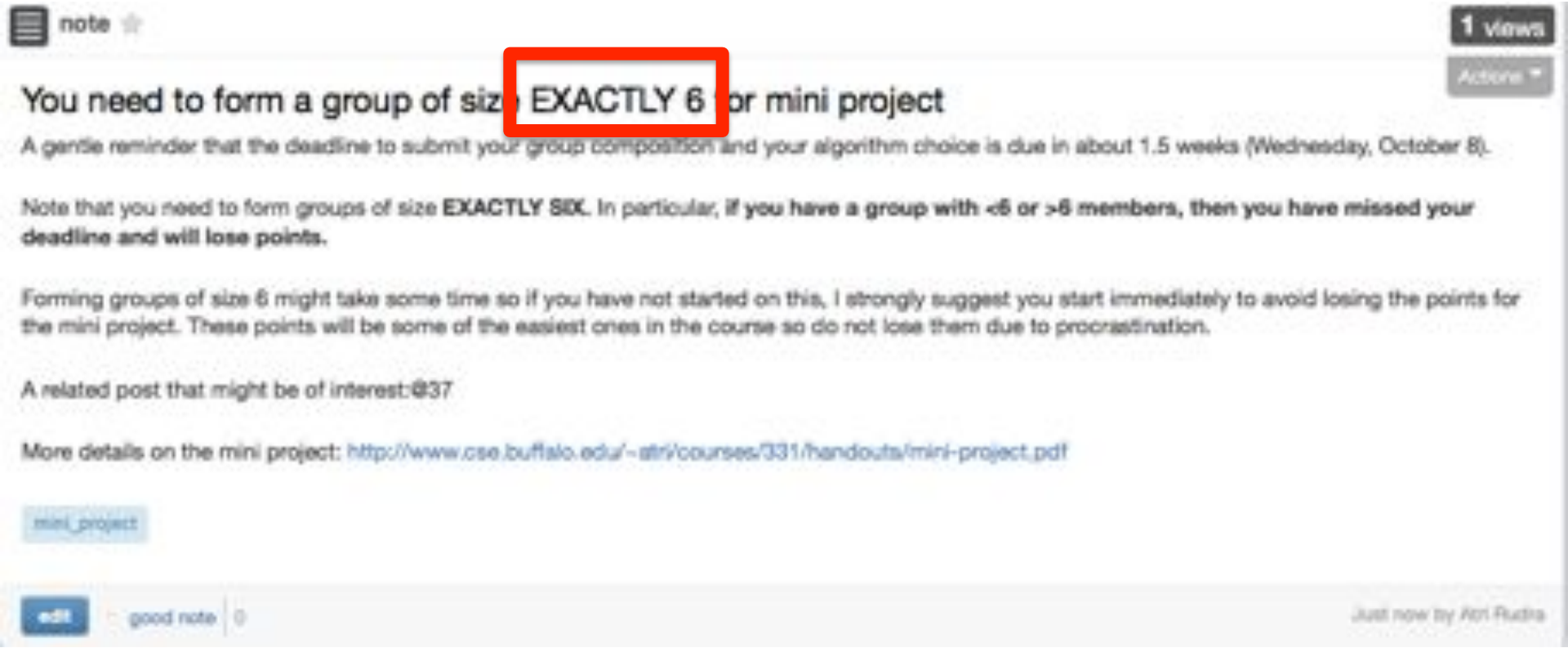
Lecture 19

CSE 331

Oct 8, 2014

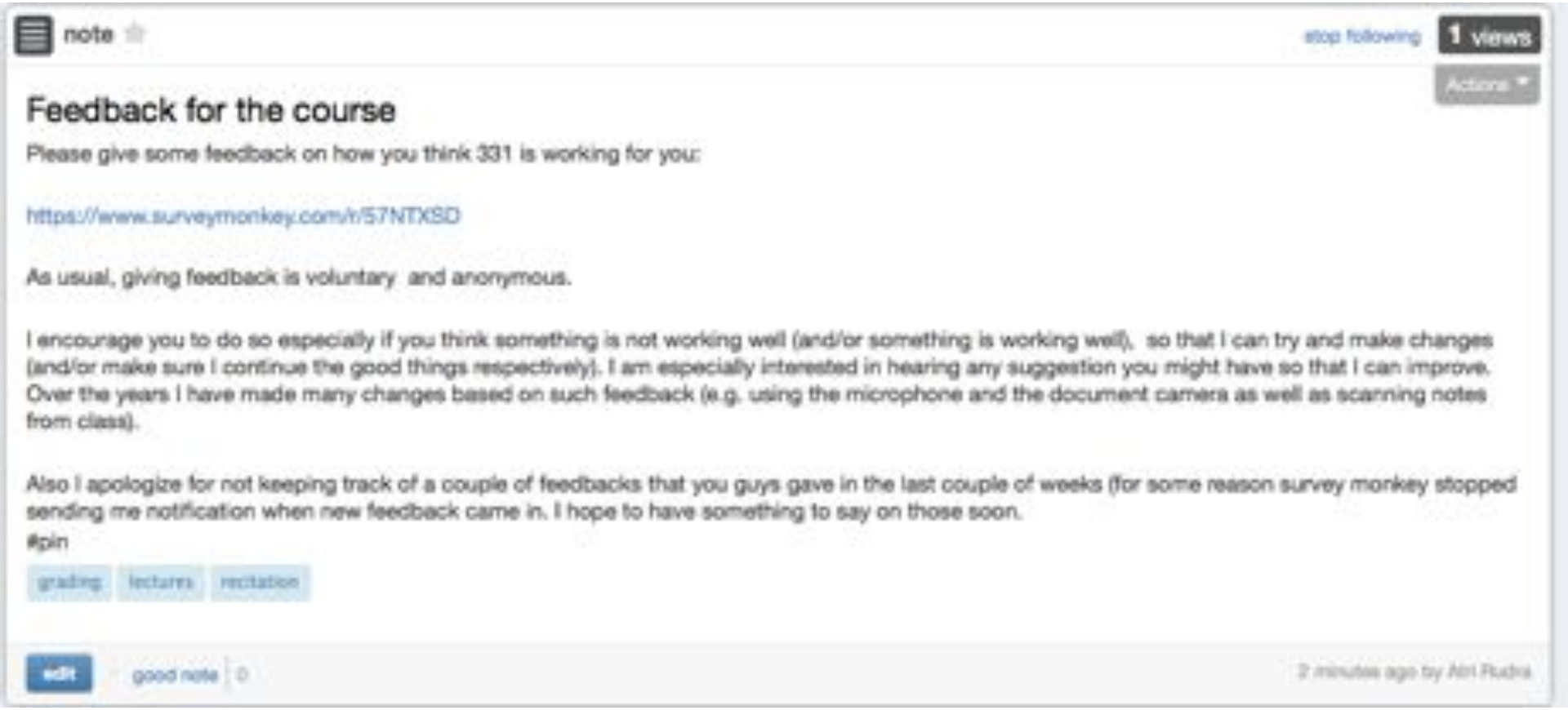
Group/Algo registration deadline

BOTH TONIGHT by 11:59pm!



Online OH 9-10pm tonight

Feedback


note

stop following

1 views

Actions

Feedback for the course

Please give some feedback on how you think 331 is working for you:

https://www.surveymonkey.com/r/57NTXSD

As usual, giving feedback is voluntary and anonymous.

I encourage you to do so especially if you think something is not working well (and/or something is working well), so that I can try and make changes (and/or make sure I continue the good things respectively). I am especially interested in hearing any suggestion you might have so that I can improve. Over the years I have made many changes based on such feedback (e.g. using the microphone and the document camera as well as scanning notes from class).

Also I apologize for not keeping track of a couple of feedbacks that you guys gave in the last couple of weeks (for some reason survey monkey stopped sending me notification when new feedback came in. I hope to have something to say on those soon.

