CSL 356: Analysis and Design of Algorithms

Ragesh Jaiswal

CSE, IIT Delhi

Greedy Algorithms

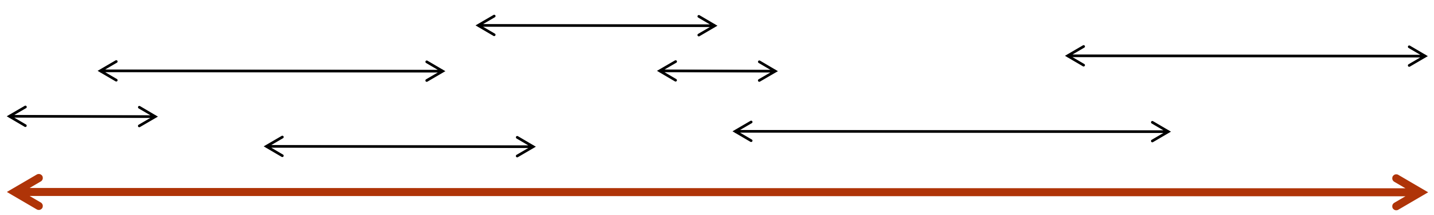
Greedy Algorithms: Introduction

- *A local (greedy) decision rule leads to a globally optimal solution.*

- Two ways to show the above property:
  1. *Greedy stays ahead.*
  2. *Exchange argument*

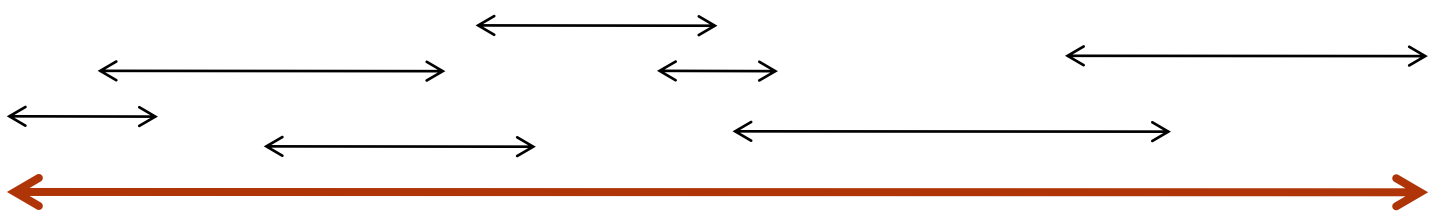
Greedy Algorithms: Example

- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.



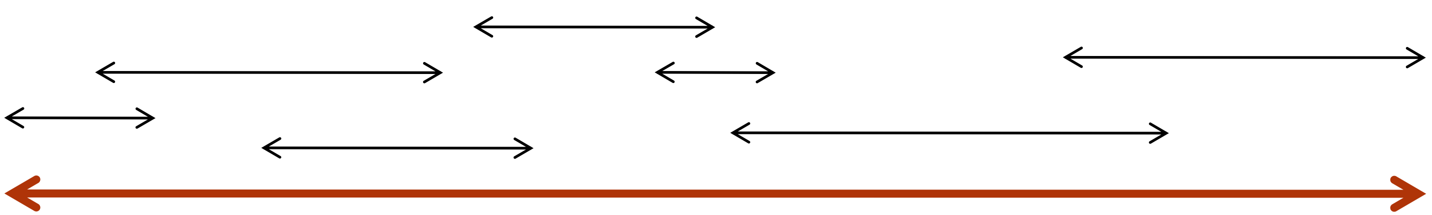
Greedy Algorithms: Example

- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Candidate greedy choices:
  - Earliest start time



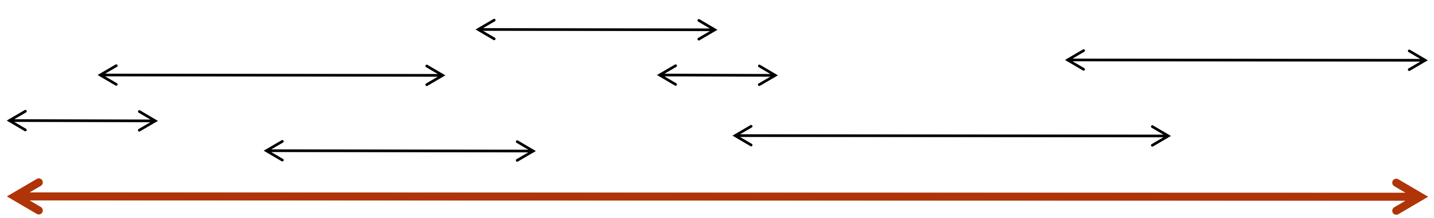
Greedy Algorithms: Example

- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Candidate greedy choices:
  - Earliest start time
  - Smallest duration



