

# Lecture 15




CSE 331

Oct 3, 2022

If you need it, ask for help



# Project groups

note @220   

stop following 12 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 after 6:30pm on Thursday

Like in [@206](#) the email will of the following format:

- Be on the lookout for an 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 its 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 :)

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

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

[project](#)

[Edit](#) good note 0 Updated 10 seconds ago by Kim Rudez

# Quiz 1 this FRIDAY

note @183

stop following 78 views

Actions

## Quiz 1 on Friday, Oct 7

The first quiz will be from 11:00-11:10am in class on Friday, October 7. 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 10:55am 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 sample mid term 1: @182.) Also quiz 1 will cover all topics we cover in class till Friday, Sep 30.

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 days ago by Atri Rudra

# Mid-term post 1



note @192



stop following

42 views

Actions

## The mid-term post

First, midterm-I is on **Monday, Oct 10** and midterm-II is on **Wednesday, Oct 12** during the usual class timings (i.e. 11:00-11:50am in Hoch 114). Below are some comments that might be helpful to prepare for the mid-term.

(Thoughts on what to do during the exam here: [@193](#))

- Work through the sample mid-term exams ([@182](#)). 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 7 ([@183](#)) to get some practice in solving T/F questions under some time pressure. Also review the T/F polls ([@81](#)) for more examples of such T/F questions.
- Review the HW problems/solutions. HW solutions are here: [@140](#).
- 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.
- 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

# Mid-term post 2

note #193

stop following

74 views

Actions




## Few thoughts on what to do during the exam

In a previous post [#182](#), I listed some pointers on how I think you should prepare for the mid-term exams.

Below are (in no particular order) some thoughts on how you should work on the actual exam:

- 1. Do NOT panic (or delay it as much as possible)** And I don't mean this in either a joking way or a scary way. In these kinds of exams once you panic everything else that follows will not be good. (Believe me I have been there.) So the idea for you will be to avoid panicking as much as possible or mitigate its effects. Here are some specific pointers in this regard.
  - Read **all** the questions even before you start writing anything. This way if you are short on time and you are not done at least you will be working on a question that you have read before: trying to make sense of a question that you are reading for the first time and under time pressure never ends well.
  - You know the structure and number of questions. Make sure you setup a time table on how much time you want to spend on each questions and stick to that plan. Make sure you keep at least 10 mins at the end to go over all your answers to make sure you were not missing something.
    - Make sure you stick to your timetable and avoid the sunk cost fallacy. Thinking that I have already spent 5 mins on a question so let me spend a couple more mins to try and crack the question often leads to you spending 15 mins on the question and then you are terribly short on time.
  - I try to order the questions from easiest to hardest and I think I do fine on the average but the ordering might not match with yours. E.g. for some reason you might have studied a particular part of the book the night before the exam and that part might be relevant to say the last question. So what I think might a hard question for an average student in the class might be easy for you. Reading through all questions upfront will also help you identify these "out of order" questions.
- 2. Try to reinvent as little of the wheel as possible.**
  - Your first attack on any problem should be to see if you can sufficiently modify the question/input to the algorithm so that you can use a solution from a previous HW problem/the book/stuff on piazza as a blackbox. Note this is the same philosophy as to why you should libraries instead of writing code from scratch.
    - Remember how easy it was to get most points on 1(a) and 2(a) by just referring to the recitation notes. Y'all should try to do that as much as possible!
    - If you try **and build something from scratch (like an algorithm or a proof) that you could have just referenced away, then you will be short on time.** The exam is timed in such a way that if something can be just referenced, then you are expected to do so.
  - If the above fails then you should see if you can slightly tweak a previous solution to work in the current situation. Most of the problems in the mid terms will either be in the previous category or this one.
  - If both of the above fail, then try to answer from scratch but **this should be your last option.**
- 3. Use the cheatsheet well.** Think about what parts of 331 you struggle with and write those down on the cheatsheet. This e.g. could be runtimes of various algorithms or maybe outline of an algorithm etc. You should not waste real estate on things you can recollect immediately rather you should use it as an aid to remember things immediately. As another example, if for some reason you cannot never remember how to solve a particular problem or T/F poll etc: just write that down. Or if you have trouble remembering precise references to existing algorithms/proofs, then write those down on the cheatsheet. On the other hand do not spend a huge amount of time writing up a cheatsheet (at the expense of reviewing the material itself).
- 4. If all fails, remember this is only an exam!** I got a C in undergrad algorithms and I would like to think I did just fine. Also remember that if you do better on the final exam, I will completely drop your mid-term score, so you should treat the mid-term as a no pressure exam.

# Feedback on CSE 331

note #321    stop following 1 view Actions

## Feedback on CSE 331

Every year, I ask y'all to give feedback on CSE 331, so here is the feedback form for this year:

[https://docs.google.com/forms/d/e/1FAIpQLSdF7WvQ3P4d0F17\\_eOx6s7AmpcDASdZs-GuJydnxbelV5aVe/viewform?usp=af\\_link](https://docs.google.com/forms/d/e/1FAIpQLSdF7WvQ3P4d0F17_eOx6s7AmpcDASdZs-GuJydnxbelV5aVe/viewform?usp=af_link)

Filling in this form is **completely optional and anonymous**.

In particular, I would love feedback (even if it is critical). Many of the aspects of CSE 331 that you (might) like were suggested by someone in a previous incarnation of CSE 331. While I'll try and incorporate as much as I can this fall, some of your suggestions might have to wait for the next offering.

I might also disagree with your feedback but after a week or so, I'll post my response to the feedback from y'all. So at the very least y'all would get to hear my reasoning for why certain things are the way they are in CSE 331. And then we can agree to disagree :-)

[Feedback](#)

Full good note 

Updated 51 seconds ago by All Pudge

# Recitation this week

note @224 🌐 ★ 🔒 23 views Actions

## Recitations This Week

Hey everyone,

Recitations this upcoming week (on 10/5 and 10/6) will be review for the quiz/midterms since you will not have a homework being released.