#pin

grading lectures recitation

edit good note | 0

2 minutes ago by Atri Rudra

Interval Scheduling Problem

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

Output: A *schedule* S of the n intervals

No two intervals in S conflict

|S| is maximized

Analyzing the algorithm

R: set of requests

Set A to be the empty set

While R is not empty

Choose i in R with the earliest finish time

Add i to A

Remove all requests that conflict with i from R

Return A* = A

A* has no conflicts

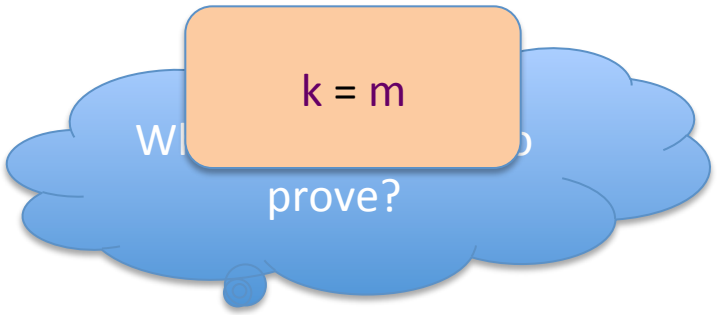
A* is an optimal solution

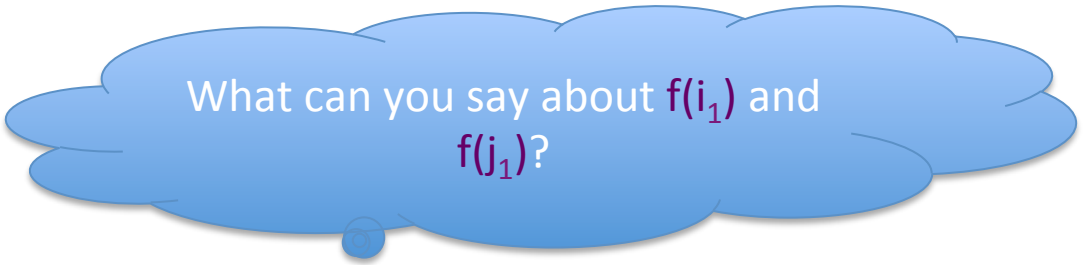
Greedy “stays ahead”

Greedy "stays ahead"

$A^* = i_1,\ldots,i_k$

$O = j_1,\ldots,j_m$





A formal lemma

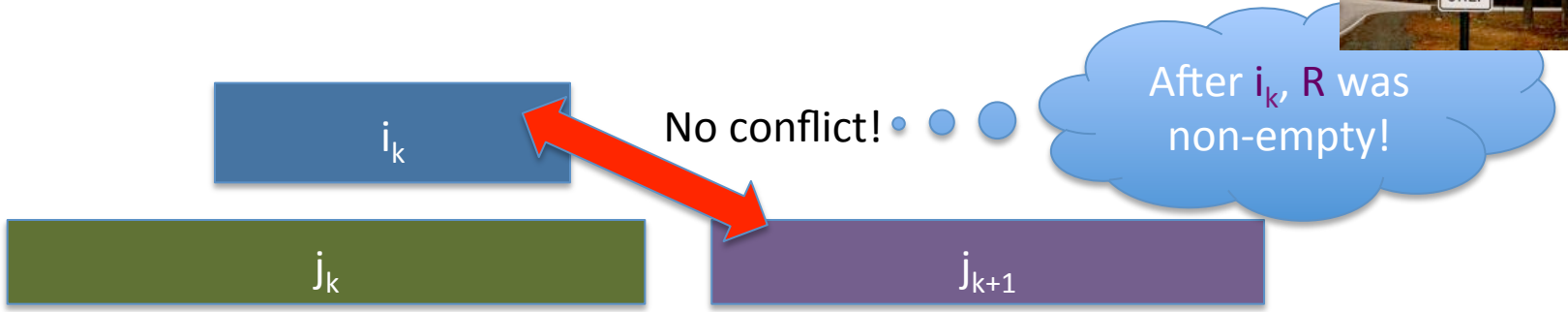
For every $r \le k$, $f(i_r) \le f(j_r)$