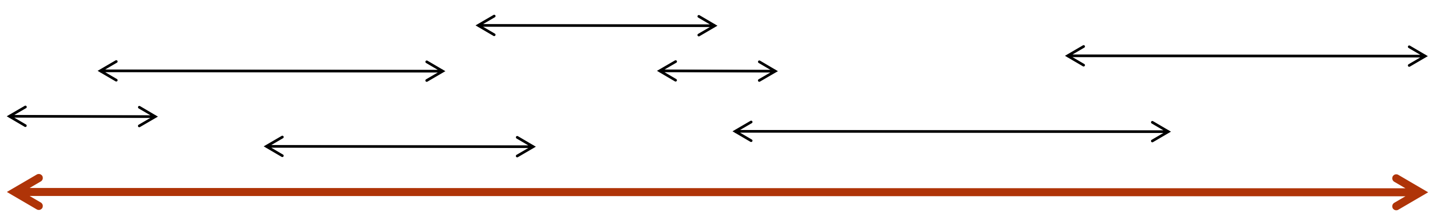
Greedy Algorithms: Example

- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Candidate greedy choices:
  - Earliest start time
  - Smallest duration
  - Least overlapping



Greedy Algorithms: Example

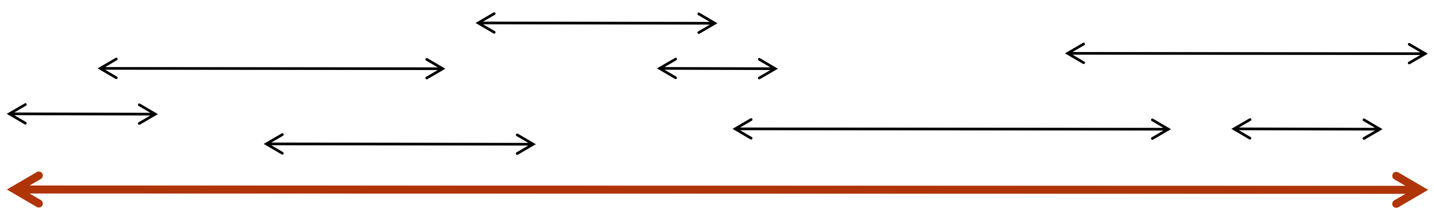
- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Candidate greedy choices:
  - Earliest start time
  - Smallest duration
  - Least overlapping
  - Earliest finish time



Greedy Algorithms: Example

- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Greedy Algorithm:

GreedySchedule
    While R is not empty
        - Choose an interval $(S(i), F(i))$ from R that has the smallest value of $F(i)$
        - Delete all intervals in R that overlap with $(S(i), F(i))$

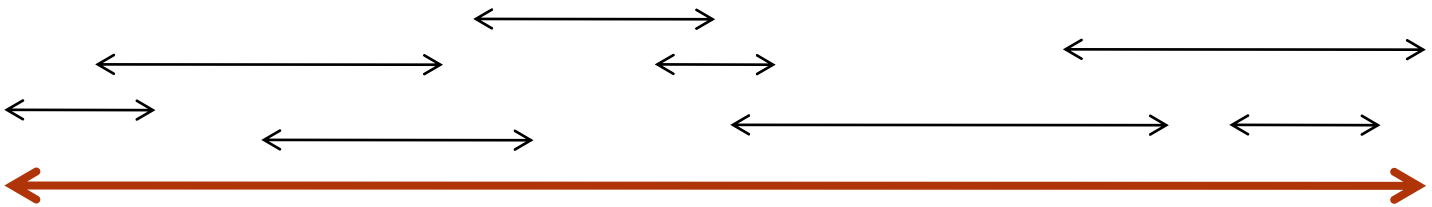


Greedy Algorithms: Example

- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Greedy Algorithm:

GreedySchedule
    While R is not empty
        - Choose an interval $(S(i), F(i))$ from R that has the smallest value of $F(i)$
        - Delete all intervals in R that overlap with $(S(i), F(i))$

- Question: Let O denote some optimal subset and A be the subset given by GreedySchedule. Can we show that $A = O$?



