

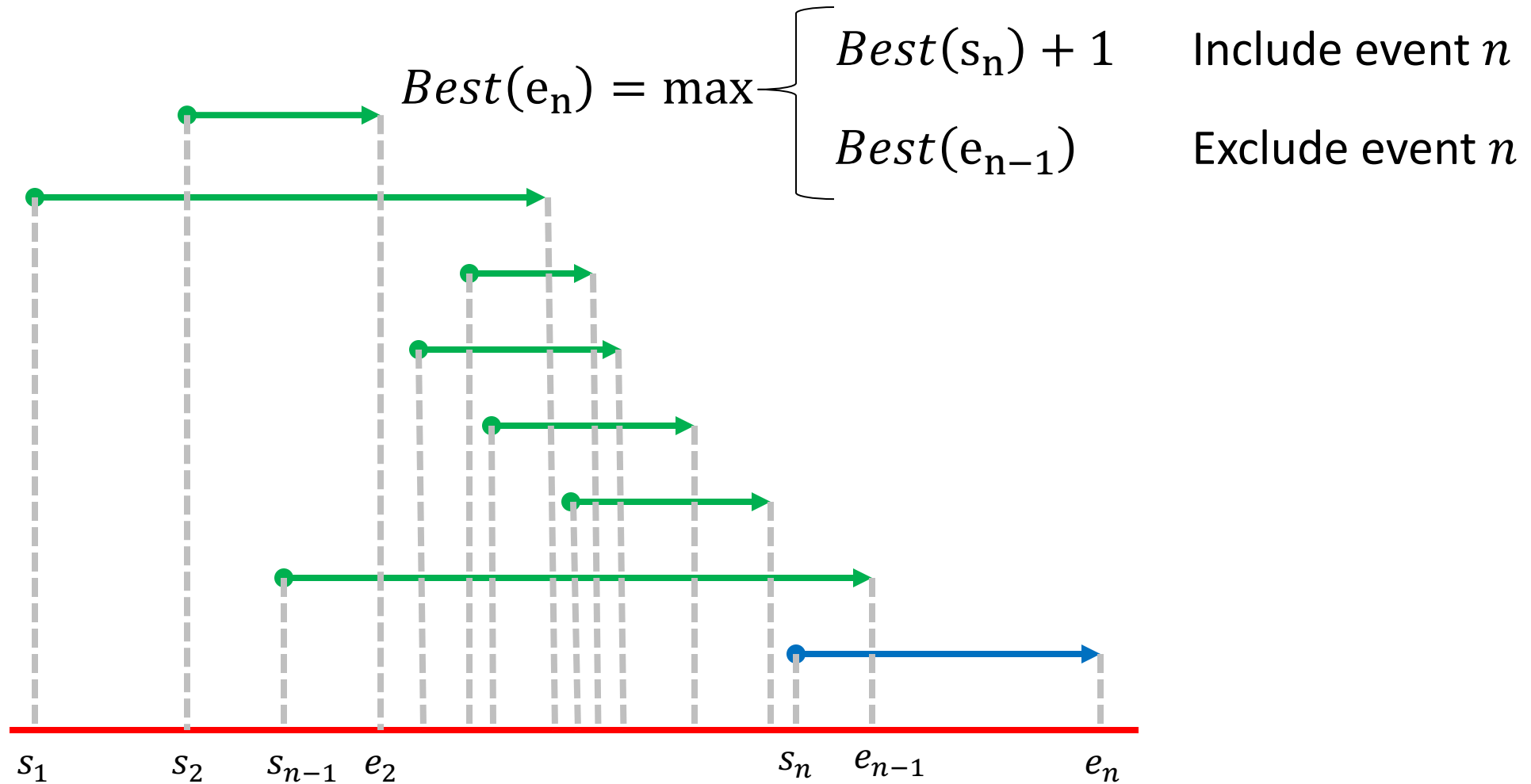
Interval Scheduling

- 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]	Lunch Zoom with friends
[2, 3.25]	CS4102 Live Zoom session
[3, 4]	Streaming department talk
[4, 5.25]	Virtual Office hours
[4.5, 6]	Zoom discussion section
[5, 7.5]	Super Smash Brothers online game night
[7.75, 11]	UVA Basketball Championship re-watch party

