Greedy scheduling

Lecture 05.02

SAMPLE PROBLEM 4

Movie star scheduling

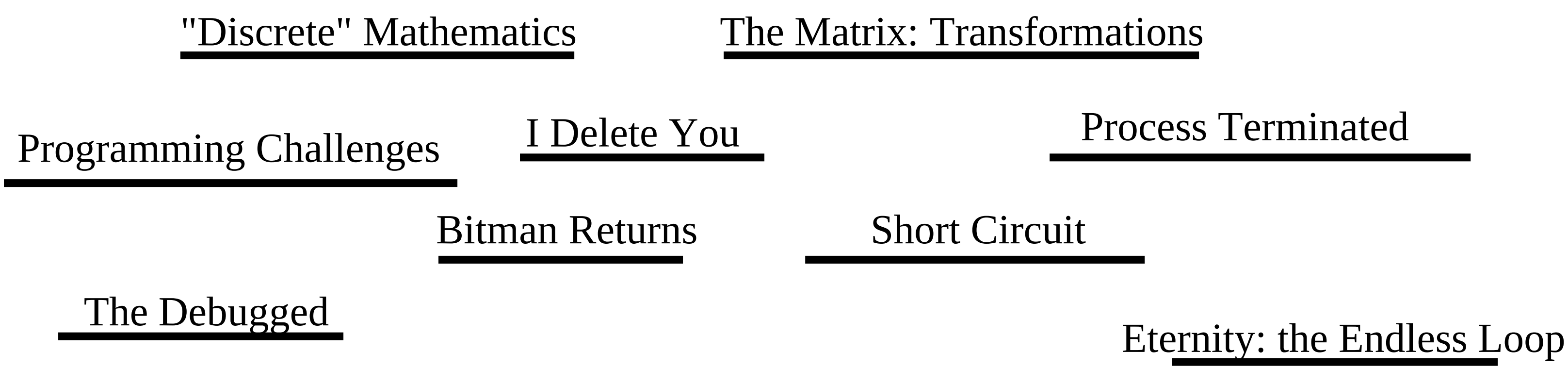
Movie Star Scheduling Problem

A movie star has been offered the leading role in several upcoming movies

They want to select the **maximum number of roles** such that no movies overlap in time

We call the movies that overlap *conflicting movies*

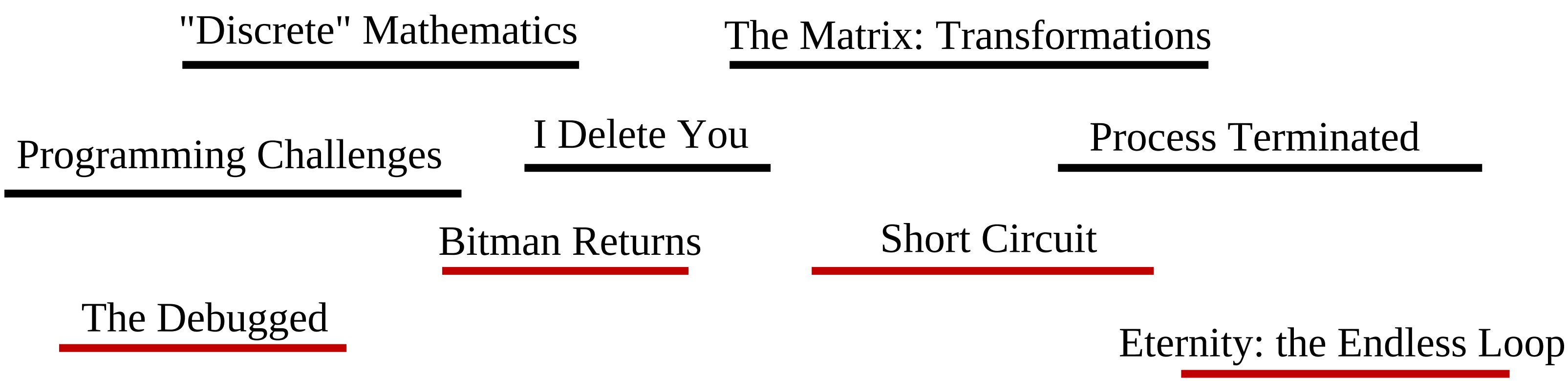
Movies with their start and end times shown as an interval of time:



What is an optimal solution for this problem instance?

