Interval Scheduling

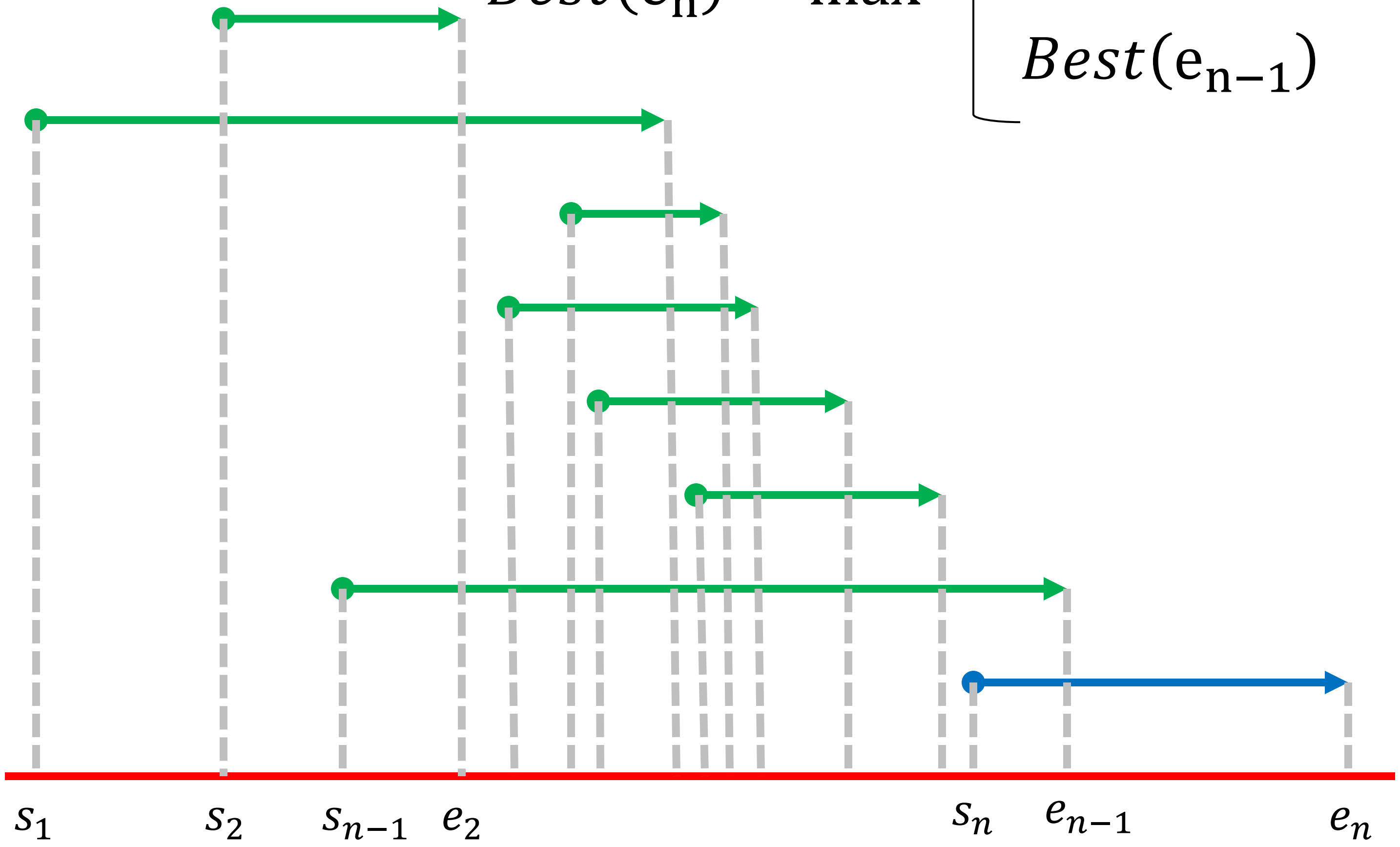
- Input: List of events with their start and end times (sorted by end time)
- Output: largest set of non-conflicting events (start time of each event is after the end time of all preceding events)

[1, 2.25]	Alumni Lunch
[2, 3.25]	CS4102
[3, 4]	CHS Prom
[4, 5.25]	Bingo
[4.5, 6]	SCUBA lessons
[5, 7.5]	Roller Derby Bout
[7.75, 11]	UVA Football watch party

Interval Scheduling DP

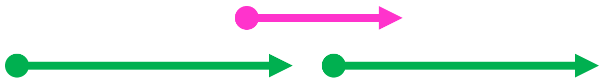
$Best(t)$ = max # events that can be scheduled before time t

$$Best(\mathrm{e_n}) = \max \begin{cases} Best(\mathrm{s_n}) + 1 & \text{Include event } n \\ Best(\mathrm{e_{n-1}}) & \text{Exclude event } n \end{cases}$$