The greedy algorithm outputs an optimal A

Proof by contradiction: A is not optimal

$A^* = i_1,\ldots,i_k$ $\quad$ $O = j_1,\ldots,j_m$

$m > k$



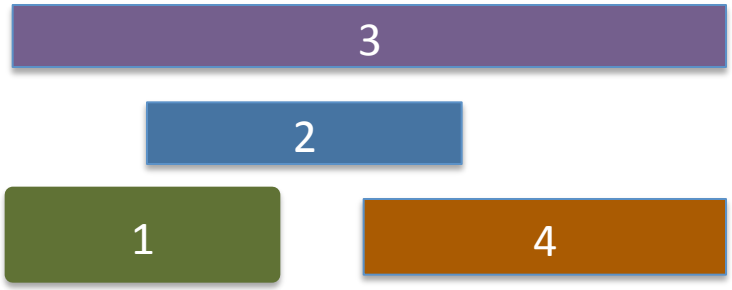
Today's agenda

Prove the lemma

Analyze run-time of the greedy algorithm

Algorithm implementation

Go through the intervals in order of their finish time



Check if s[i] < f(1)

In general, if jth interval is the last one chosen

Pick smallest i>j such that s[i] ≥ f(j)

O(n log n) run time

The final algo

O(n log n) time sort intervals such that $f(i) \leq f(i+1)$

O(n) time build array $s[1..n]$ s.t. $s[i]$ = start time for i

Add 1 to A and set $f = f(1)$

For $i = 2 .. n$

If $s[i] \geq f$

Add i to A

Set $f = f(i)$

Return $A^* = A$

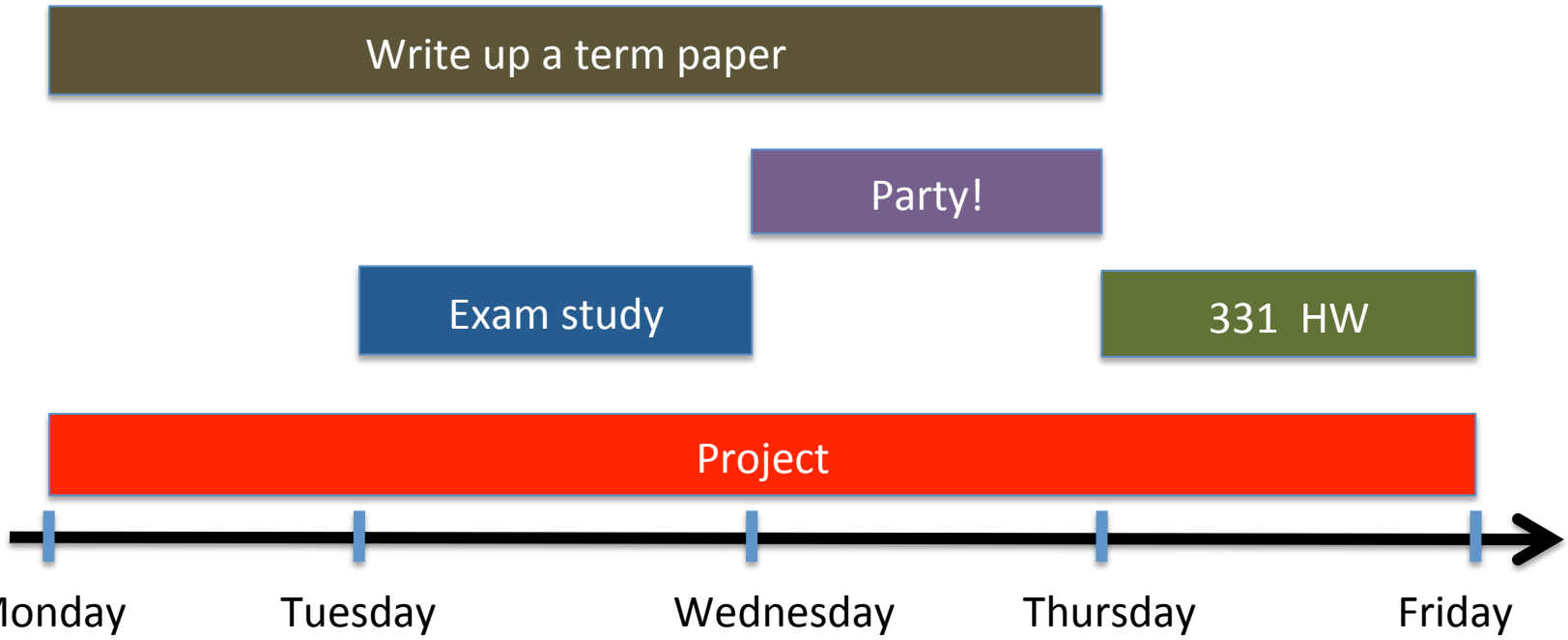
Reading Assignment

Sec 4.1of [KT]

Questions?

