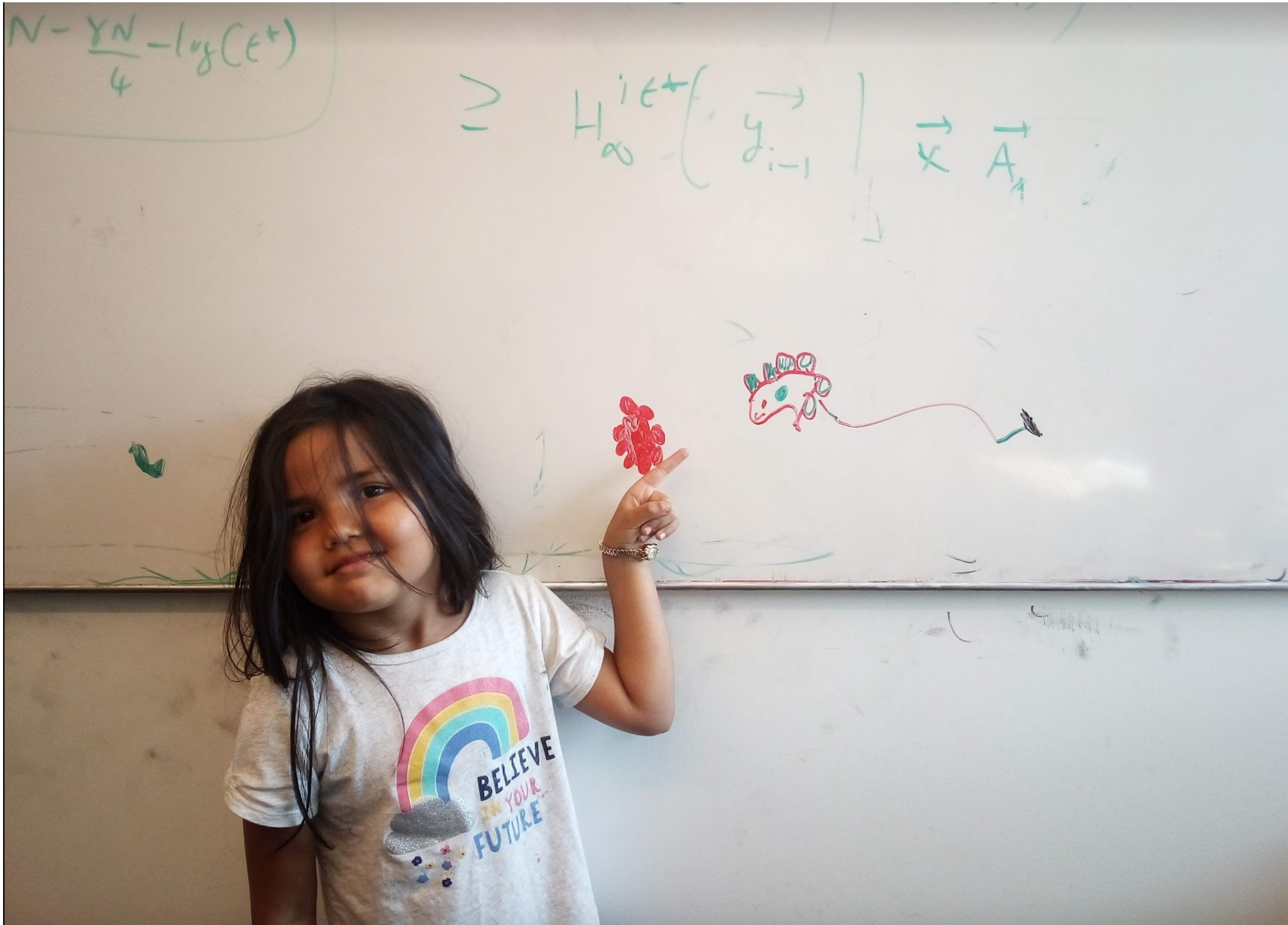
CC[u] = T

For every edge (u,w)

Push w to the top of Ŝ

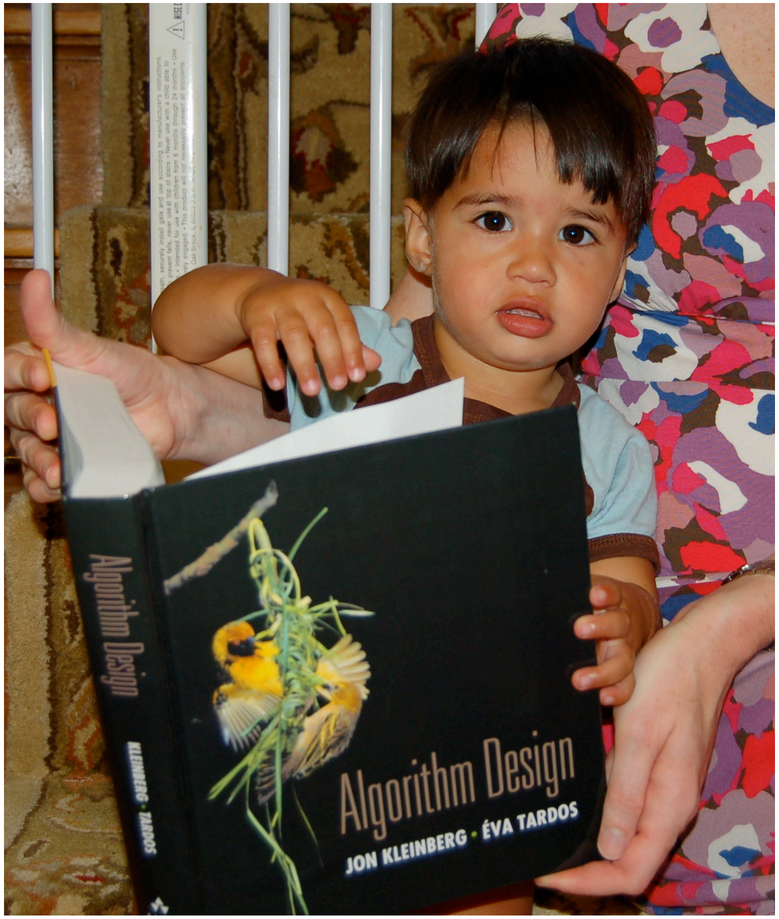
Same O(m+n) run time analysis as for BFS

Questions/Comments?



