Greedy Interval Scheduling

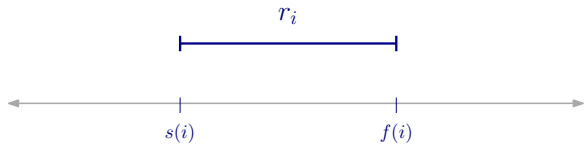
- Interval Scheduling - Generic Greedy Algorithm
- Sub-Optimal Greedy Algorithms
  - Earliest Starting Request First
  - Latest Finishing Request First
  - Shortest Duration Request First
  - Least Conflicting Request First
- Earliest Finish Time First Algorithm
  - Correctness and Optimality
  - Implementation and Runtime

Imdad ullah Khan

Interval Scheduling: Problem

- $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ (set of requests)
- Starting and finishing time of r_i: $s(i)$ and $f(i)$

  for $1 \leq i \leq n$ $\quad s(i) < f(i)$
- Duration of request r_i is d_i is $f(i)$ - $s(i)$



r_i and r_j are **compatible** if they do not overlap in time

Otherwise r_i and r_j are **conflicting**

$$\underbrace{s(i) < f(i) < s(j) < f(j)}_{r_i \text{ is to the left of } r_j} \quad \text{OR} \quad \underbrace{s(j) < f(j) < s(i) < f(i)}_{r_i \text{ is to the right of } r_j}$$

A set is compatible if all pairs in it are compatible

Interval Scheduling: Generic Greedy Algorithm

Input: A set $\mathcal{R}$ of requests

Output: A largest compatible subset $S \subset \mathcal{R}$

Generic Greedy algorithm

Process requests in a fixed order and select compatible requests greedily

---

Algorithm Geedy Interval Scheduling Algorithm ($\mathcal{R}$)

$A \leftarrow \emptyset$
while $\mathcal{R} \neq \emptyset$ **do**
  select a request r_x from $\mathcal{R}$
  remove from $\mathcal{R}$ all those requests conflicting with r_x
  $A \leftarrow A \cup \{r_x\}$
return A

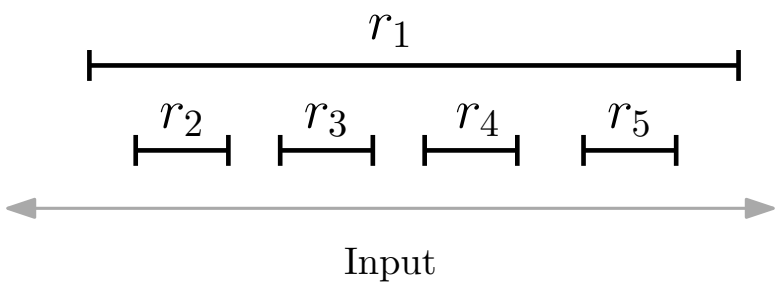
By construction the algorithm is correct ▷ (A is a compatible subset)

Interval Scheduling: Earliest Starting Request First

Greedy Algorithm: Earliest starting request first

▷ Idea is to start using resource as early as possible

Optimal?

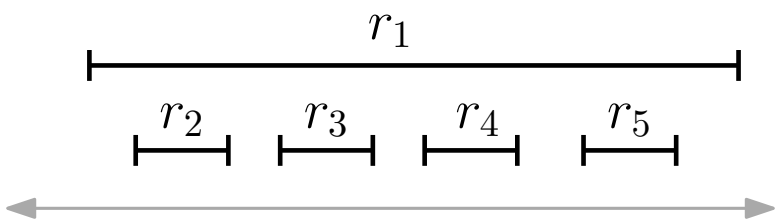


Interval Scheduling: Earliest Starting Request First

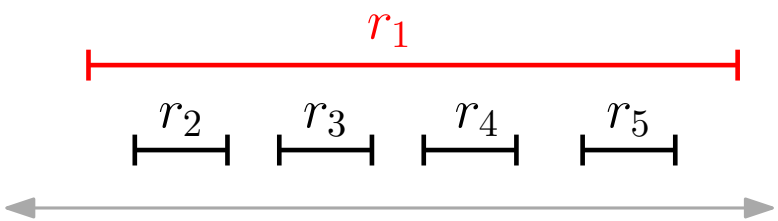
Greedy Algorithm: Earliest starting request first