Movie Star Scheduling Problem

Movies with their start and end times shown as an interval of time:



Optimal solution: 4 jobs

Movie Star Scheduling Problem

- This problem is known as *maximum independent set* in an *interval graph*.
- Each interval has a start and end value.

Movie Scheduling Problem	
Input:	A set I of n intervals on the line.
Output:	*T*he largest subset of non-conflicting intervals which can be selected from I

How long will exhaustive computation take?

Movie Star Scheduling Problem

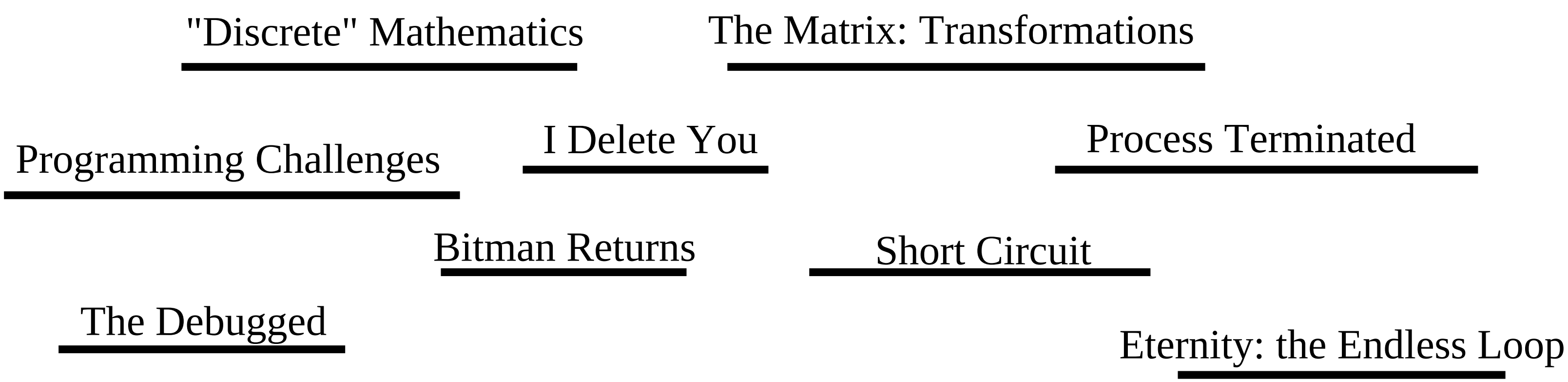
- This problem is known as *maximum independent set* in an *interval graph*.
- Each interval has a start and end value.

Movie Scheduling Problem	
Input:	A set *I* of *n* intervals on the line.
Output:	*T*he largest subset of non-conflicting intervals which can be selected from *I*

2^n different interval subsets: similar to knapsack 01 – the interval is either selected or not

Greedy move: version 1

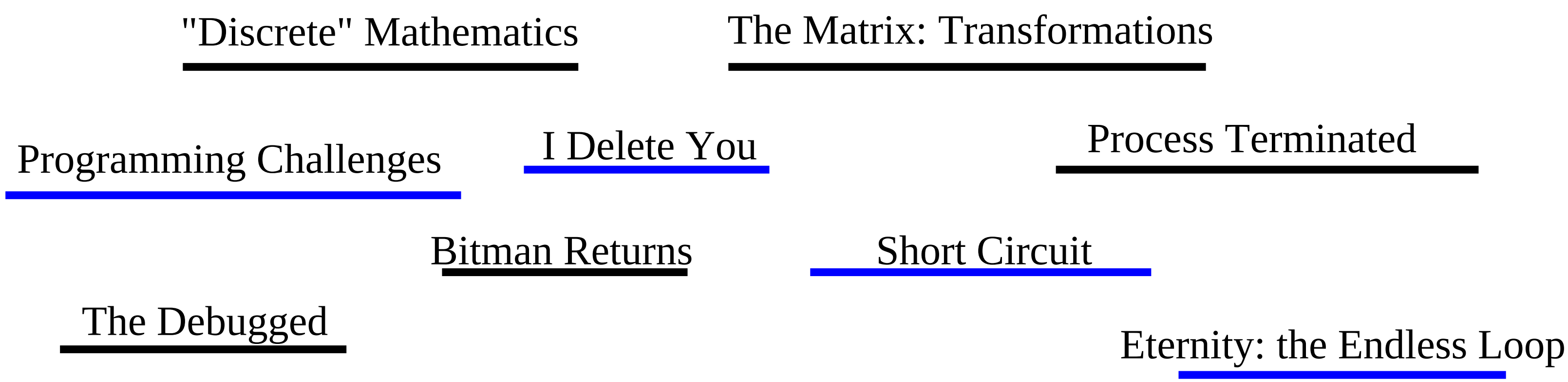
*1.**Starting-First*. Accept the job that starts soonest and doesn't conflict.