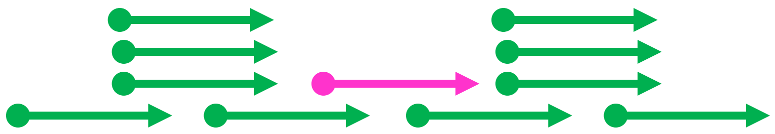


Greedy Interval Scheduling

- Step 1: Identify a greedy choice property
  - Options:
    - Shortest interval



    - Fewest conflicts



    - Earliest start

    - Earliest end

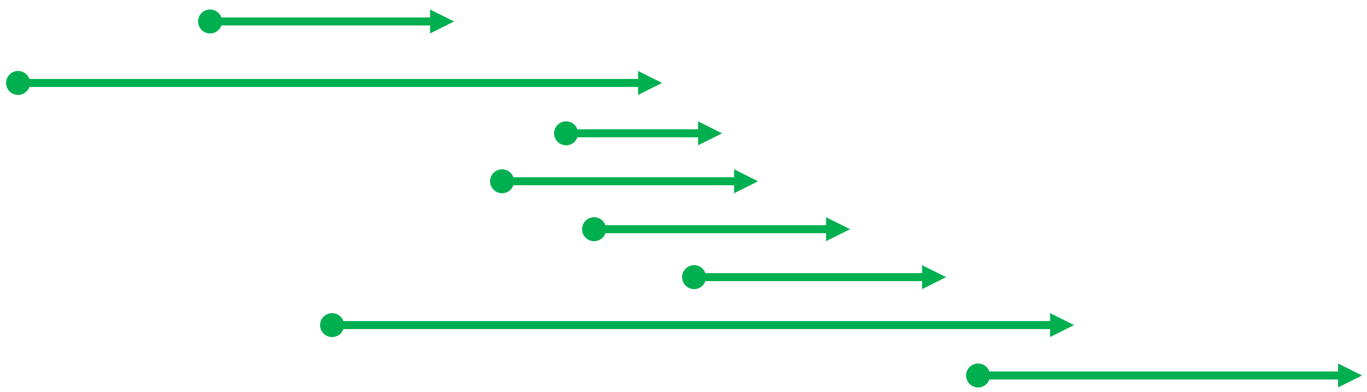
Prove using Exchange Argument

Interval Scheduling Algorithm

Find event ending earliest, add to solution,

Remove it and all conflicting events,

Repeat until all events removed, return solution

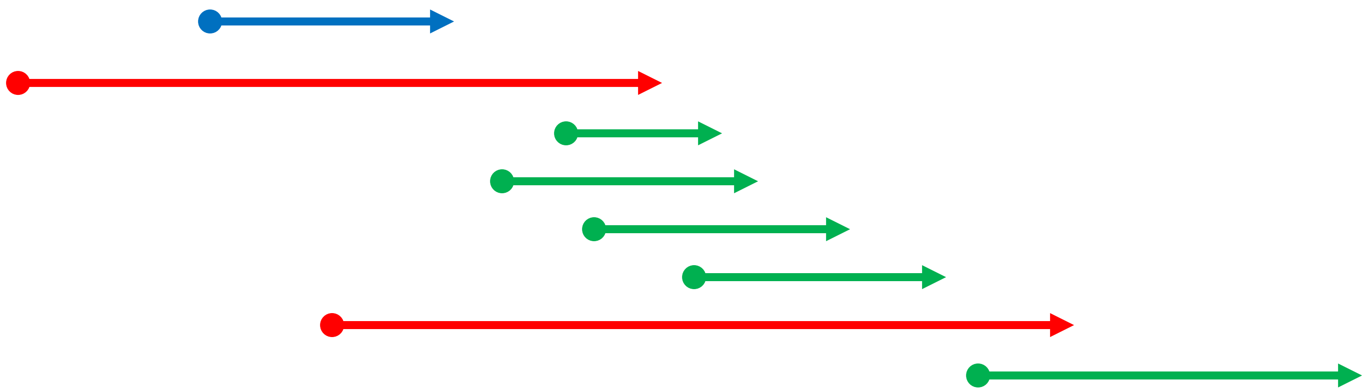


Interval Scheduling Algorithm

Find event ending earliest, add to solution,

Remove it and all conflicting events,

Repeat until all events removed, return solution

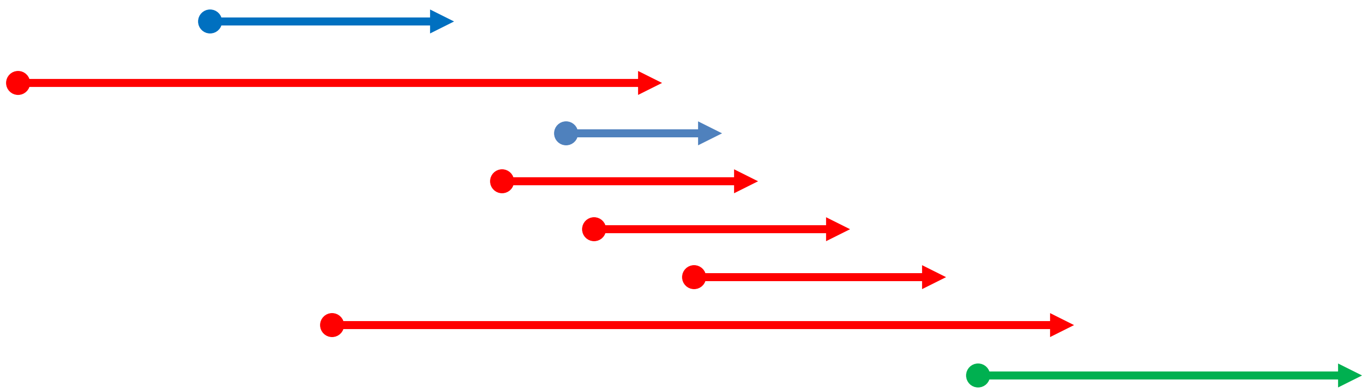


Interval Scheduling Algorithm

Find event ending earliest, add to solution,

Remove it and all conflicting events,

Repeat until all events removed, return solution

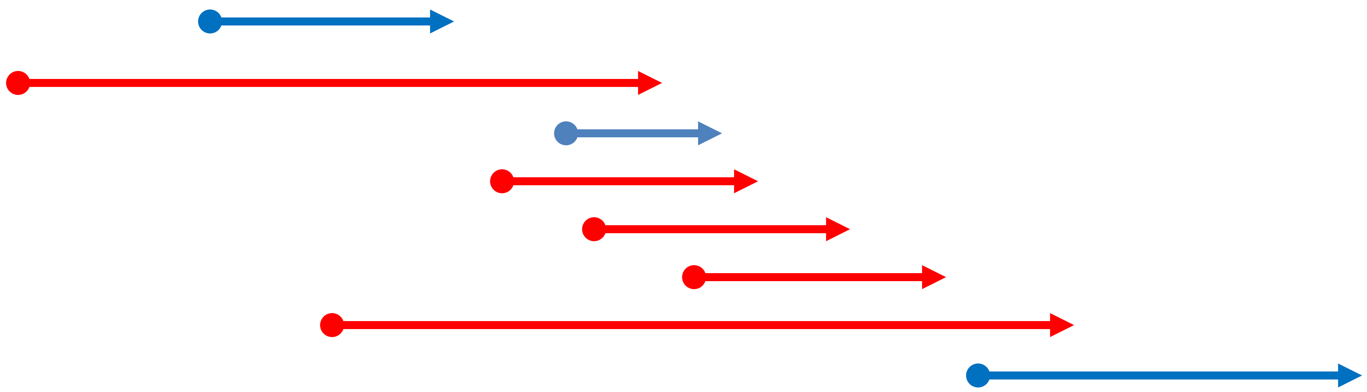


Interval Scheduling Algorithm

Find event ending earliest, add to solution,

Remove it and all conflicting events,

Repeat until all events removed, return solution



Interval Scheduling Run Time

Find event ending earliest, add to solution,

Remove it and all conflicting events,

Repeat until all events removed, return solution

```
Equivalent way

StartTime = 0
For each interval (in order of finish time):                     O(n)
        if begin of interval < StartTime or end of interval < StartTime:     O(1)
                do nothing
        else:
                add interval to solution                          O(1)
                StartTime = end of interval
```

Exchange argument

- Shows correctness of a greedy algorithm
- Idea:
  – Show exchanging an item from an arbitrary optimal solution with your greedy choice makes the new solution no worse
  – How to show my sandwich is at least as good as yours:
    - Show: “I can remove any item from your sandwich, and it would be no worse by replacing it with the same item from my sandwich”

Exchange Argument for Earliest End Time

- Claim: earliest ending interval is always part of <u>some</u> optimal solution

- Let $OPT_{i,j}$ be an optimal solution for time range $[i, j]$
- Let a^* be the first interval in $[i, j]$ to finish overall
- If $a^* \in OPT_{i,j}$ then claim holds
- Else if $a^* \notin OPT_{i,j}$, let a be the first interval to end in $OPT_{i,j}$
  - By definition a^* ends before a, and therefore does not conflict with any other events in $OPT_{i,j}$
  - Therefore $OPT_{i,j} - \{a\} + \{a^*\}$ is also an optimal solution
  - Thus claim holds