▷ Idea is to start using resource as early as possible

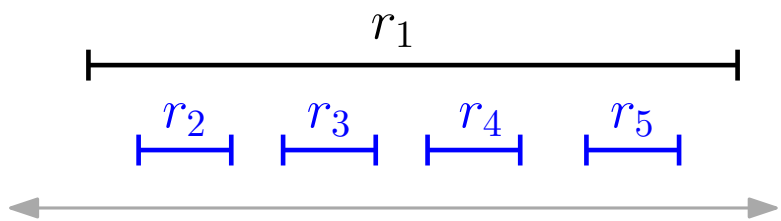
Optimal?



Input



Algorithm's solution



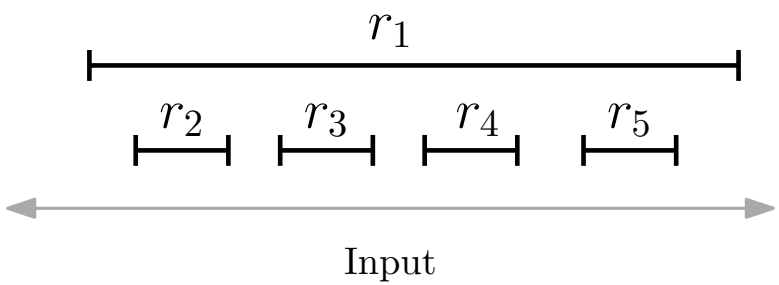
Optimal solution

Interval Scheduling: Latest Finishing Request First

Greedy Algorithm: Latest finishing request first

▷ Idea is to keep using resource as long as possible

Optimal?

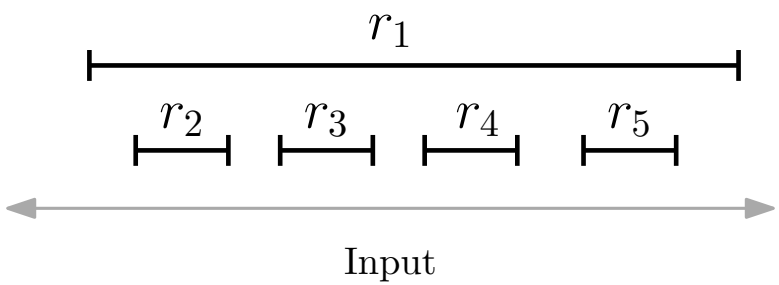


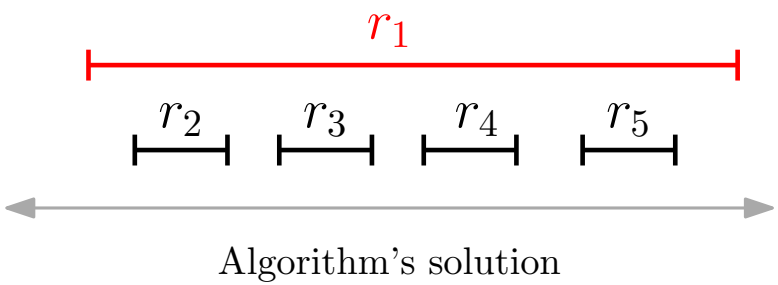
Interval Scheduling: Latest Finishing Request First

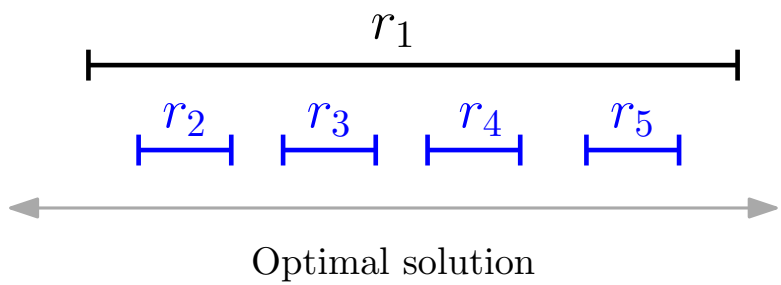
Greedy Algorithm: Latest finishing request first

▷ Idea is to keep using resource as long as possible

Optimal?