What jobs will be selected by the starting-first algorithm?

Greedy move: version 1

1. *Starting-First.* Accept the job that starts soonest and doesn't conflict.



What jobs will be selected by the starting-first algorithm?

Is "starting-first" a safe move?

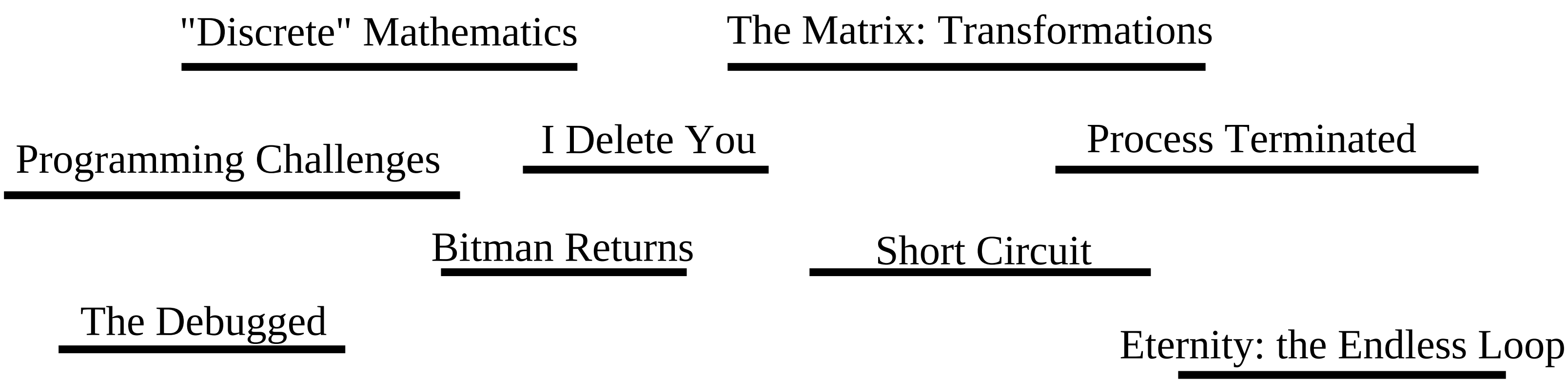
Can we miss an optimal solution?