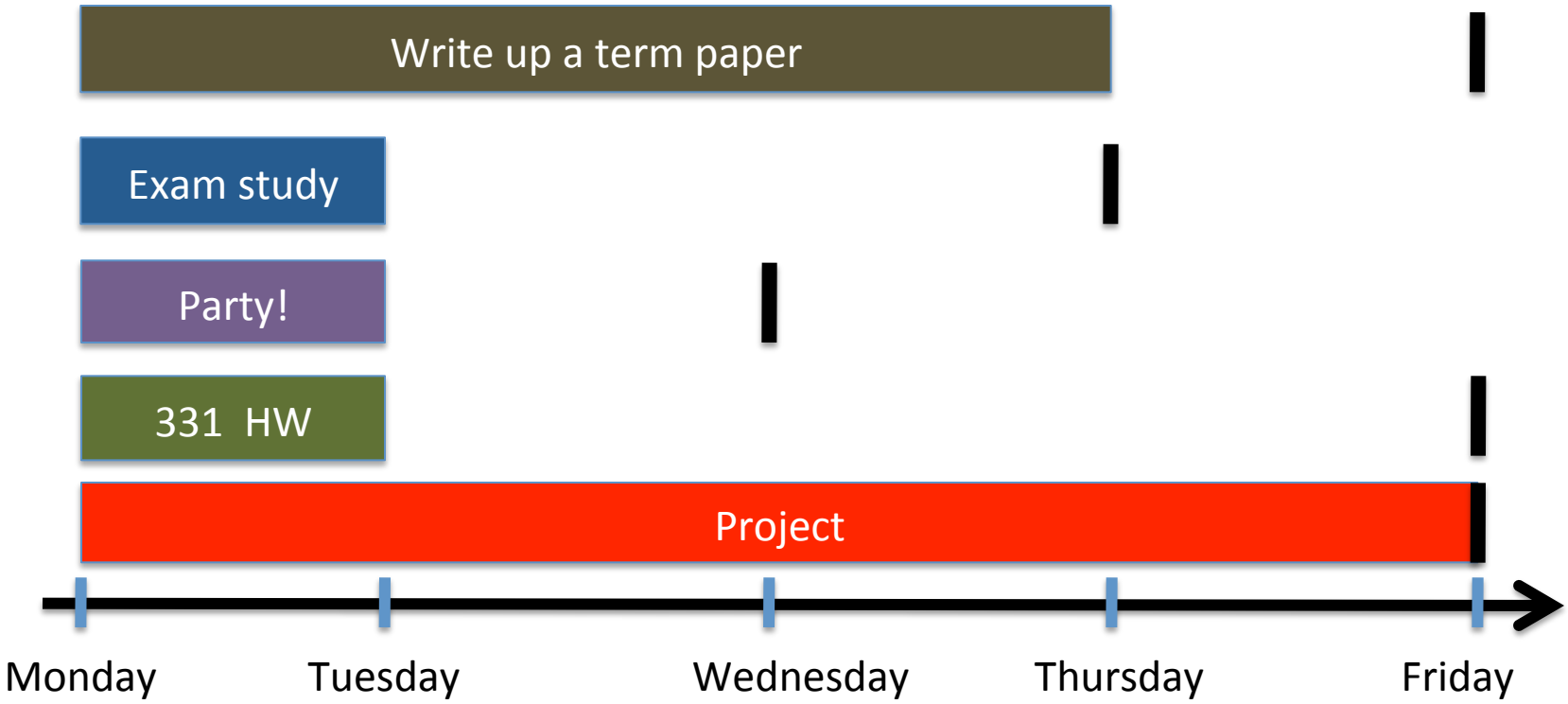
The "real" end of Semester blues

There are deadlines and durations of tasks



The "real" end of Semester blues

There are deadlines and durations of tasks



The algorithmic task

YOU decide when to start each task

Write up a term paper

Exam study

Party!