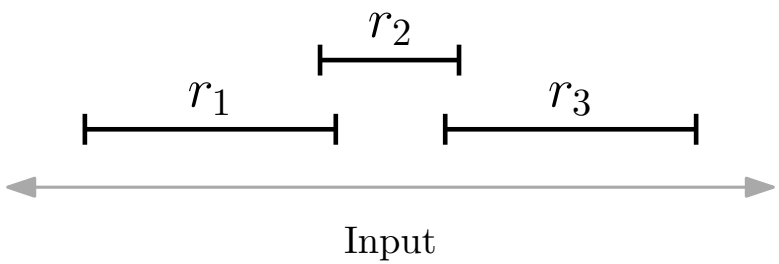


Interval Scheduling: Shortest Duration Request First

Greedy Algorithm: Shortest duration request first

▷ Idea is to incorporate duration of requests

Optimal?

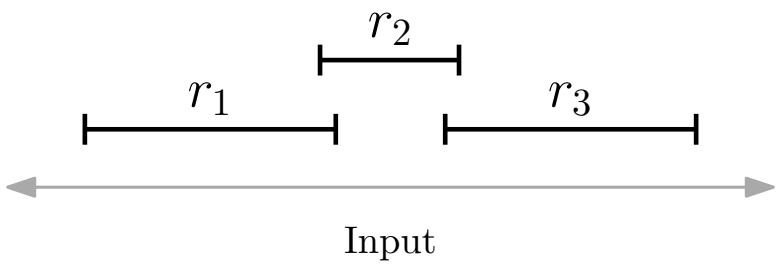


Interval Scheduling: Shortest Duration Request First

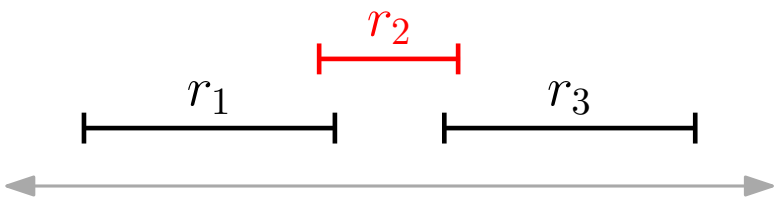
Greedy Algorithm: Shortest duration request first

▷ Idea is to incorporate duration of requests

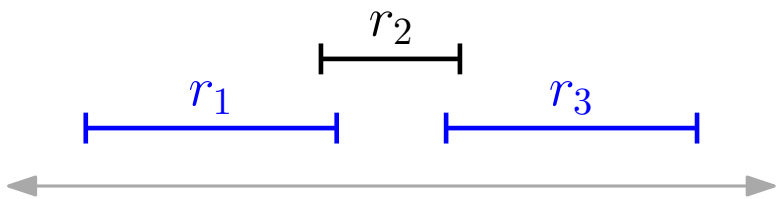
Optimal?



Input



Algorithm's solution



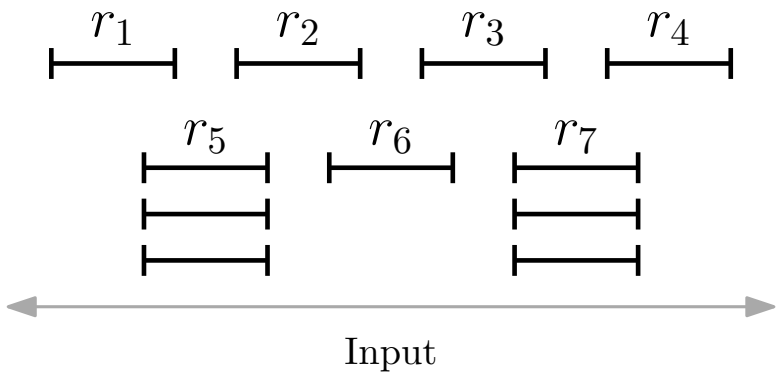
Optimal solution

Interval Scheduling: Least Conflicting Request First

Greedy Algorithm: Least conflicting request first

▷ Idea is to incorporate conflicts between requests

Optimal?

