CSL 356: Analysis and Design of Algorithms

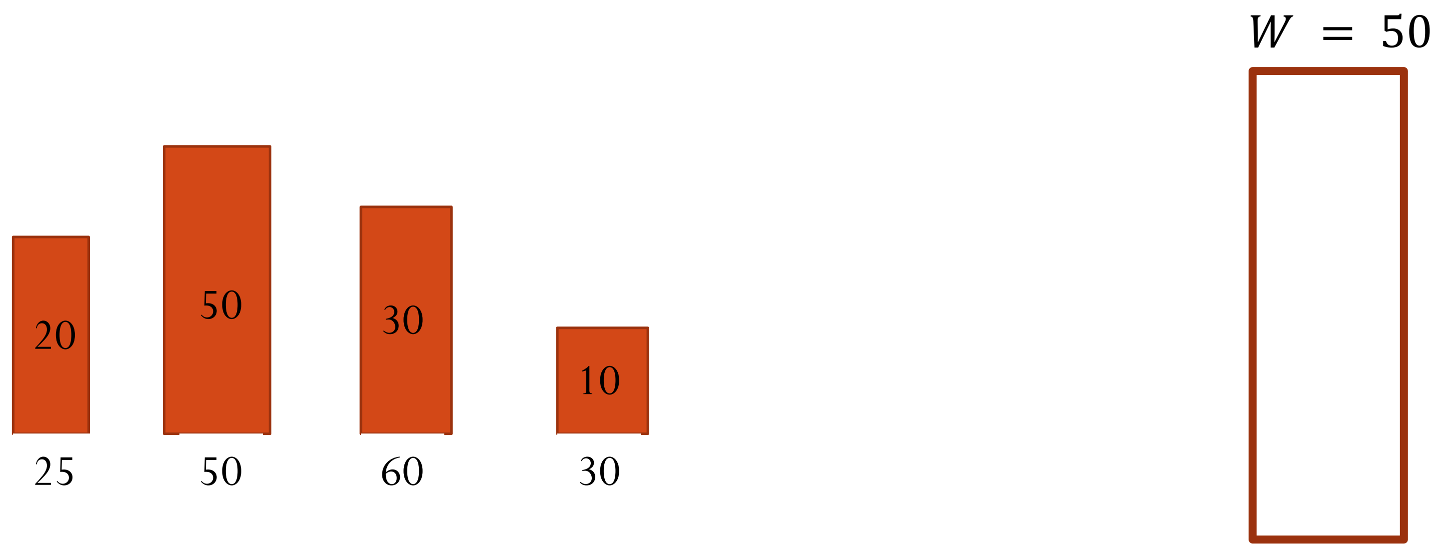
Ragesh Jaiswal

CSE, IIT Delhi

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?

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

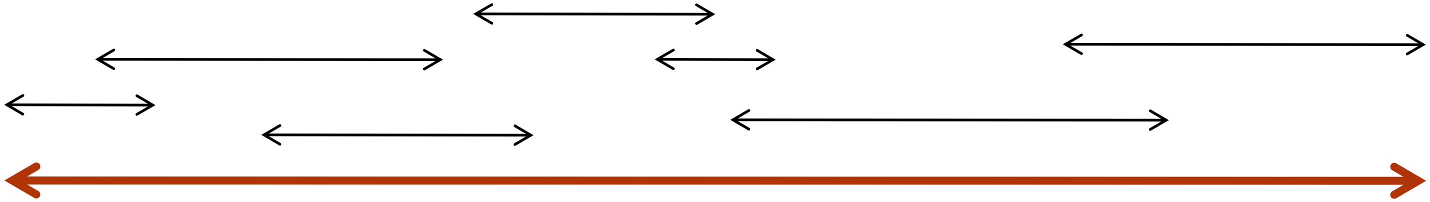


Greedy Algorithms: Example

- 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.
- <u>Claim</u>: For all i, $G_1 * d_1 + \cdots + G_i * d_i \geq O_1 * d_1 + \cdots + O_i * d_i$

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 Algorithms: Example

- Job Scheduling: You are given n jobs and you are supposed to schedule these jobs on a machine. Each job i consists of a duration $T(i)$ and a deadline $D(i)$. The lateness of a job wrt. a schedule is defined as $\max(0, F(i) - D(i))$, where $F(i)$ is the finishing time of job i as per the schedule. The goal is to minimize the maximum lateness.
- Greedy Strategies:
  - Smallest jobs first.

Greedy Algorithms: Example

- Job Scheduling: You are given n jobs and you are supposed to schedule these jobs on a machine. Each job i consists of a duration $T(i)$ and a deadline $D(i)$. The lateness of a job wrt. a schedule is defined as $\max(0, F(i) - D(i))$, where $F(i)$ is the finishing time of job i as per the schedule. The goal is to minimize the maximum lateness.
- Greedy Strategies:
  - Smallest jobs first.
  - Earliest deadline first.