Counterexample for Starting-First

Starting-first is not a safe move ∎

Greedy move: version 2

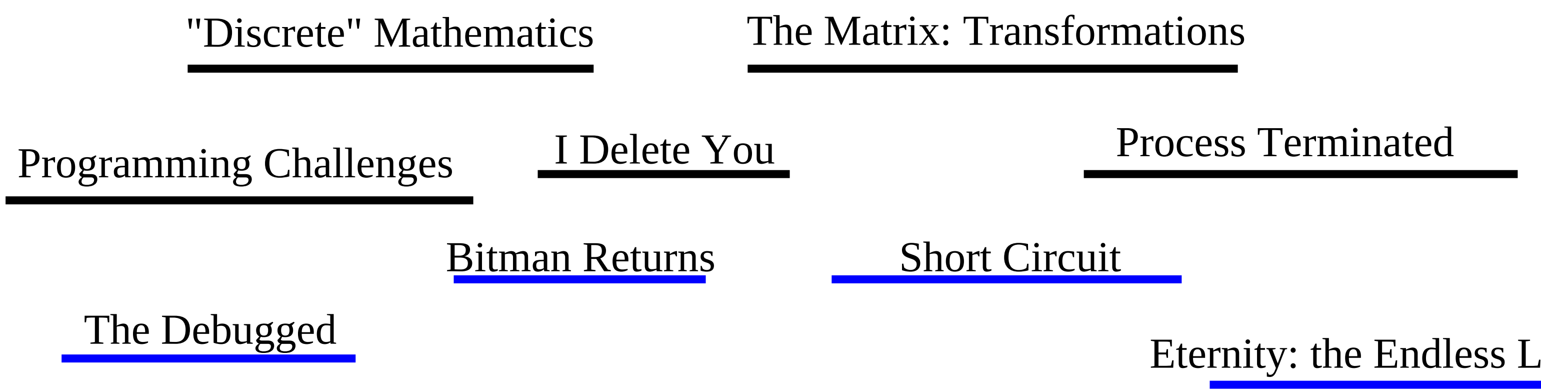
2. *Shortest-First*. Accept the shortest job that doesn't conflict.



What jobs will be selected by the shortest-first algorithm?

Greedy move: version 2

2. Shortest-First. Accept the shortest job that doesn't conflict.



What jobs will be selected by the shortest-first algorithm?

Is "shortest-first" a safe move?

Can we miss an optimal solution?

Counterexample for shortest-first

Shortest-first is not a safe move ∎

Movie Star Problem

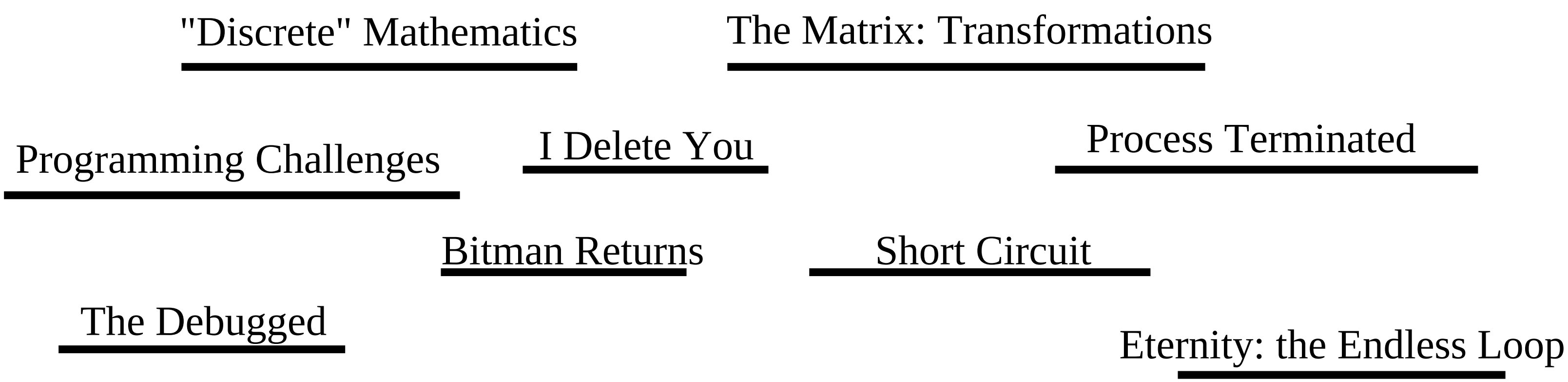
Input:	A set *I* of *n* intervals on the line.
Output:	*T*he largest subset of conflicting intervals which can be selected from *I*

Maybe greedy approach does not work here?

Maybe we need to do an exhaustive search?
Or use another strategy?