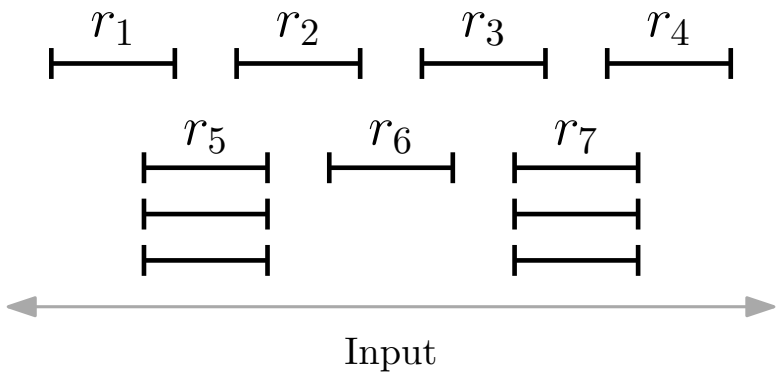


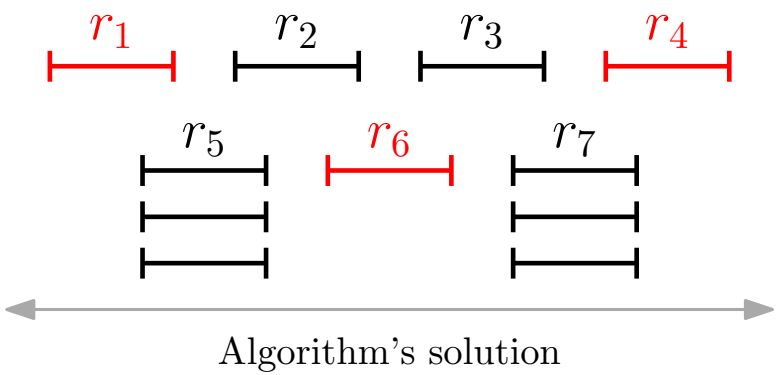
Interval Scheduling: Least Conflicting Request First

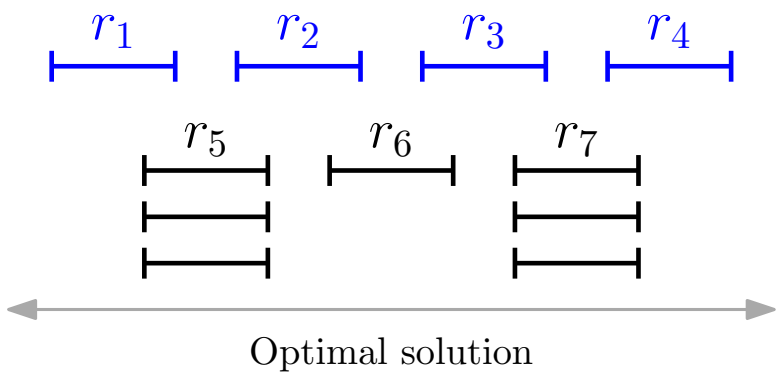
Greedy Algorithm: Least conflicting request first

▷ Idea is to incorporate conflicts between requests

Optimal?





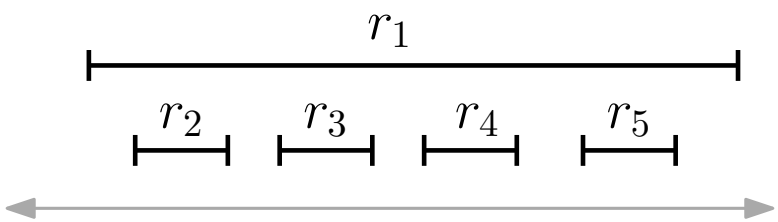


Interval Scheduling: Earliest Finishing Request First

Greedy Algorithm: Earliest finishing request first

▷ Idea is to make resource free as soon as possible

Optimal?