**Please come with questions. If you do not ask questions, we will likely go over the practice midterms (which you could easily do on your own).**

If there are topics that many people want covered, feel free to post them as a follow up to this note. Responses that have a lot of "likes" (upvotes, etc.) will be a priority for us to cover if no one has questions.

recitation

Edit good note | 0

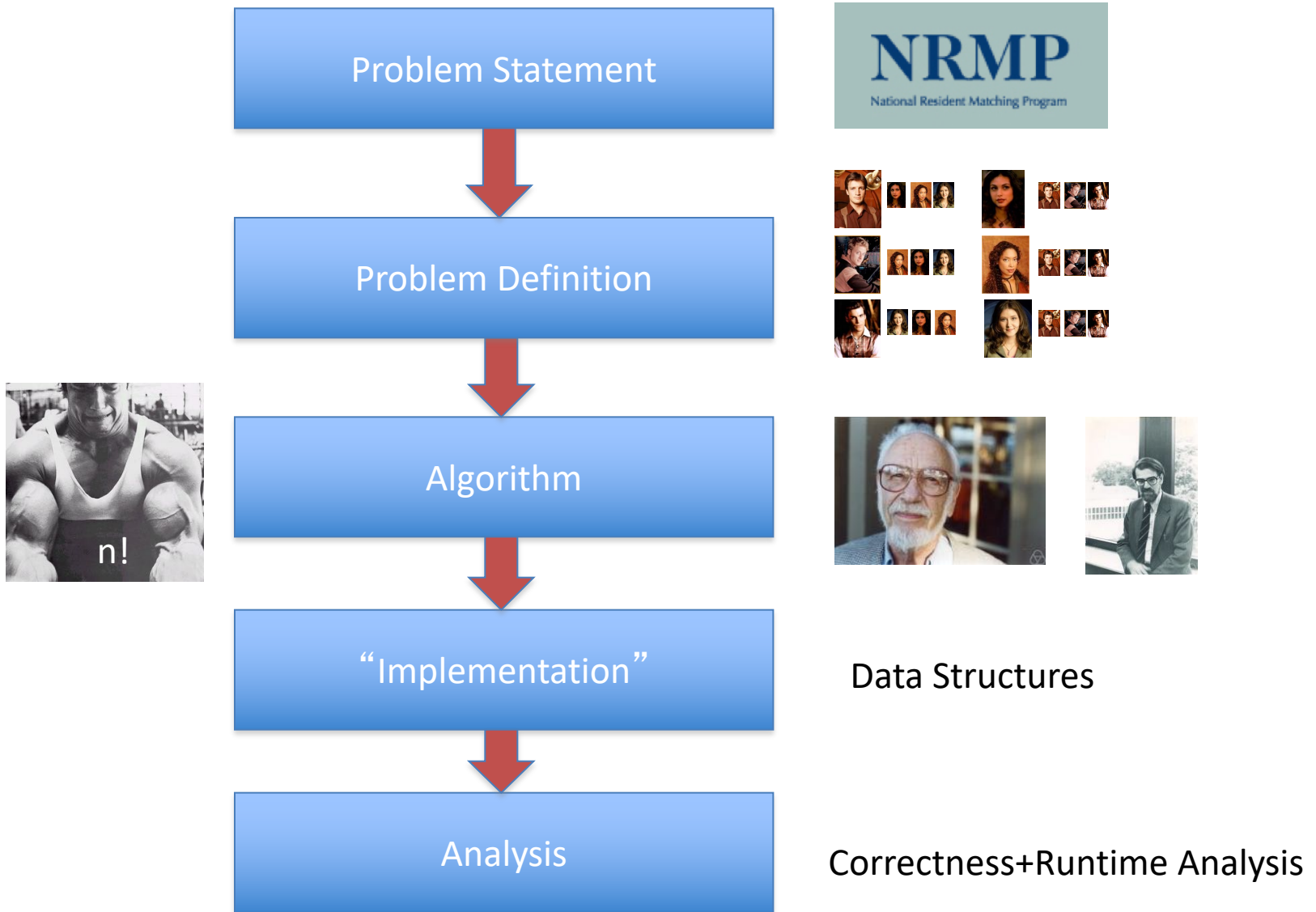
Updated 3 hours ago by Trevor Schneggenburger