Greedy move: version 3

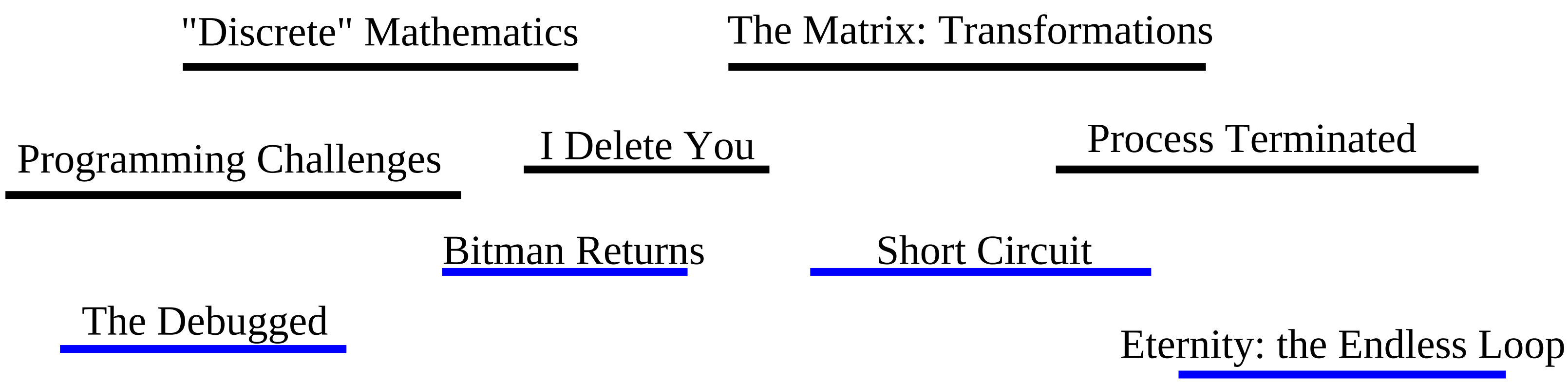
3. *Ending*-First. Accept the job that ends soonest and doesn't conflict



What jobs will be selected by the ending-first algorithm?

Greedy move: version 3

3. Ending-First. Accept the job that ends soonest and doesn't conflict



What jobs will be selected by the ending-first algorithm?

Is "ending-first" a safe move?

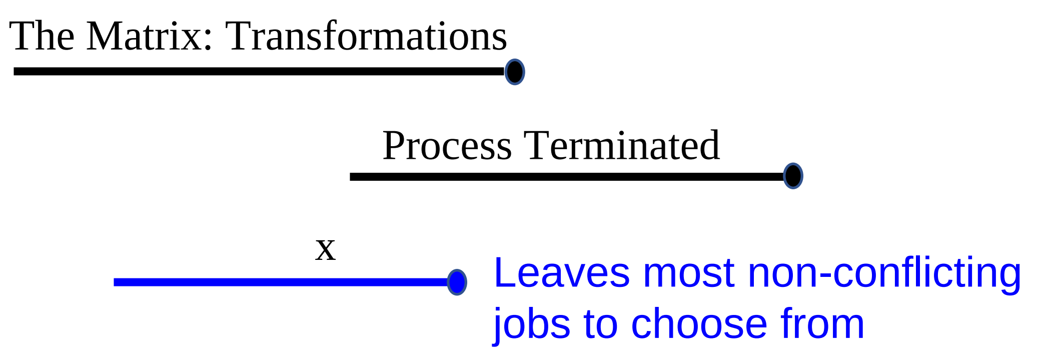
Can we miss an optimal solution?