Reading Assignment

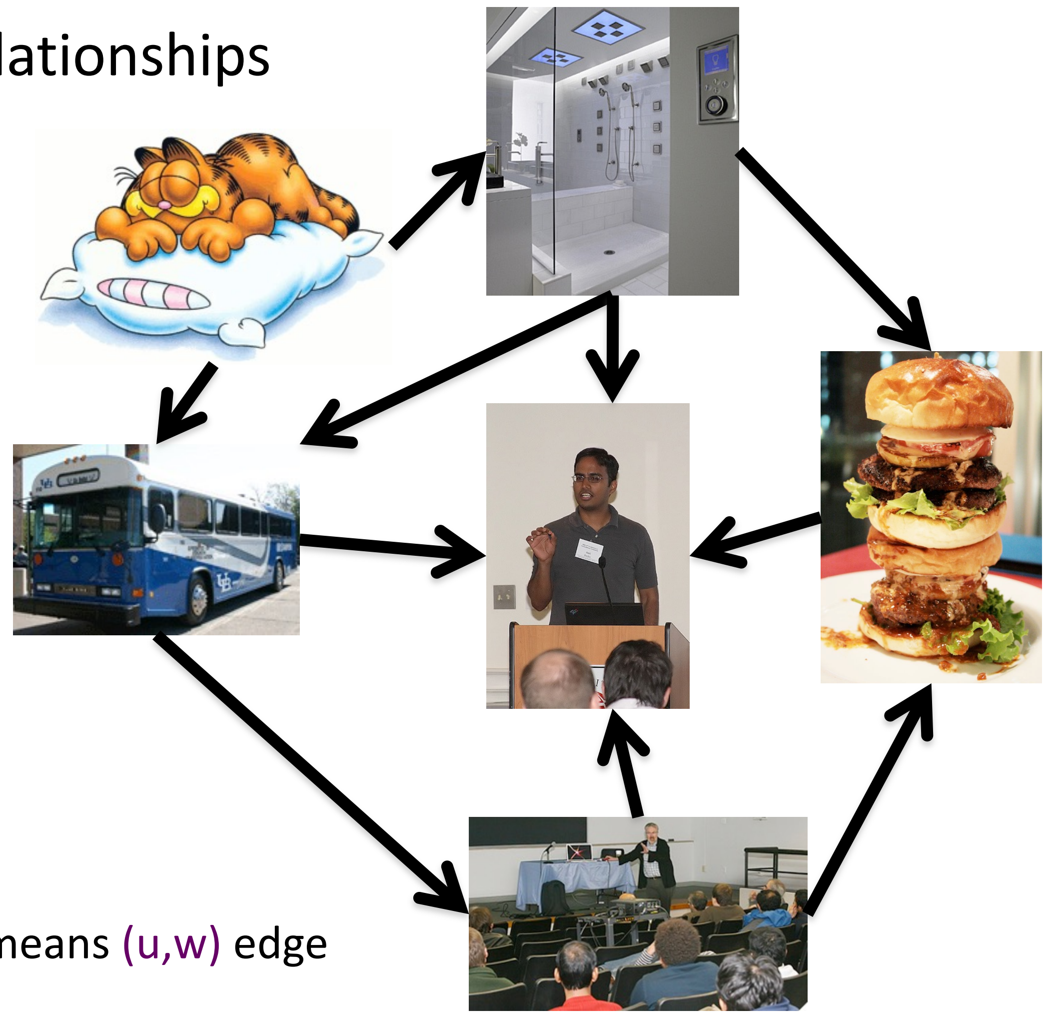
Sec 3.3, 3.4, 3.5 and 3.6 of [KT]