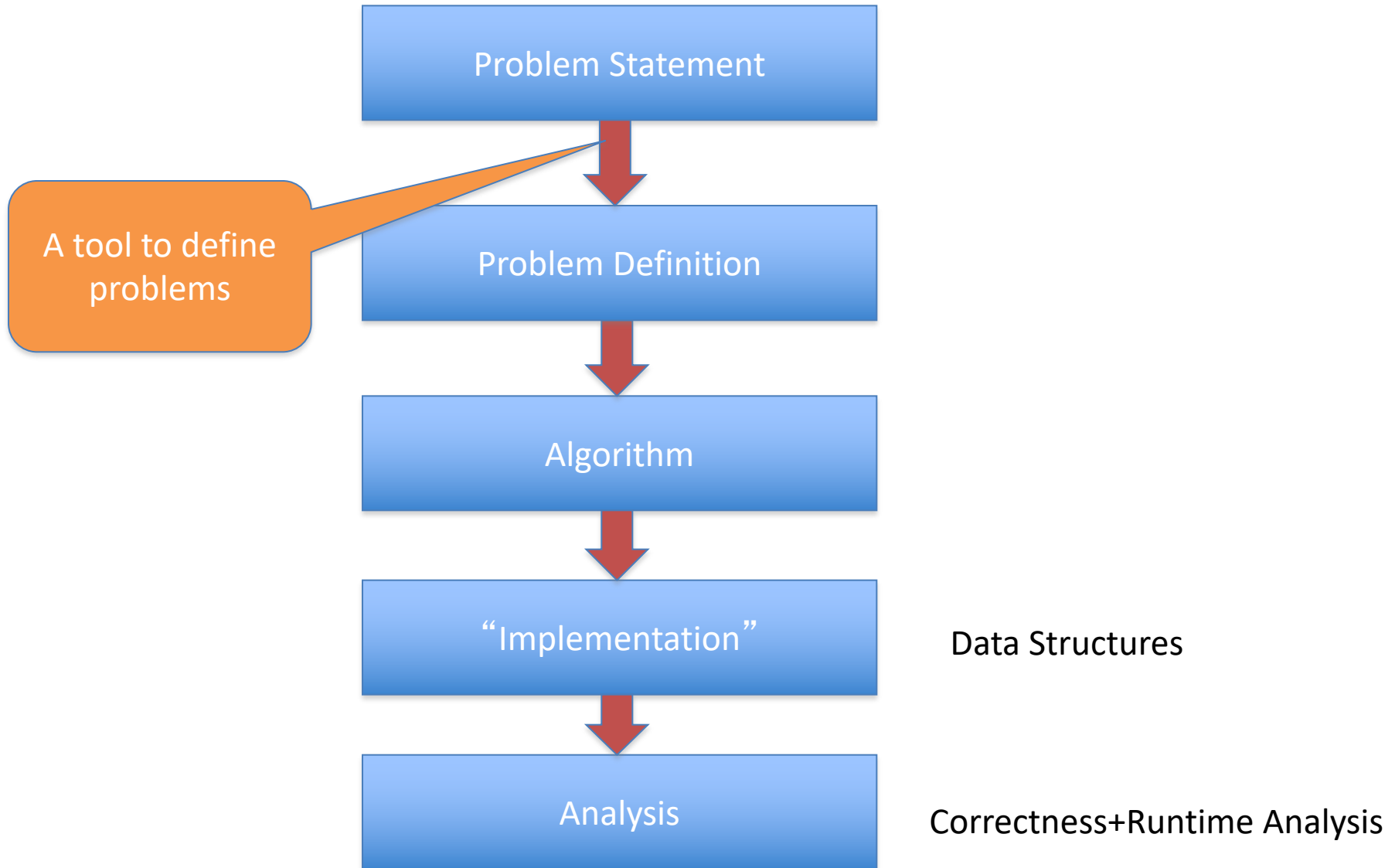
# Questions?



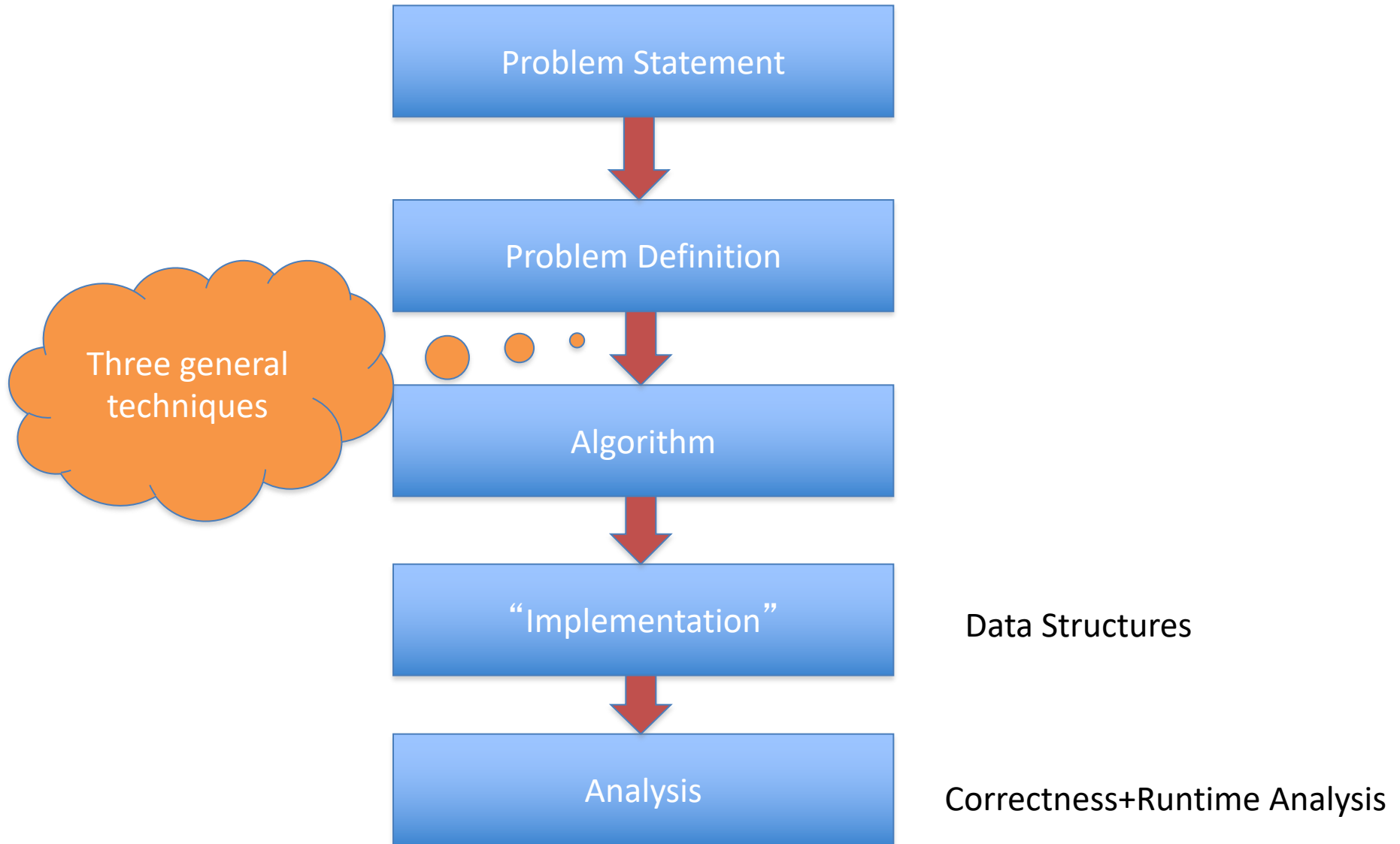
# Main Steps in Algorithm Design



# 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?



Write up a term paper

Party!