Interval Scheduling DP

$Best(t) = \max \# \text{ events that can be scheduled before time } t$



Greedy Algorithms

- Require **Optimal Substructure**
 - Solution to larger problem contains the solution to a smaller one
 - Only one subproblem to consider!
- Idea:
 1. Identify a greedy **choice property**
 - How to make a choice guaranteed to be included in some optimal solution
 2. Repeatedly apply the choice property until no subproblems remain

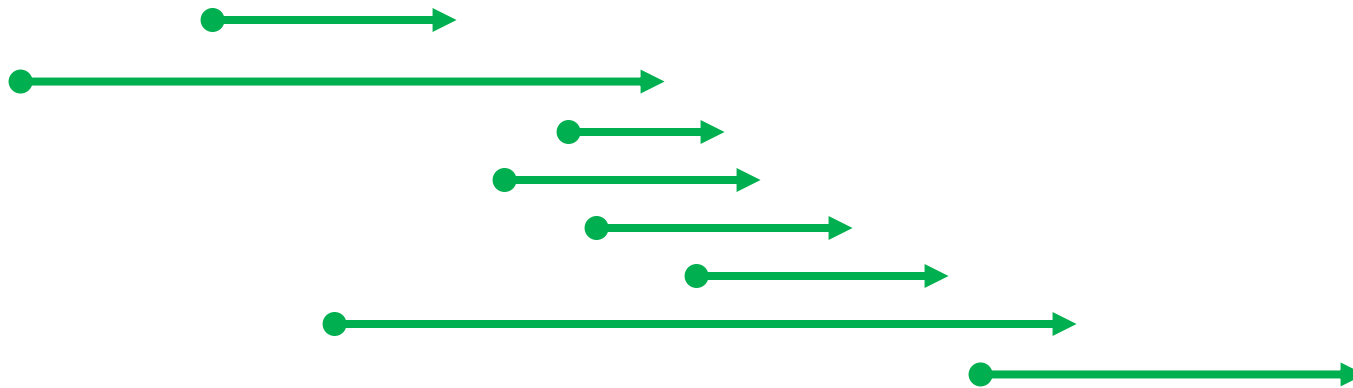
Greedy Interval Scheduling

Interval Scheduling Algorithm

Find event ending earliest, add to solution,

Remove **it** and **all conflicting events**,

Repeat until all events removed, return **solution**

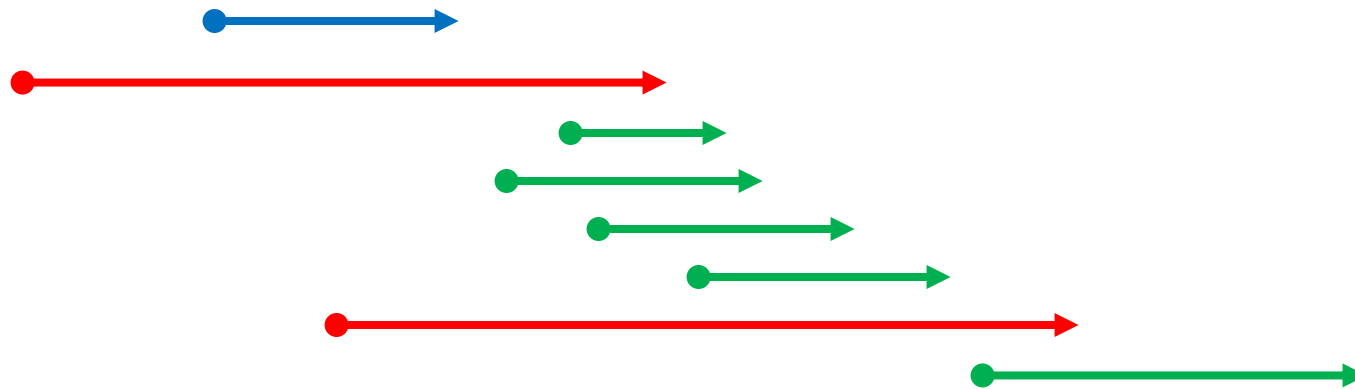