Directed graphs

Model asymmetric relationships

Precedence relationships

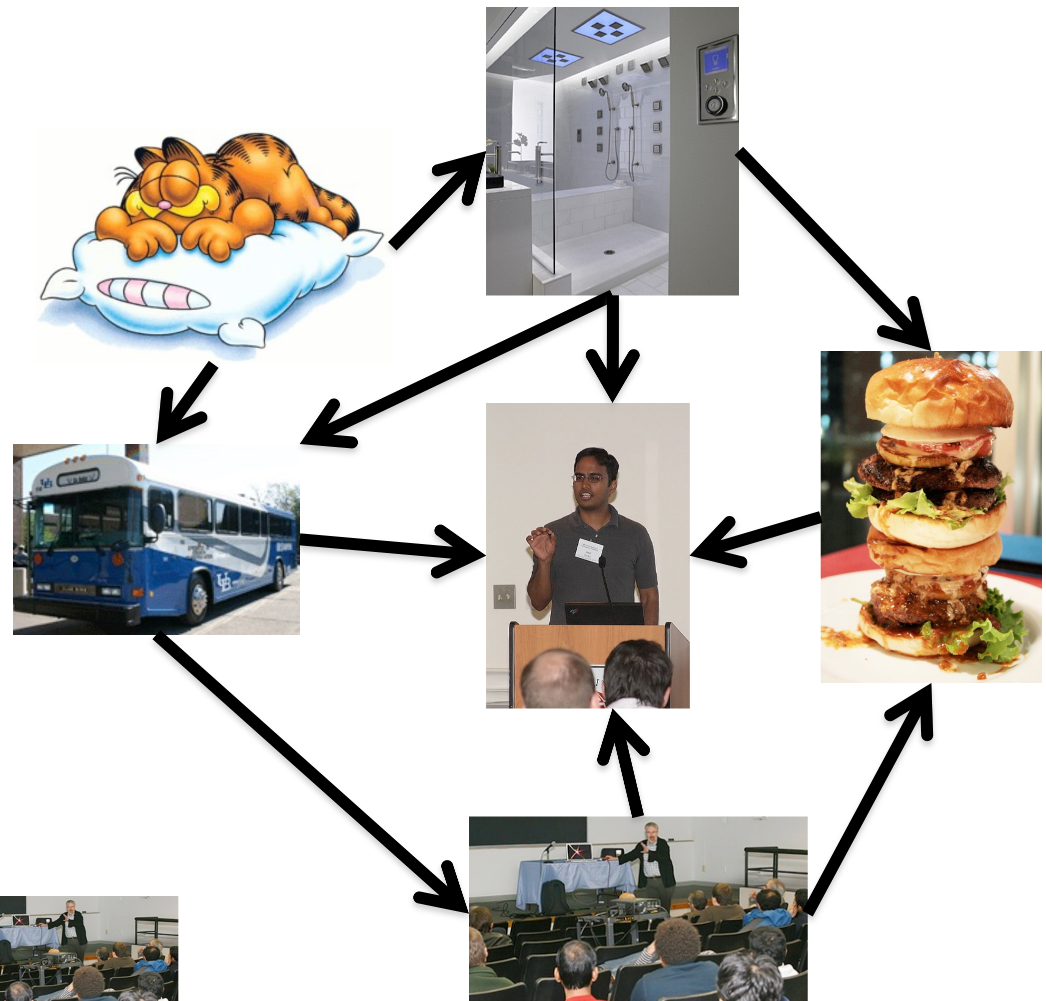


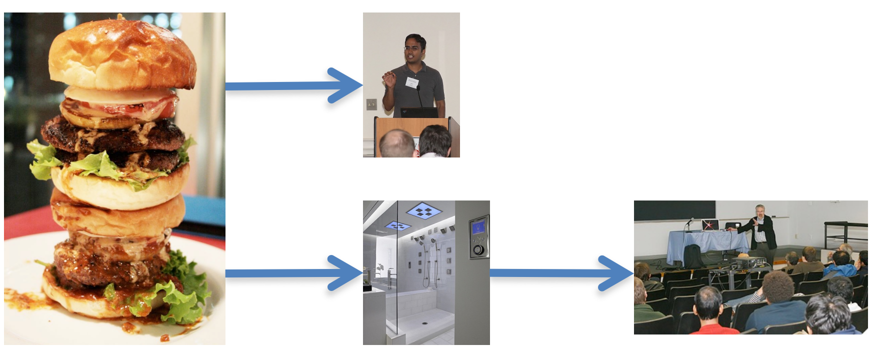
u needs to be done before w means (u,w) edge

Directed graphs

Adjacency matrix is not symmetric

Each vertex has two lists in Adj. list rep.

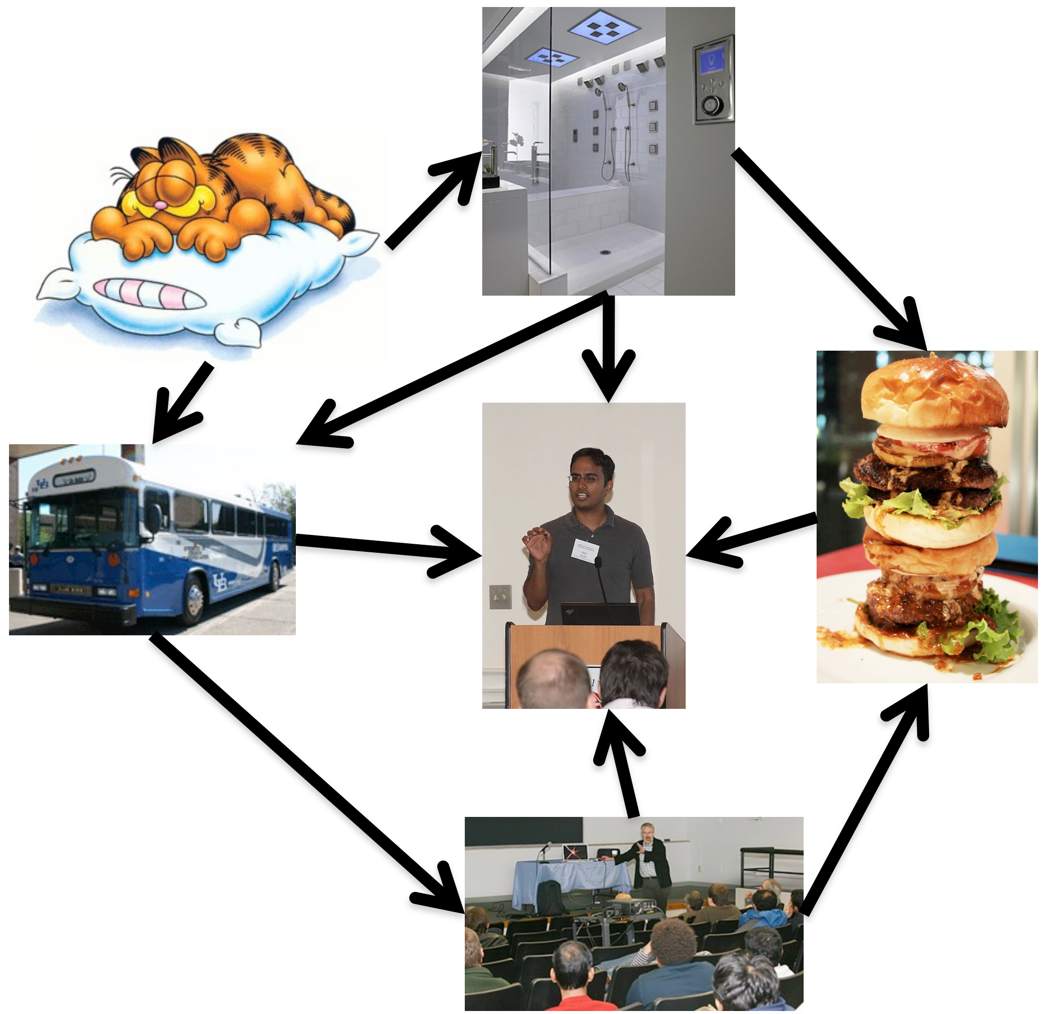




Directed Acyclic Graph (DAG)