Exam study

Homework

331 HW

Project

Friday

Saturday

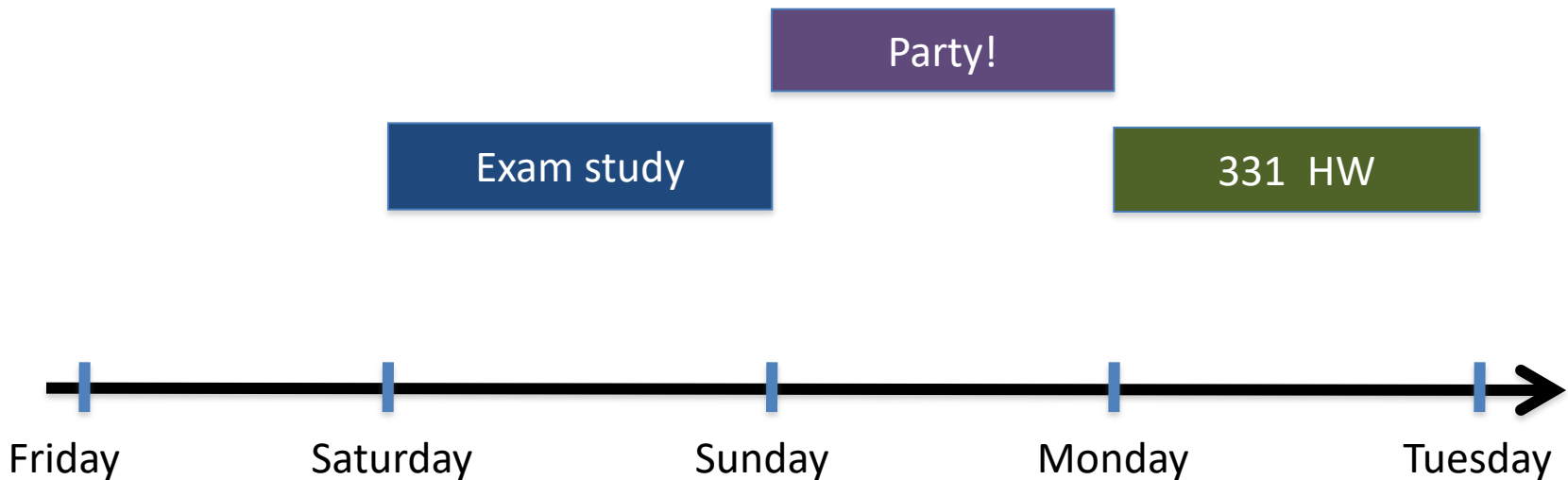
Sunday

Monday

Tuesday

# 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  $1 \leq i \leq n$



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

**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,  $\{s(i), \dots, f(i) - 1\}$  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:**

# Re-define problem on the board...



# Example 1

No intervals overlap



# Algorithm?



No intervals overlap

$R$ : set of requests

Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$

    Add  $i$  to  $S$

    Remove  $i$  from  $R$

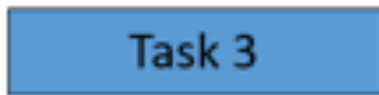
Return  $S^* = S$

# Questions/Comments?

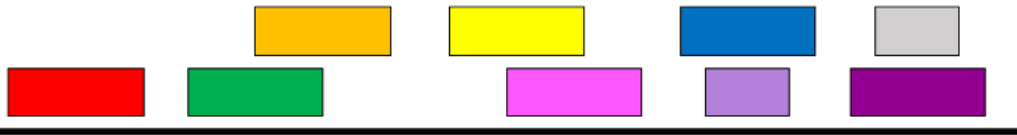


# Example 2

At most one overlap/task



# Algorithm?



At most one overlap

$R$ : set of requests

Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$

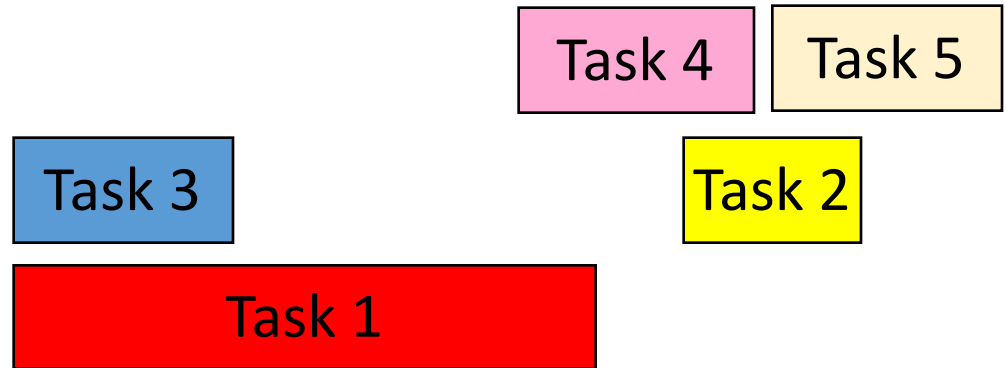
    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

# Example 3

More than one conflict



Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$



# Greedily solve your blues!

Arrange tasks in some order and iteratively pick non-overlapping tasks



Write up a term paper

Party!