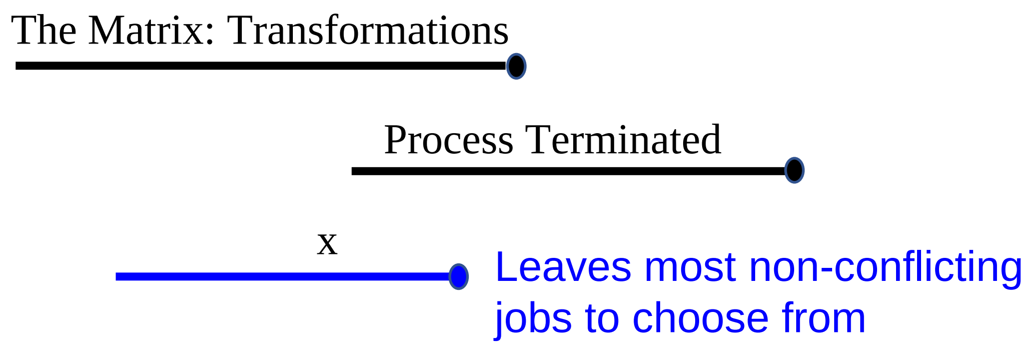
Theorem

Ending-first is a safe move

Proof

- Let *x* be a job which contains the end point which is leftmost among all remaining intervals.
- Other jobs may well have started before *x*, but all of these must conflict with *x*, so we can select at most one from the group.
- The first of these jobs to terminate is *x*, so any of the other conflicting jobs potentially block out more opportunities to the right of it.
- Clearly we can never lose by picking *x*. ∎





- Note that the proof of this theorem did not use an exchange argument
- Instead, we use a *lower-bound* argument: we argue that any solution will be no better without this greedy choice
- Our greedy choice achieves the lower bound of all possible choices

Read more about Greedy Algorithms

10.1 and 10.2 of the textbook

Puzzle: Bridge Crossing at Night

- A group of 4 people with 1 flashlight need to cross a rickety bridge at night.
- A maximum of 2 people can cross the bridge at one time, and any party that crosses (either 1 or 2 people) must have the flashlight with them.
- A pair must walk together at the rate of the slower person's pace.
- The flashlight must be walked back and forth - it cannot be thrown.

The time taken by each person:

Ann takes 1 minute

Bob takes 2 minutes

Cat takes 5 minutes

Don takes 10 minutes

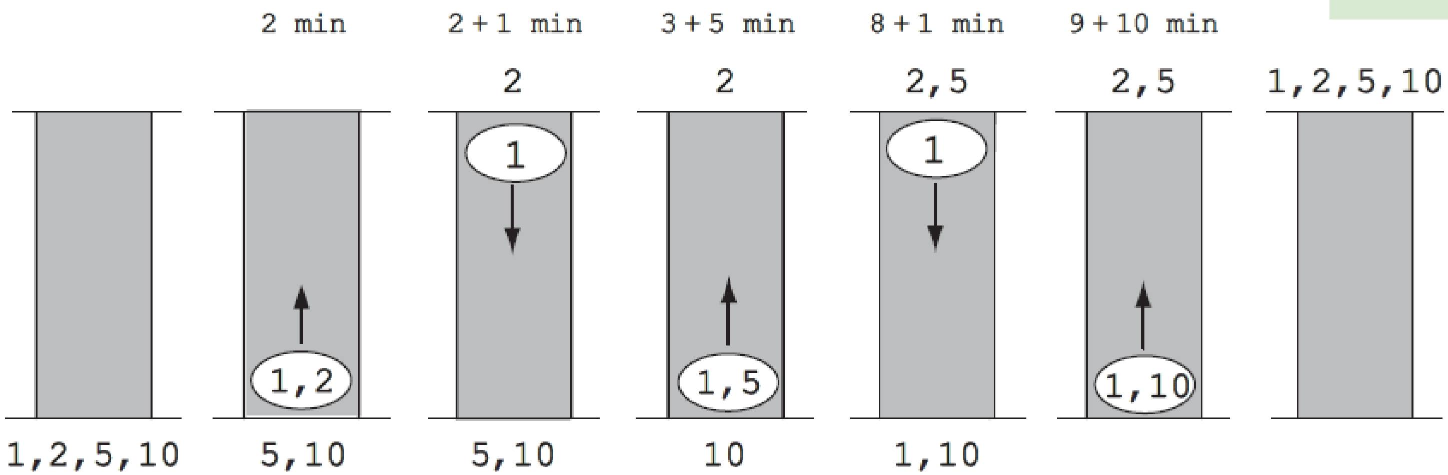
Find the fastest way they all can cross the bridge

Greedy Algorithm

1. Ann takes 1 minute
2. Bob takes 2 minutes
5. Cat takes 5 minutes
10. Don takes 10 minutes

Always send 2 fastest people available, and always send the fastest person back to return the light

19 min in total



Did the algorithm find an optimal solution?

How to reduce the total time?

Play here:

https://www.inwebson.com/demo/cross-the-bridge/

What is the main idea of the optimal solution?
What is a greedy move?