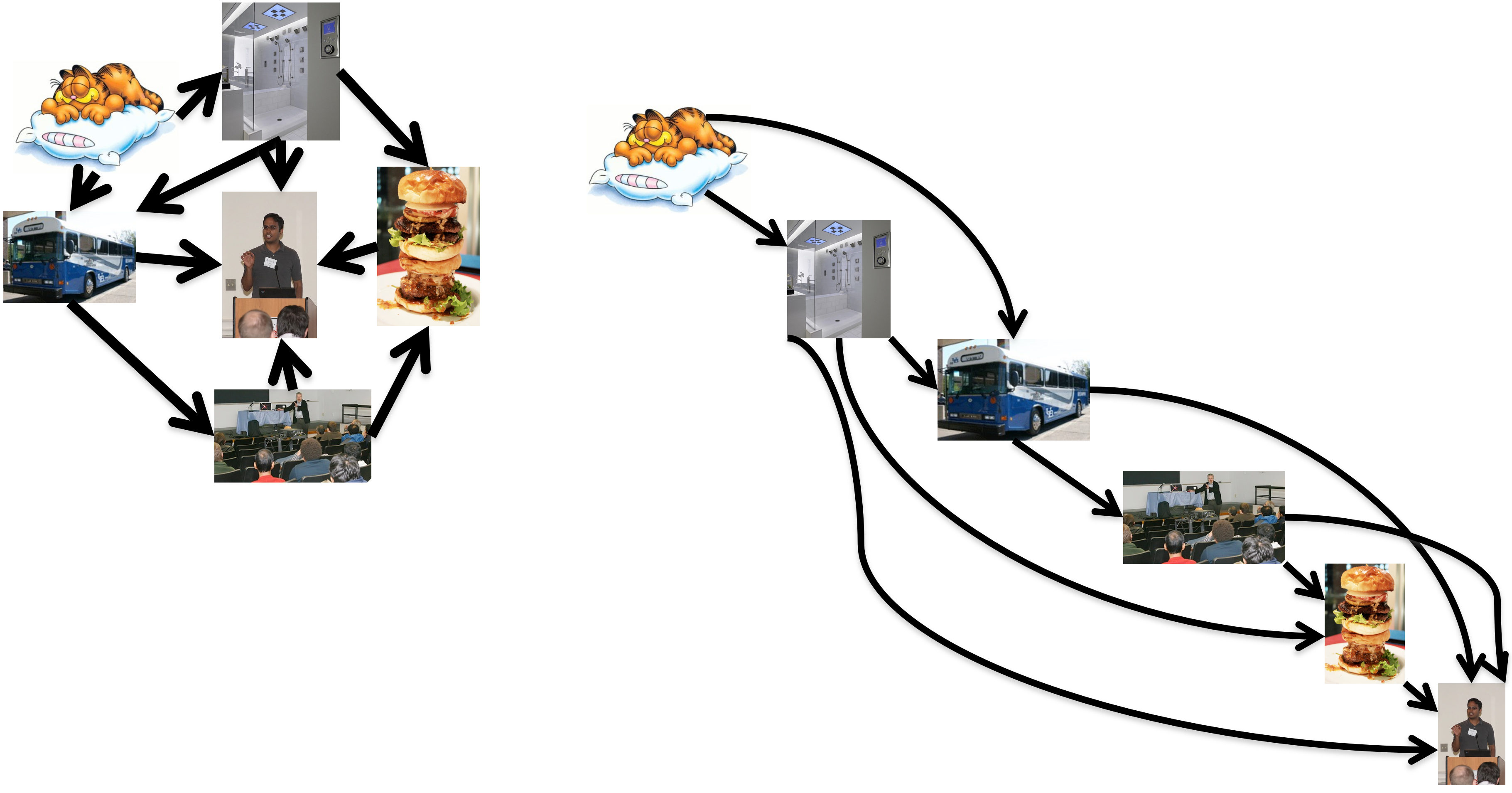
No directed cycles

Precedence relationships are consistent



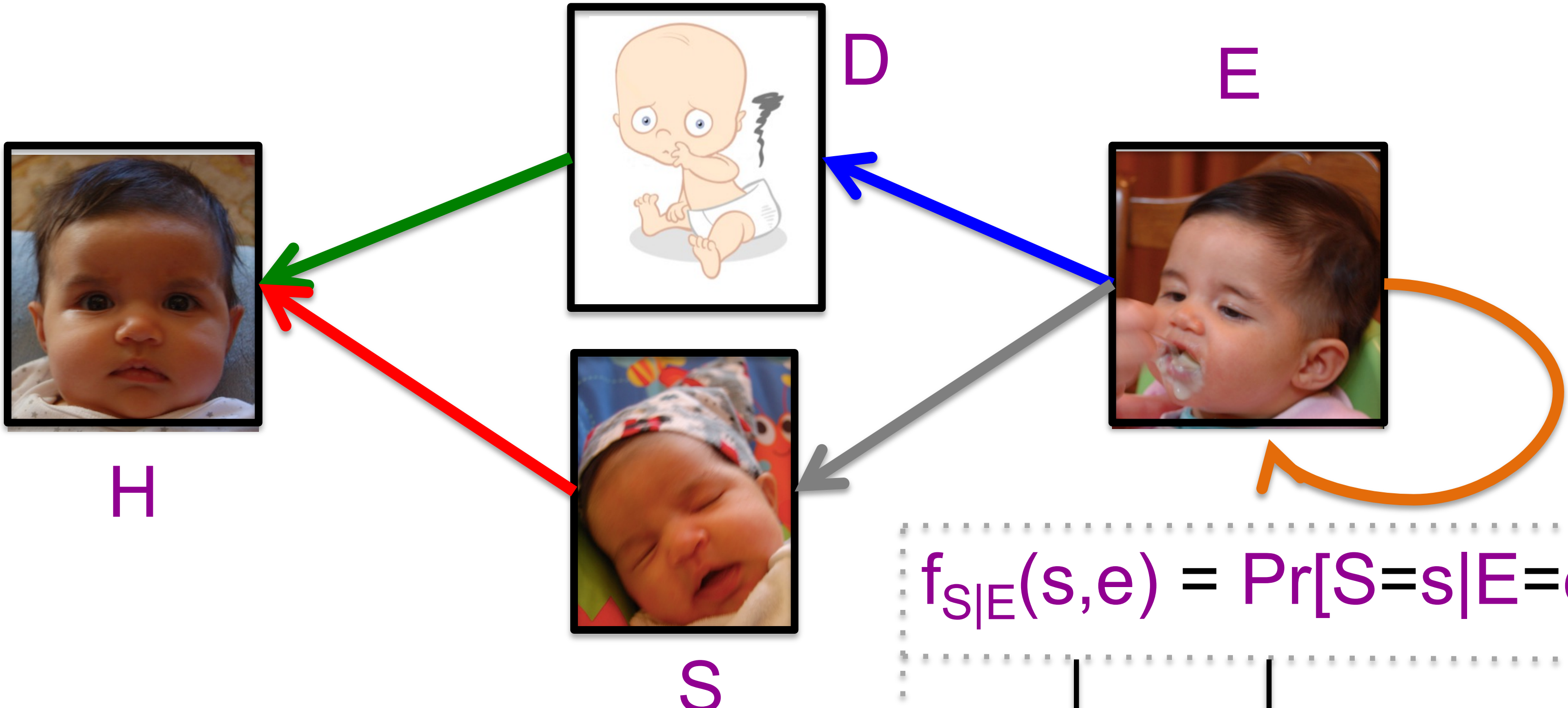
Topological Sorting of a DAG

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.

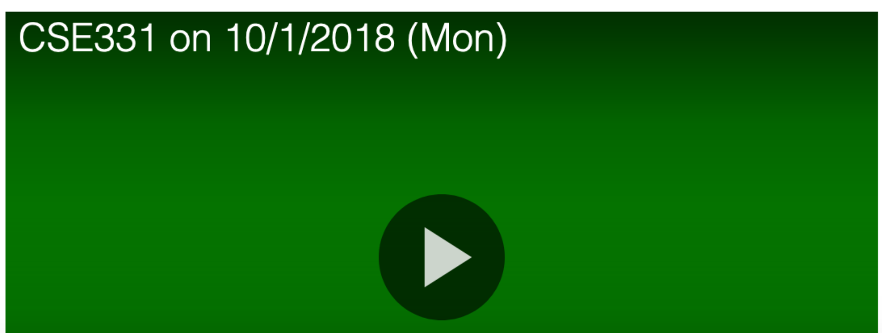
Where does the textbook talk about this?