Greedy Algorithms: Example

- Can we show that $|A| = |O|$?
- Yes we can! We will use the *greedy stays ahead* method to show this.
- Proof: Let $a_1, a_2, \ldots, a_k$ be the sequence of requests that GreedySchedule picks and $o_1, o_2, \ldots, o_l$ be the requests in O sorted by finishing time.
  - Claim: $F(a_1) \leq F(o_1)$

Greedy Algorithms: Example

- Can we show that $|A| = |O|$?
- Yes we can! We will use the *greedy stays ahead* method to show this.
- Proof: Let $a_1, a_2, \ldots, a_k$ be the sequence of requests that GreedySchedule picks and $o_1, o_2, \ldots, o_l$ be the requests in O sorted by finishing time.
  - Claim: $F(a_1) \leq F(o_1)$
  - Claim: If $F(a_1) \leq F(o_1), F(a_2) \leq F(o_2), \ldots, F(a_{i-1}) \leq F(o_{i-1})$, then $F(a_i) \leq F(o_i)$.

Greedy Algorithms: Example

- Can we show that $|A| = |O|$?
- Yes we can! We will use the *greedy stays ahead* method to show this.
- Proof: Let $a_1, a_2, \ldots, a_k$ be the sequence of requests that GreedySchedule picks and $o_1, o_2, \ldots, o_l$ be the requests in O sorted by finishing time.
  - Claim: $F(a_1) \leq F(o_1)$
  - Claim: If $F(a_1) \leq F(o_1), F(a_2) \leq F(o_2), \ldots, F(a_{i-1}) \leq F(o_{i-1})$, then $F(a_i) \leq F(o_i)$.
  - GreedySchedule could not have stopped after a_k

Greedy Algorithms: Example

- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Greedy Algorithm:

> GreedySchedule
> While R is not empty
> - Choose an interval $(S(i), F(i))$ from R that has the smallest value of $F(i)$
> - Delete all intervals in R that overlap with $(S(i), F(i))$

- Running time:

Greedy Algorithms: Example

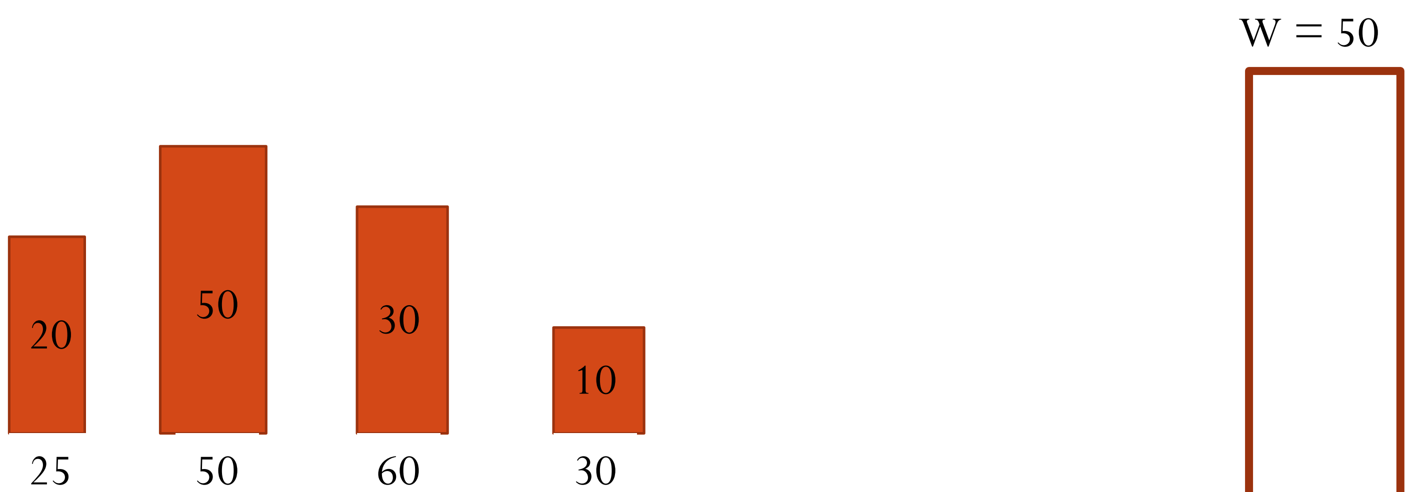
- Interval scheduling: Given a set of n intervals of the form $(S(i), F(i))$, find the largest subset of non-overlapping intervals.
- Greedy Algorithm:

> GreedySchedule
> While R is not empty
> - Choose an interval $(S(i), F(i))$ from R that has the smallest value of $F(i)$
> - Delete all intervals in R that overlap with $(S(i), F(i))$