Exam study

331 HW

Project

Saturday

Sunday

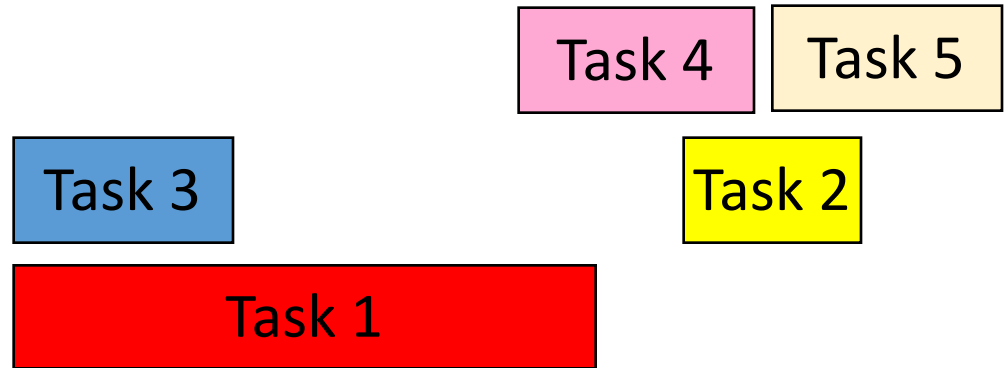
Monday

Tuesday

Wednesday

# Making it more formal

More than one conflict



Set  $S$  to be the empty set

While  $R$  is not empty

**Choose**  $i$  **in**  $R$  that minimizes  $v(i)$

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

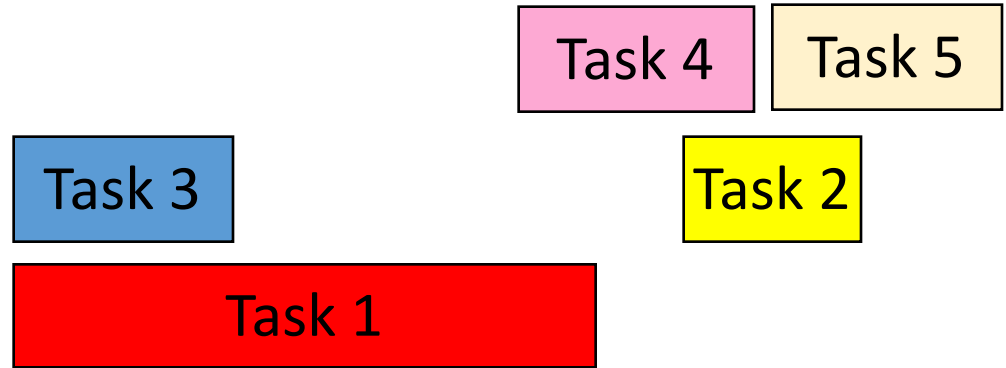
Associate a  
value  $v(i)$   
with task  $i$

# Questions/Comments?



# What is a good choice for $v(i)$ ?

More than one conflict



Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$  that minimizes  $v(i)$

    Add  $i$  to  $S$

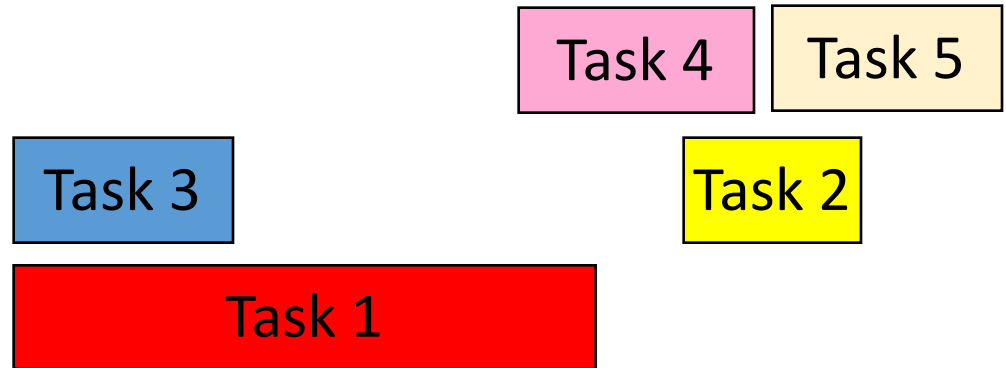
    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

Associate a  
value  $v(i)$   
with task  $i$

$$v(i) = f(i) - s(i)$$

Smallest duration first



Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$  that minimizes  $f(i) - s(i)$

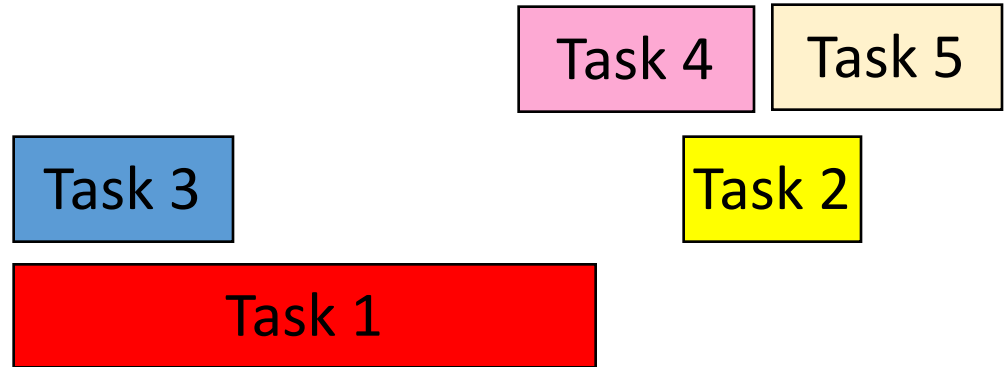
    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

$$v(i) = s(i)$$

Earliest time first?



Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$  that minimizes  $s(i)$

    Add  $i$  to  $S$

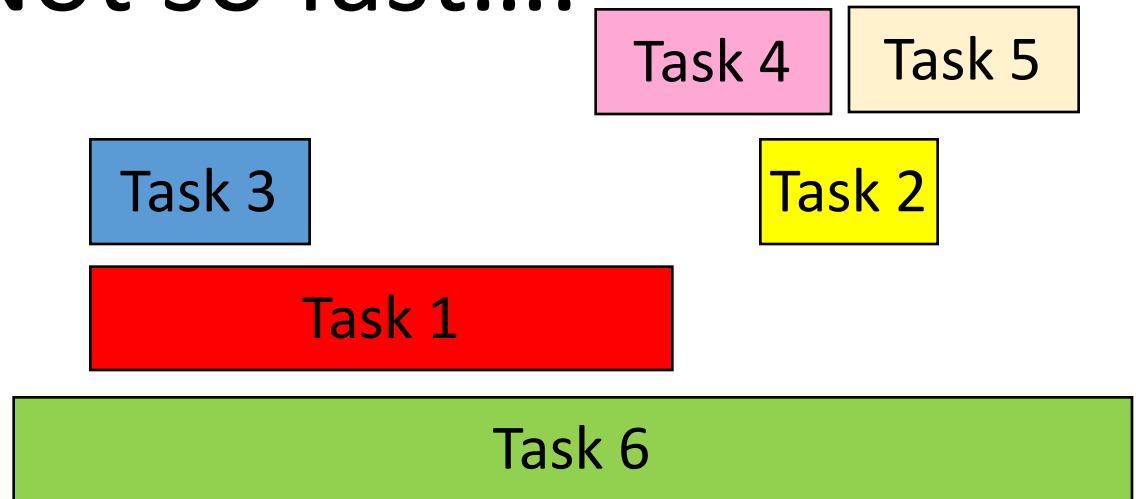
    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

So are we  
done?

# Not so fast....

Earliest time first?



Set  $S$  to be the empty set

While  $R$  is not empty

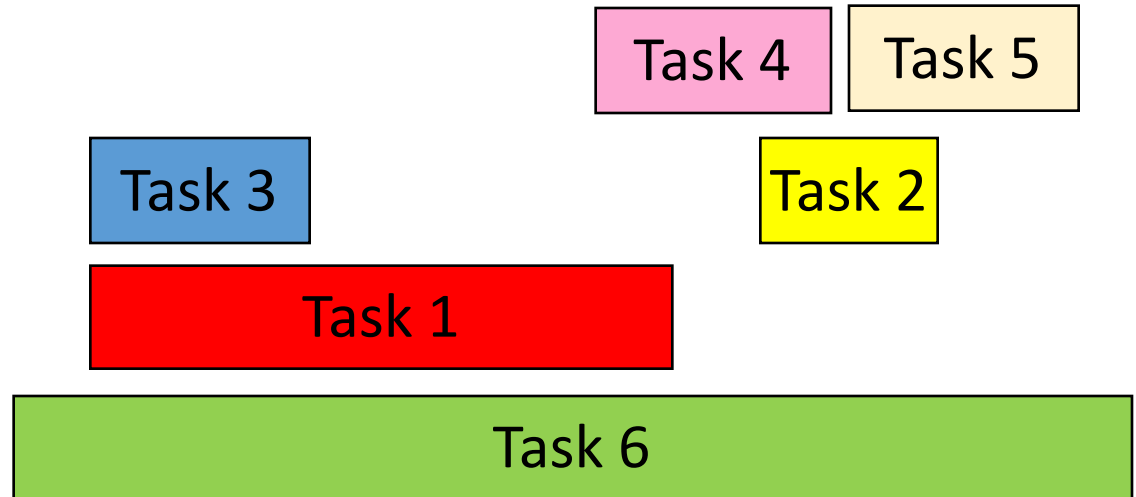
    Choose  $i$  in  $R$  that minimizes  $s(i)$

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

# Pick job with minimum conflicts



Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$  that has smallest number of conflicts

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

So are we  
done?