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

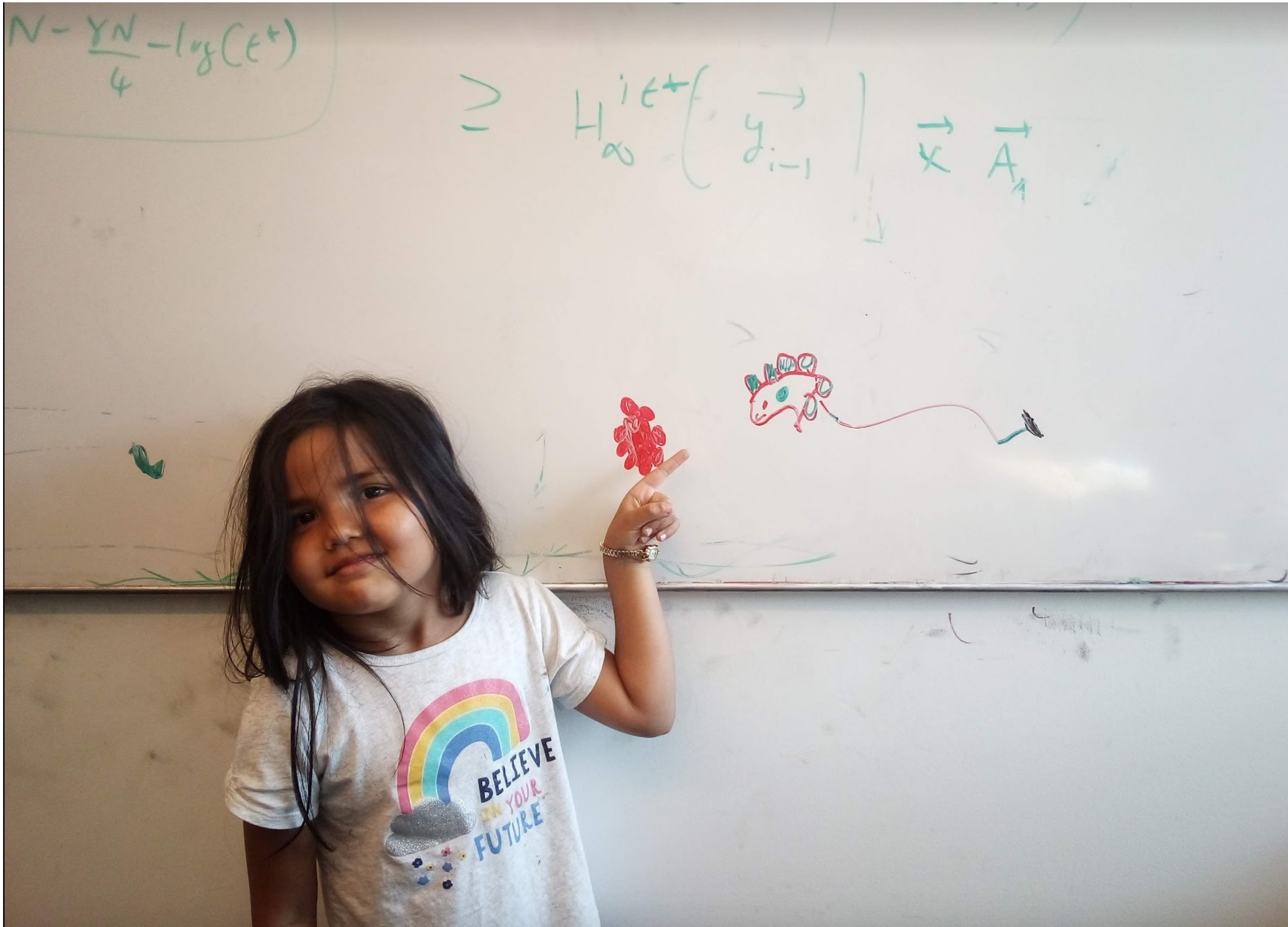
Fall 2018 material

First lecture

Here is the lecture video:



Questions/Comments?



Mid-term material until here

Main Steps in Algorithm Design

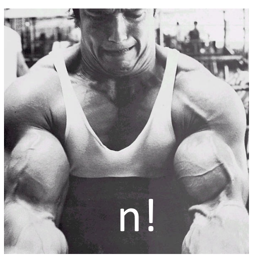
Problem Statement

NRMP
National Resident Matching Program

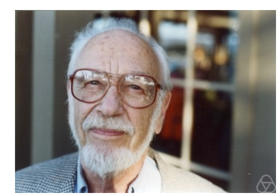
↓

Problem Definition

↓



Algorithm



↓

“Implementation”