- Running time: $O(n \log n)$

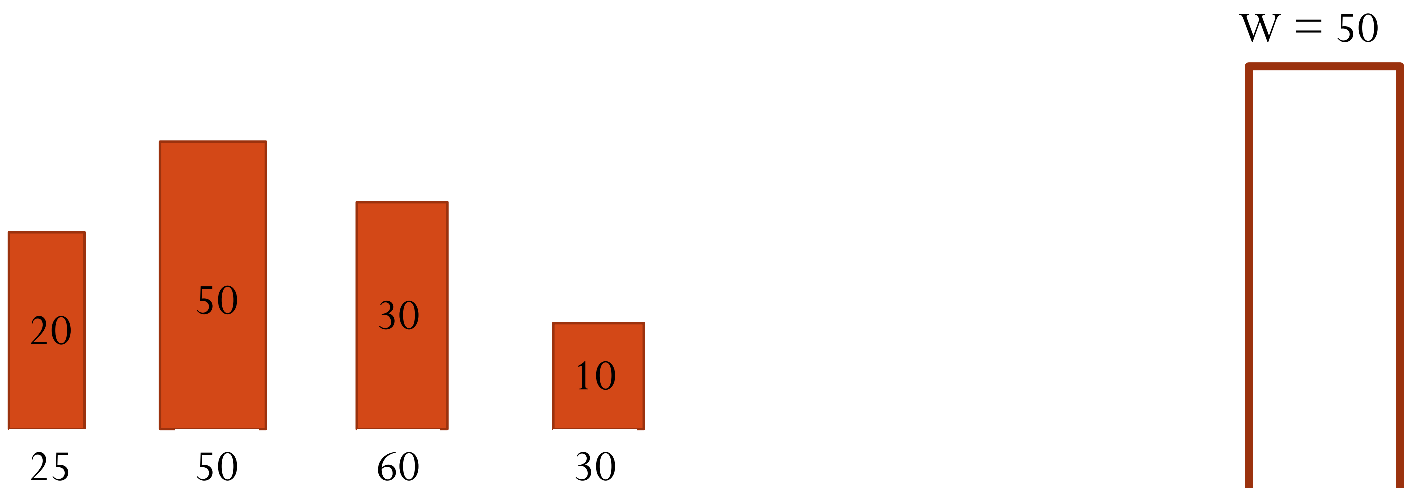
Greedy Algorithms: Example

- Fractional Knapsack: You are a thief and you have a sack of size W. There are n divisible items. Each item i has a volume $W(i)$ and total value $V(i)$. How will you maximize your profit?
- Greedy strategies:
  - Pick items with largest value first.