# Questions/Comments?



# Nope (but harder to show)

Set  $S$  to be the empty set

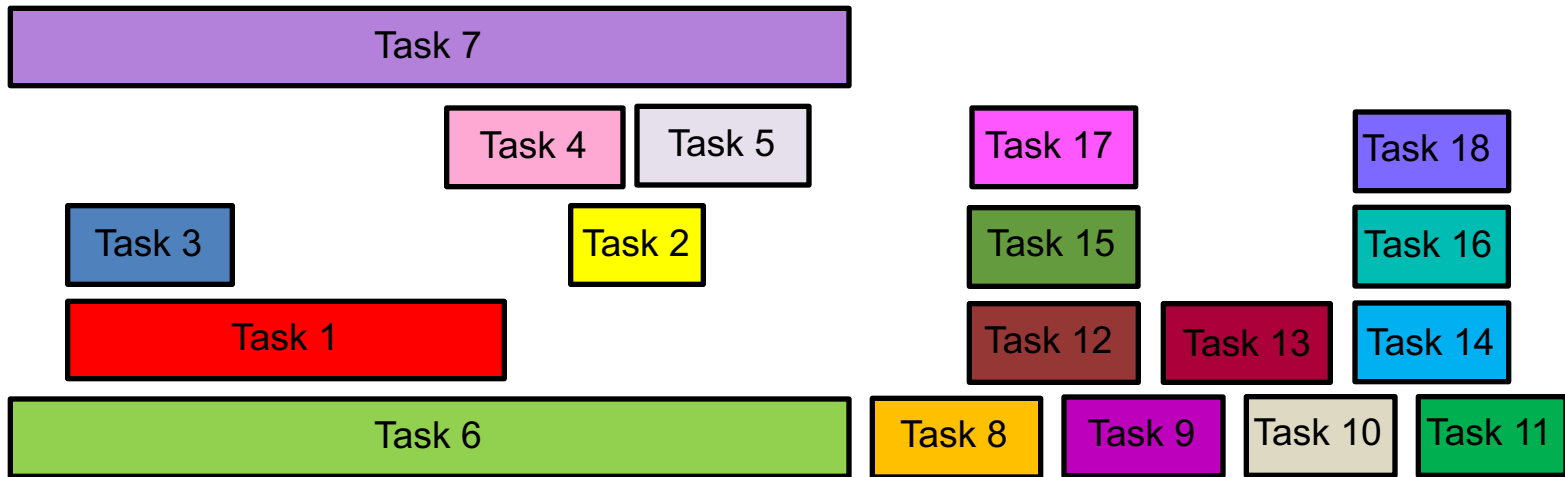
While  $R$  is not empty

    Choose  $i$  in  $R$  that has smallest number of conflicts

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$



Set  $S$  to be the empty set

While  $R$  is not empty

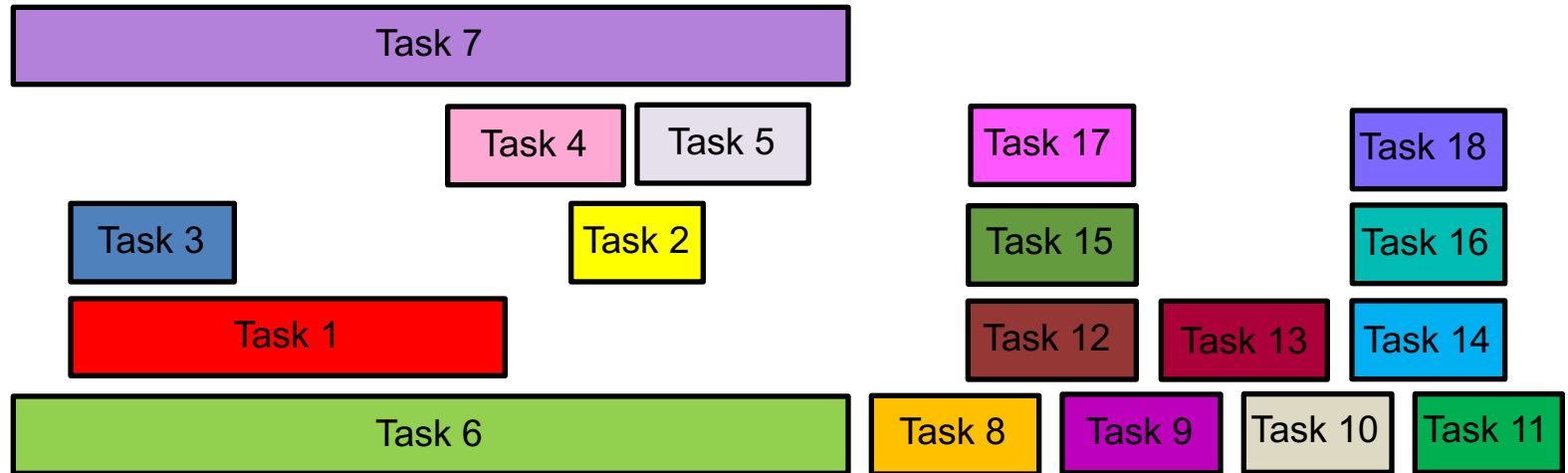
    Choose  $i$  in  $R$  that has smallest number of conflicts

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

# Algorithm?



Set  $S$  to be the empty set

While  $R$  is not empty

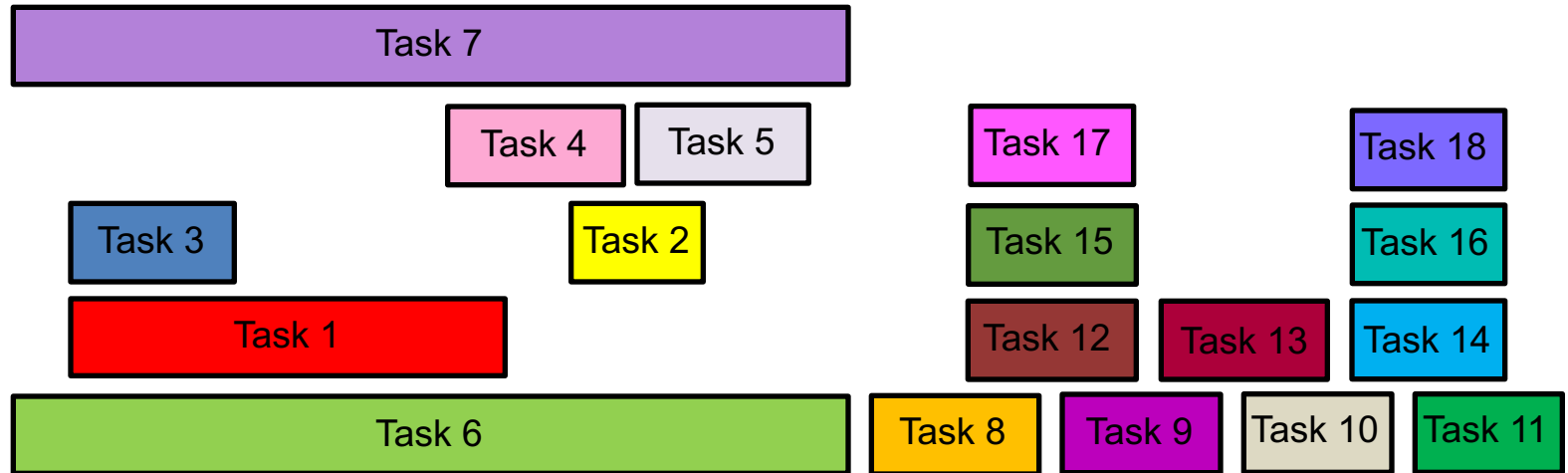
    Choose  $i$  in  $R$  that minimizes  $v(i)$