331 HW

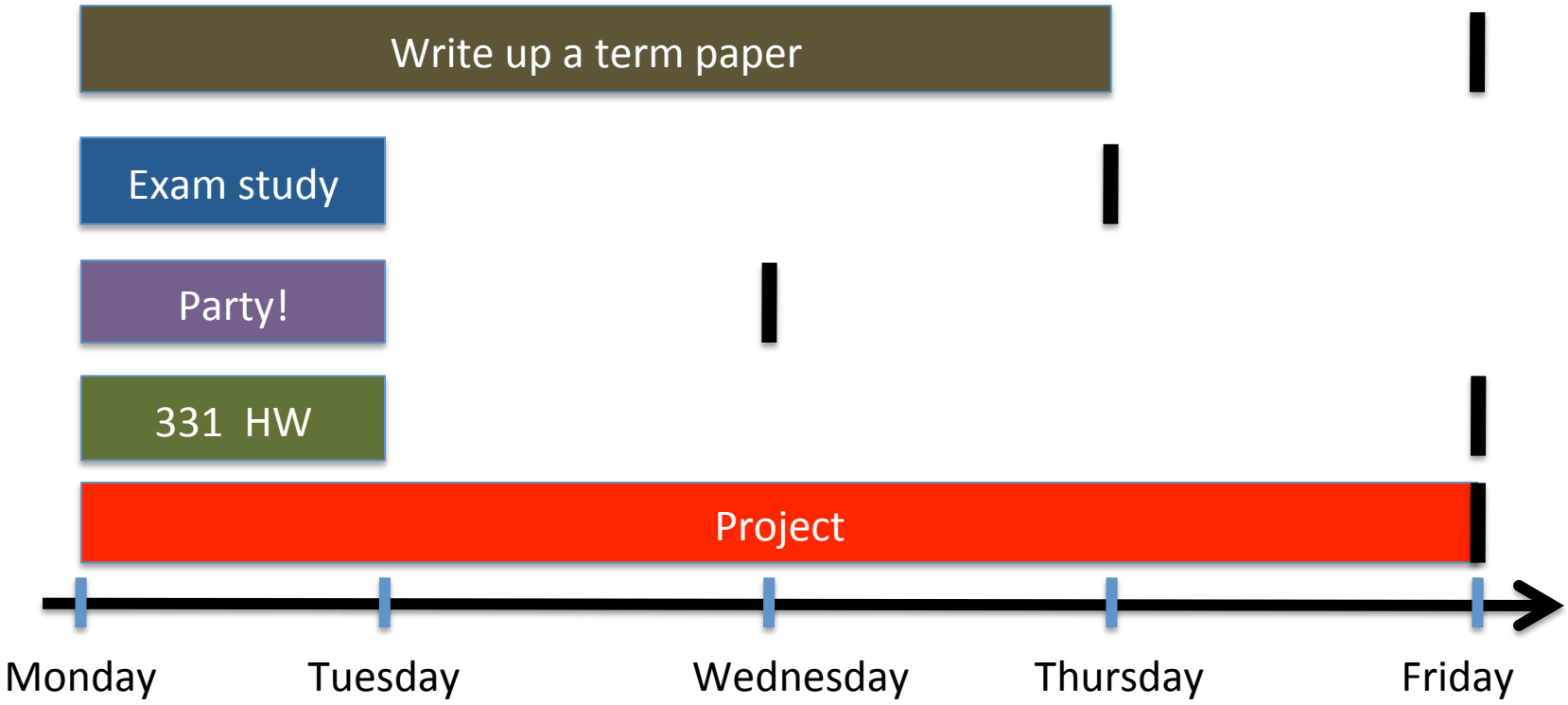
Project

You have to do ALL the tasks

Monday Tuesday Wednesday Thursday Friday

Scheduling to minimize lateness

All the tasks have to be scheduled
GOAL: minimize maximum lateness



One possible schedule

All the tasks have to be scheduled
GOAL: minimize maximum lateness