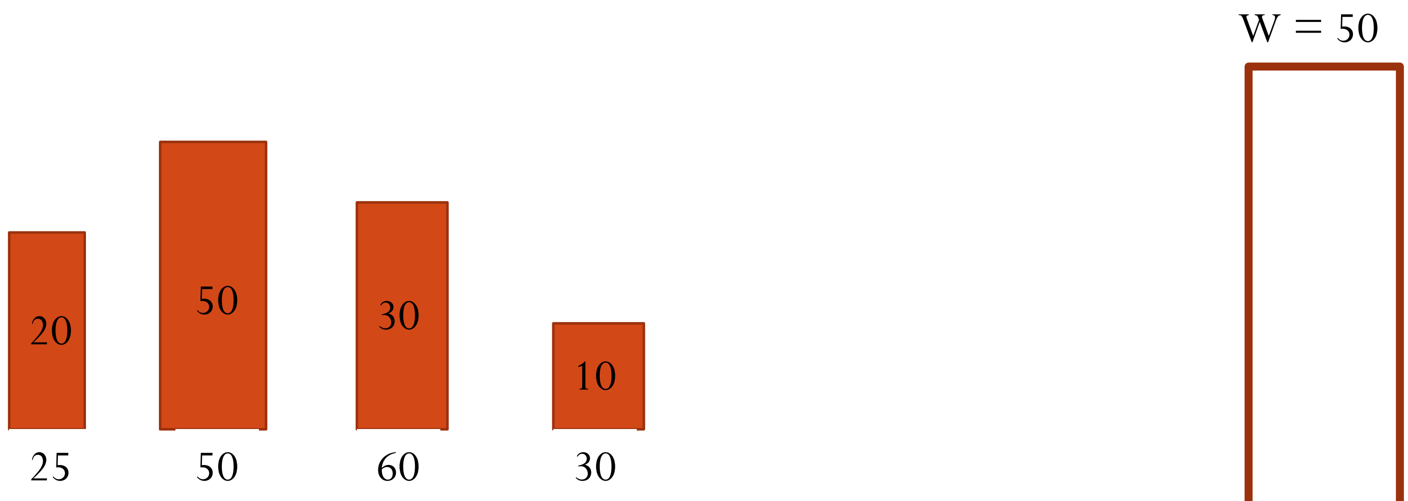
Greedy Algorithms: Example

- Fractional Knapsack: You are a thief and you have a sack of size W. There are n divisible items. Each item i has a volume $W(i)$ and total value $V(i)$. How will you maximize your profit?
- Greedy strategies:
  - Pick items with largest value first.
  - Pick items with smallest volume first.



Greedy Algorithms: Example

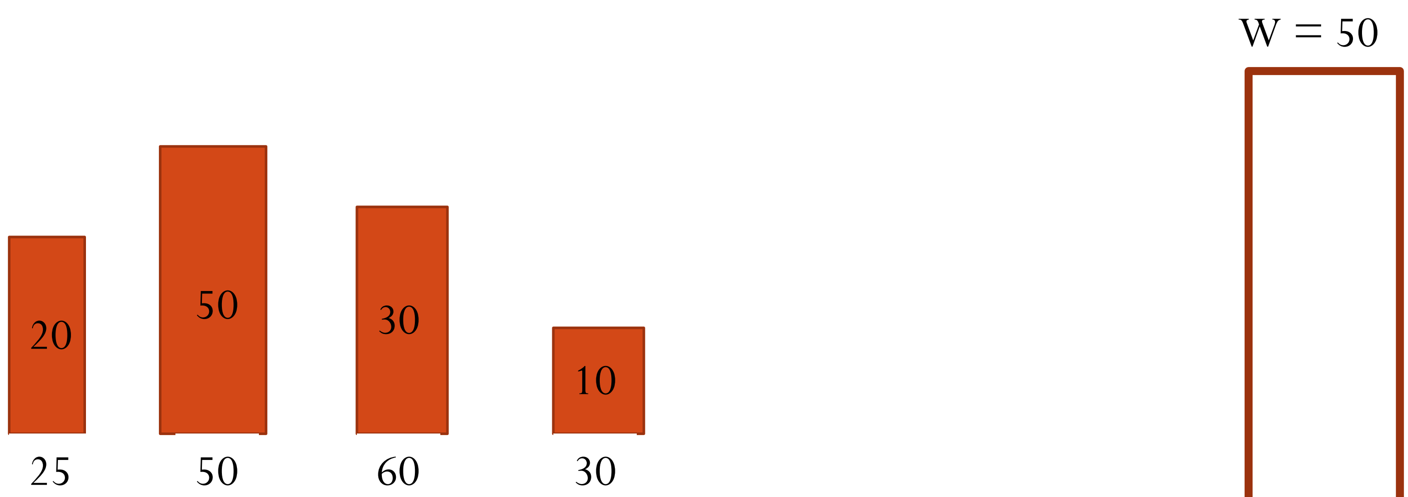
- <u>Fractional Knapsack</u>: You are a thief and you have a sack of size W. There are n divisible items. Each item i has a volume $W(i)$ and total value $V(i)$. How will you maximize your profit?
- Greedy strategies:
  - Pick items with largest value first.
  - Pick items with smallest volume first.
  - Pick items with largest cost per unit volume.



Greedy Algorithms: Example

- <u>Fractional Knapsack</u>: You are a thief and you have a sack of size W. There are n divisible items. Each item i has a volume $W(i)$ and total value $V(i)$. How will you maximize your profit?

GreedySteal
    While Sack is not full
        - Choose an item i from R that has the largest cost per unit volume
        - Put as much as you can of this item in the sack and delete i from R



Greedy Algorithms: Example

- Consider items in decreasing order of the cost per unit volume value.
- Let $(G_1, \ldots, G_n)$ be the volume of items in the sack chosen by GreedySteal.
- Let $(O_1, \ldots, O_n)$ be some optimal volume of items that maximizes the profit.
- Claim: For all i,

$$G_1 * d_1 + \cdots + G_i * d_i \geq O_1 * d_1 + \cdots + O_i * d_i$$

End

Problems to think about:

1. Consider the fractional-knapsack problem. Think of an *exchange argument* to prove that the greedy algorithm gives the optimal solution.