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

# Earliest finish time first



Set  $S$  to be the empty set

While  $R$  is not empty

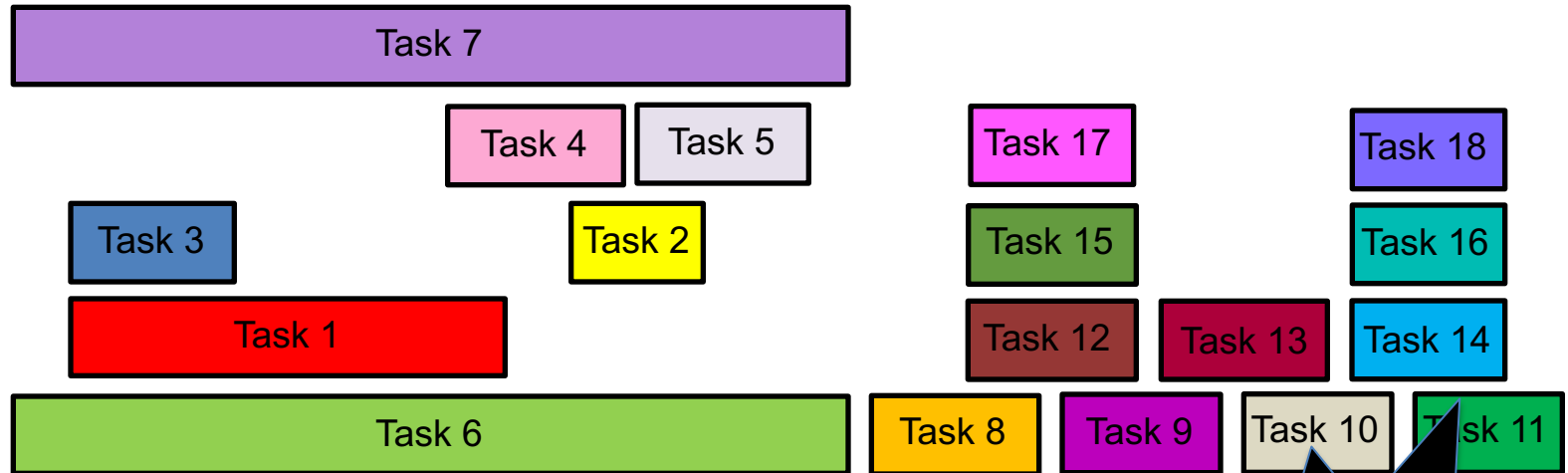
    Choose  $i$  in  $R$  that minimizes  $f(i)$

    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

# Find a counter-example?



Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$  that minimizes  $f(i)$

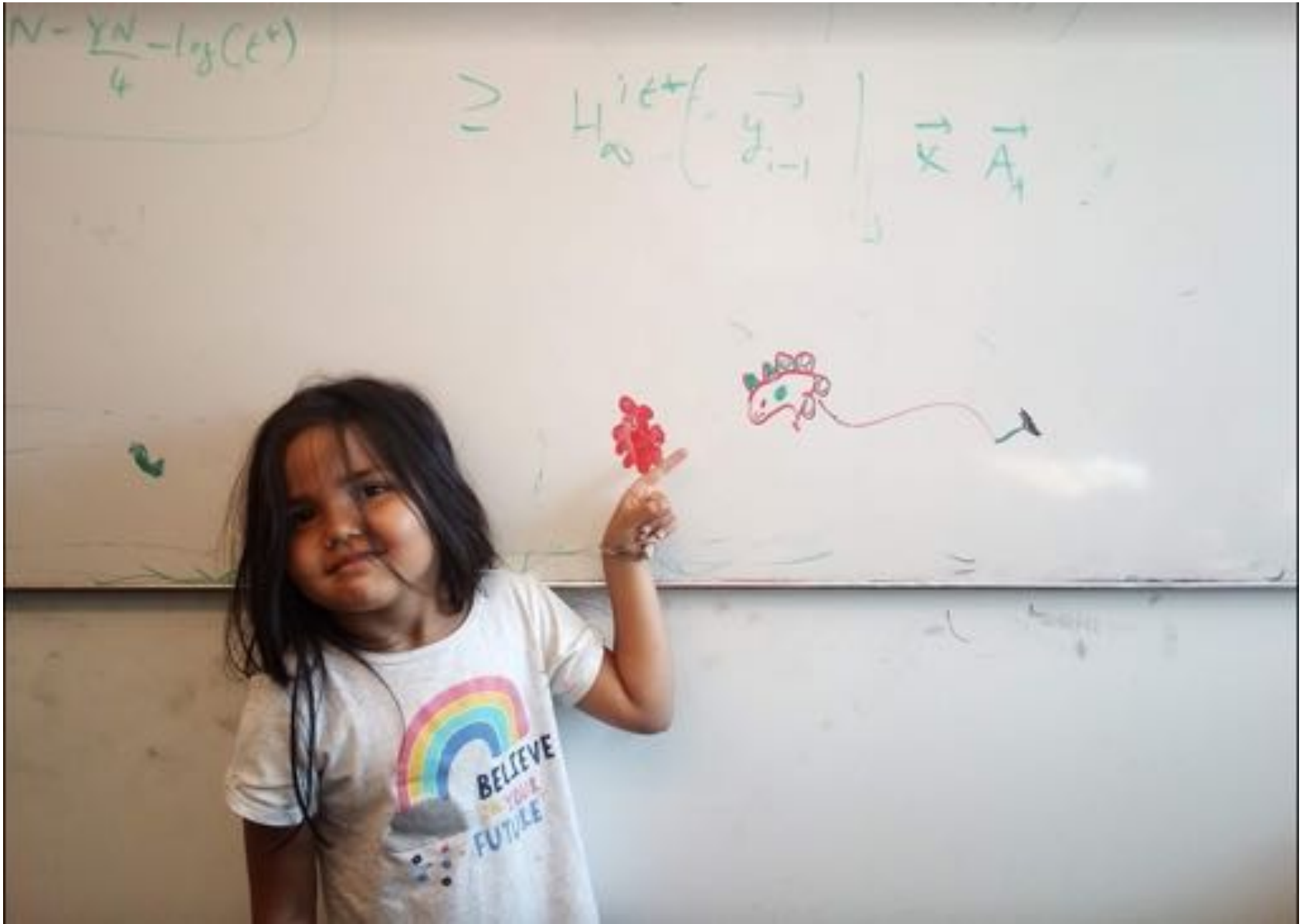
    Add  $i$  to  $S$

    Remove all tasks that conflict with  $i$  from  $R$

Return  $S^* = S$

It  
works!

# Questions/Comments?



# Today's agenda

Prove the correctness of the algorithm



# Final Algorithm

$R$ : set of requests

Set  $S$  to be the empty set

While  $R$  is not empty

    Choose  $i$  in  $R$  with the earliest finish time

    Add  $i$  to  $S$

    Remove all requests that conflict with  $i$  from  $R$

Return  $S^* = S$

# Argue correctness on the board...