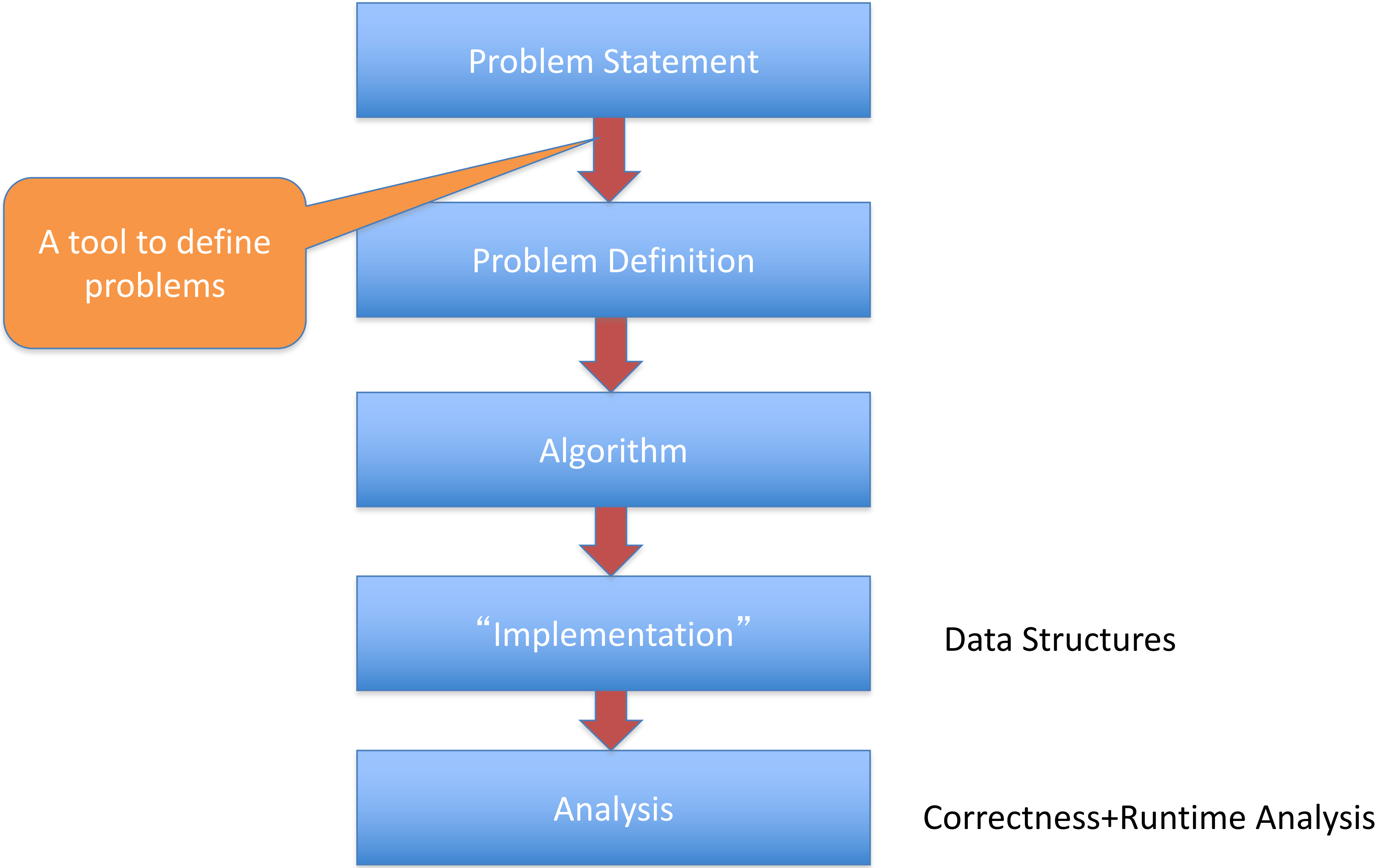
Data Structures

↓

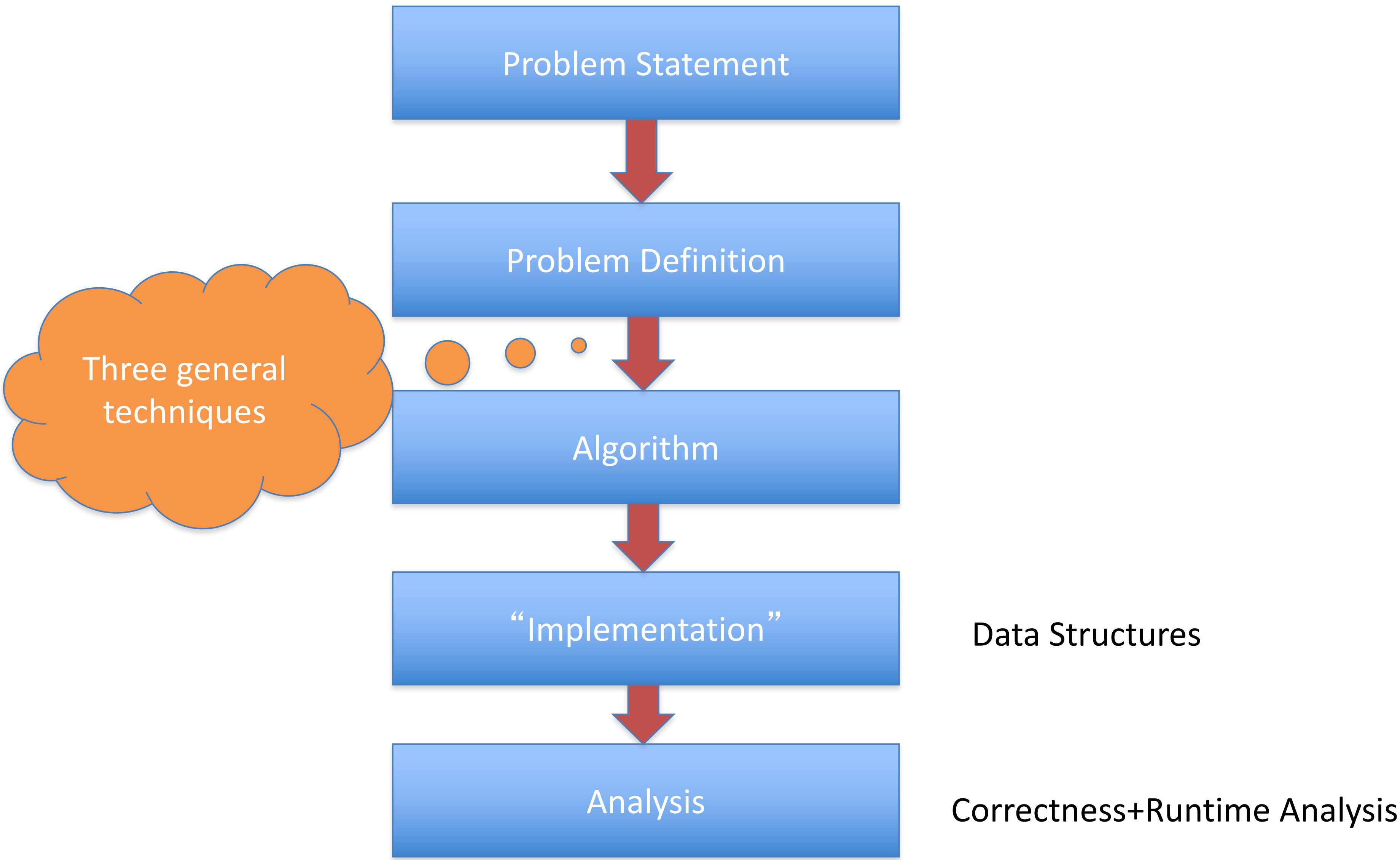
Analysis

Correctness+Runtime Analysis

Where do graphs fit in?