Interval Scheduling Algorithm

Find event ending earliest, add to solution,

Remove **it** and **all conflicting events**,

Repeat until all events removed, return **solution**

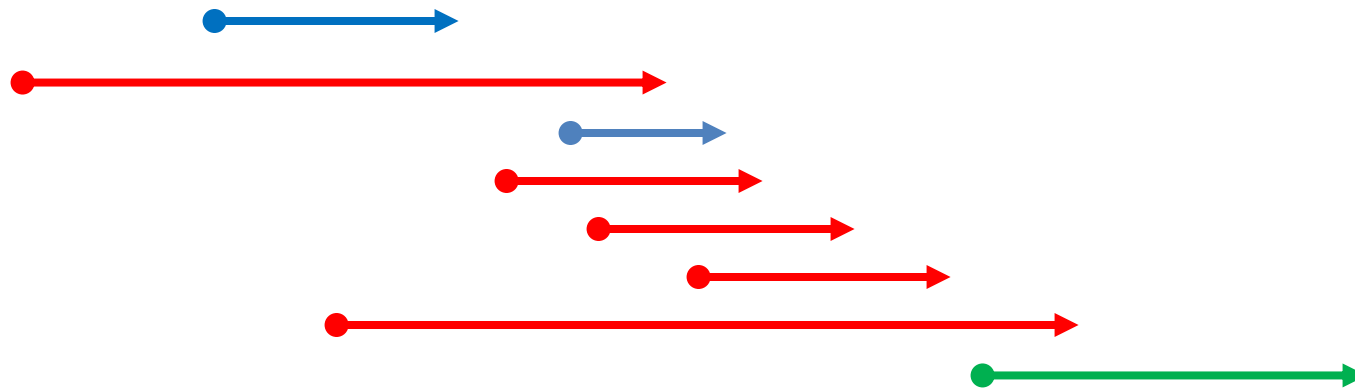


Interval Scheduling Algorithm

Find event ending earliest, add to solution,

Remove **it** and **all conflicting events**,

Repeat until all events removed, return **solution**

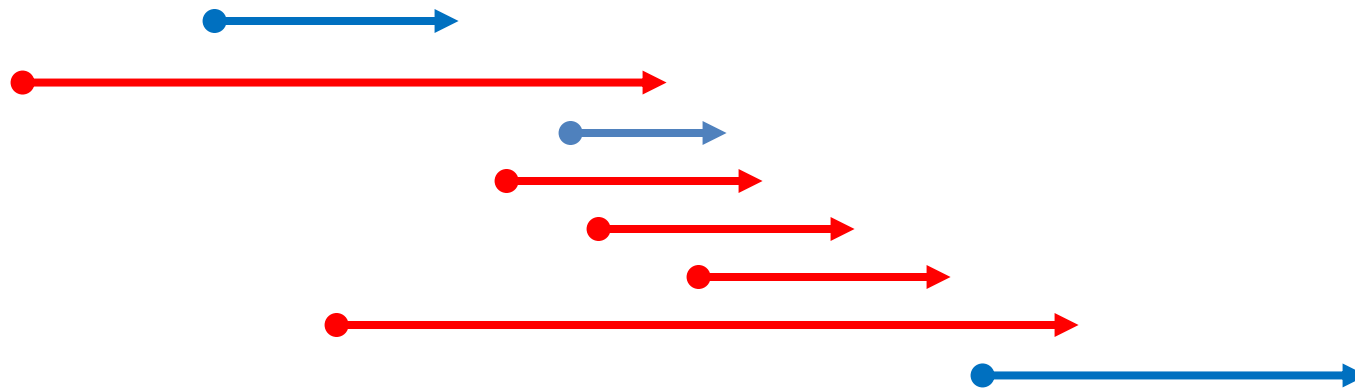


Interval Scheduling Algorithm

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**

Sort intervals by finish time

StartTime = 0

for each interval (in order of finish time):

 if begin of interval > StartTime:

 add interval to solution

 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