GreedyJobSchedule
- Sort the Jobs in increasing order of deadlines and schedule the jobs on the machine in this order.

Greedy Algorithms: Example

- Claim: There is an optimal schedule with no idle time (time when the machine is idle).
- Definition: A schedule is said to have inversion if there are a pair of jobs (i, j) such that
  1. $D(i) < D(j)$, and
  2. Job j is performed before job i as per the schedule.
- Claim: There is an optimal schedule with no idle time and no inversion.
- Proof: Consider an optimal schedule O. First if there is any idle time we obtain another optimal schedule O_1 without idle time. Suppose O_1 has inversions. Consider one such inversion (i, j).

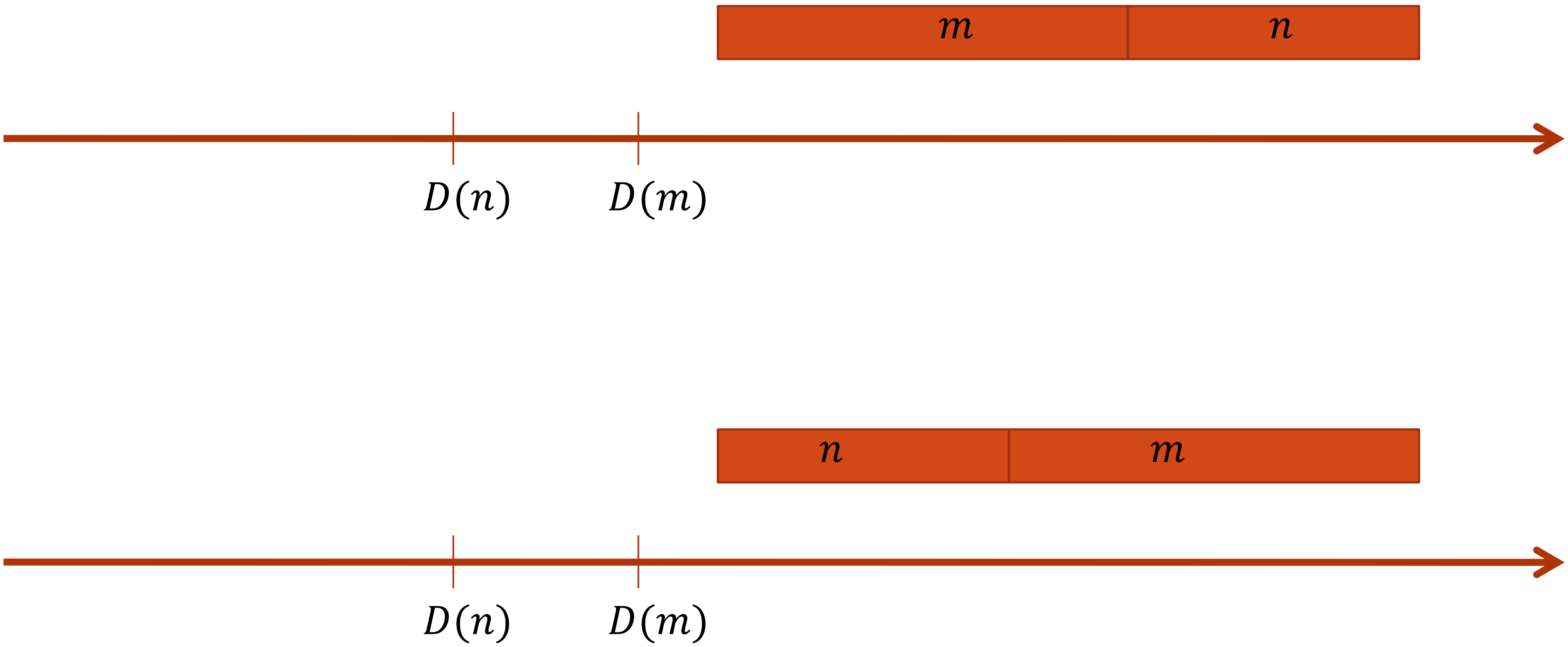
Greedy Algorithms: Example

- Claim: There is an optimal schedule with no idle time and no inversion.
- Proof: Consider an optimal schedule O. First if there is any idle time we obtain another optimal schedule O_1 without idle time. Suppose O_1 has inversions. Consider one such inversion (i, j).

- There exists a pair of adjacently scheduled jobs (m, n) such that the schedule has an inversion wrt. (m, n).

Greedy Algorithms: Example

- Claim: *Exchanging* m and n does not increase the maximum lateness.



End

Problems to think about:

1. Consider the job scheduling problem. Can you think of an example where there is a **unique** optimal schedule.