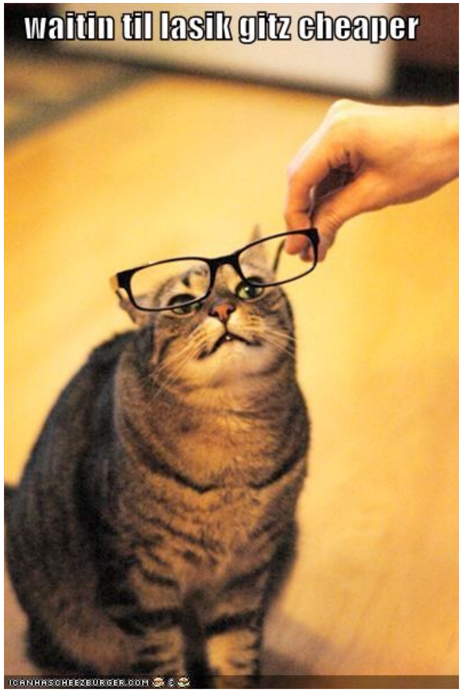


Rest of the course*



Greedy algorithms

Build the final solution piece by piece



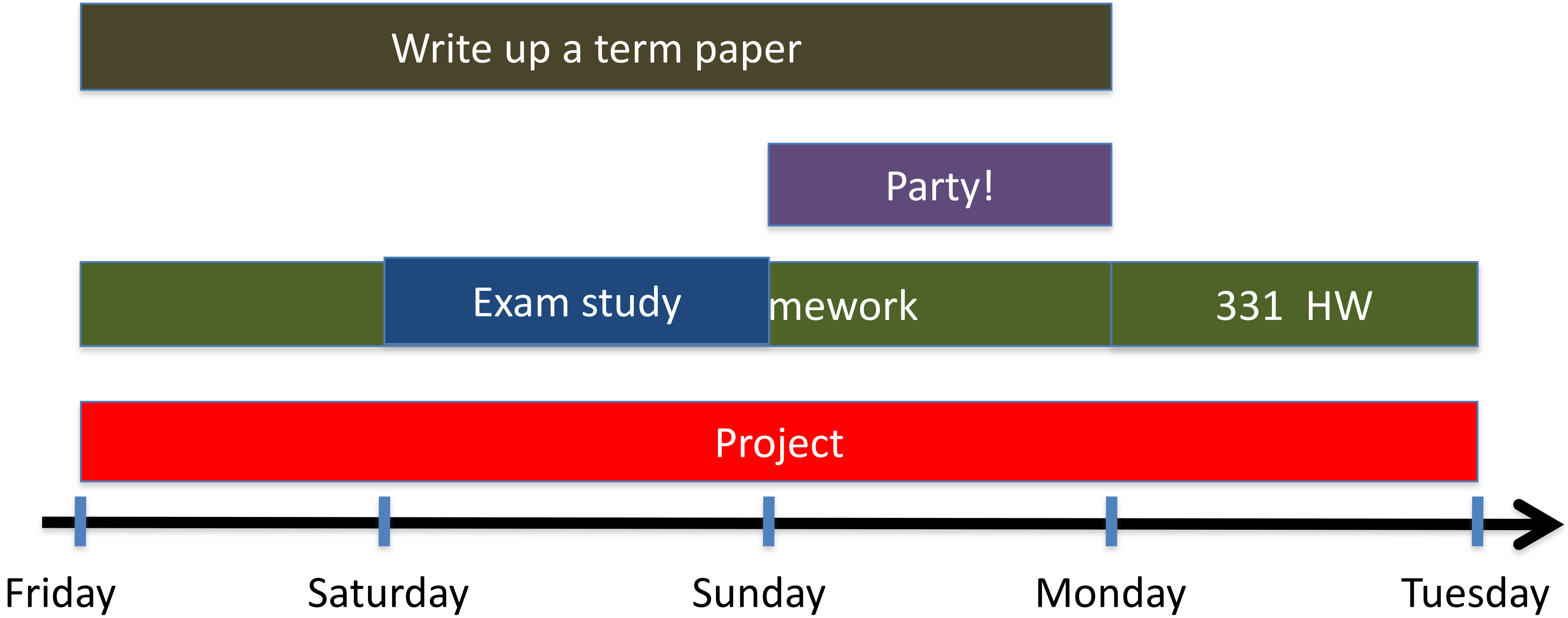
Being short sighted on each piece

Never undo a decision

Know when you see it

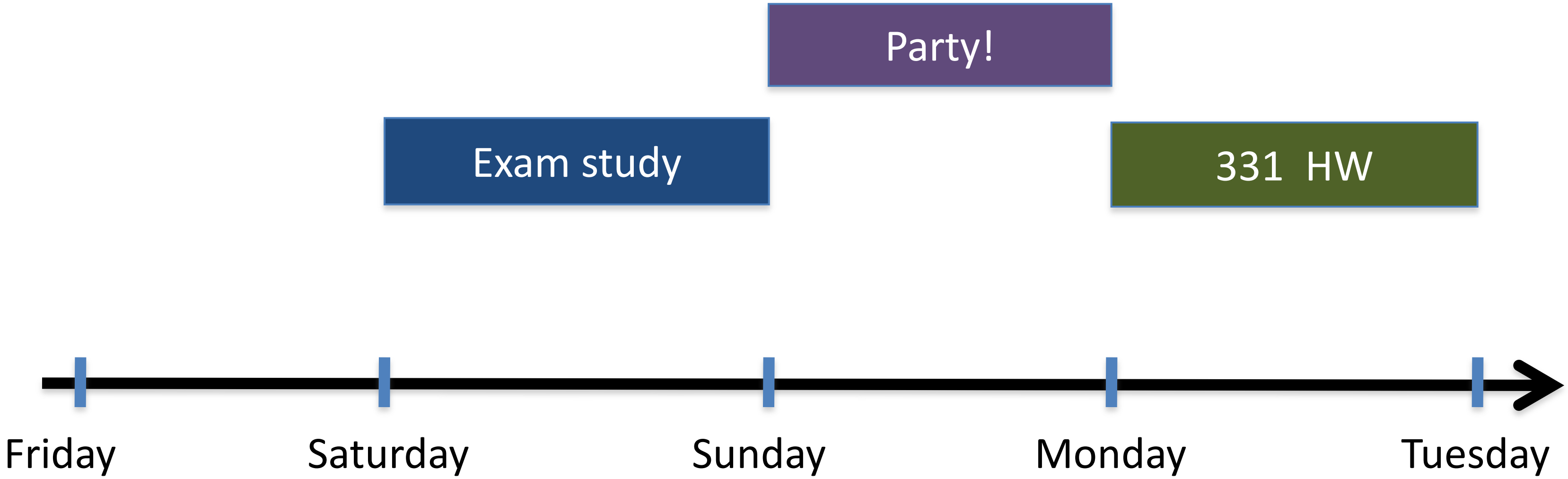
End of Semester blues

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



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 $1 \le i \le n$

{ s(i), … ,f(i)-1 }

Output: A *schedule* S of the n intervals

No two intervals in S conflict

$|S|$ is maximized

Algorithm with examples

Interval Scheduling via examples

In which we derive an algorithm that solves the Interval Scheduling problem via a sequence of examples.

The problem

In these notes we will solve the following problem:

Interval Scheduling Problem

`Input:` An input of n intervals $[s(i), f(i))$, or in other words, $\{s(i), \ldots, f(i) - 1\}$ for $1 \leq i \leq n$ where i represents the intervals, $s(i)$ represents the start time, and $f(i)$ represents the finish time.

`Output:` A schedule S of n intervals where no two intervals in S conflict, and the total number of intervals in S is maximized.

Sample Input and Output

`Input:`