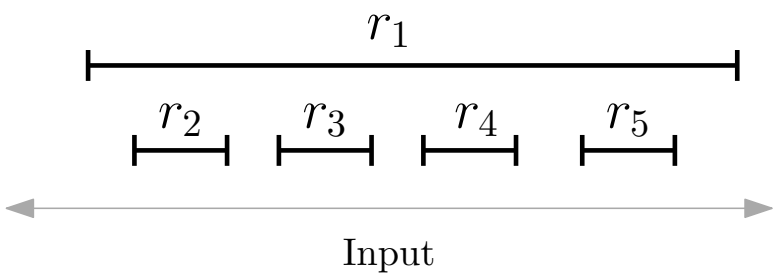


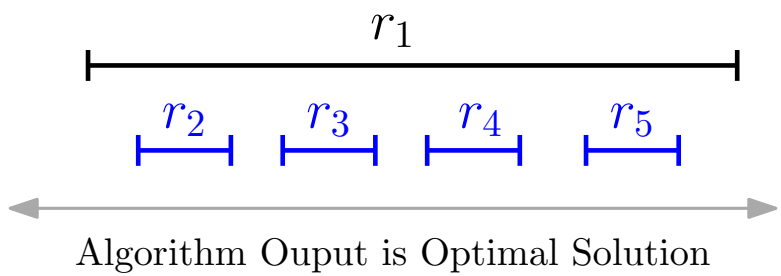
Interval Scheduling: Earliest Finishing Request First

Greedy Algorithm: Earliest finishing request first

▷ Idea is to make resource free as soon as possible

Optimal?



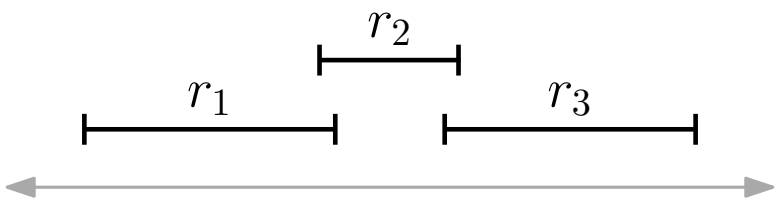


Interval Scheduling: Greedy Algorithm

Greedy Algorithm: Earliest finishing request first

▷ Idea is to make resource free as soon as possible

Optimal?

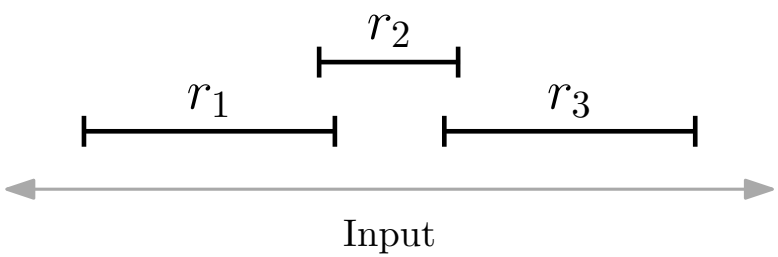


Interval Scheduling: Greedy Algorithm

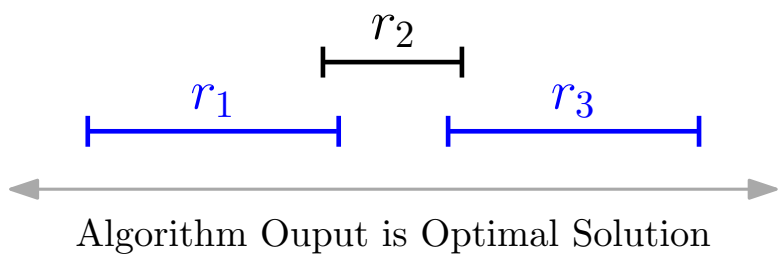
Greedy Algorithm: Earliest finishing request first

▷ Idea is to make resource free as soon as possible

Optimal?



Input



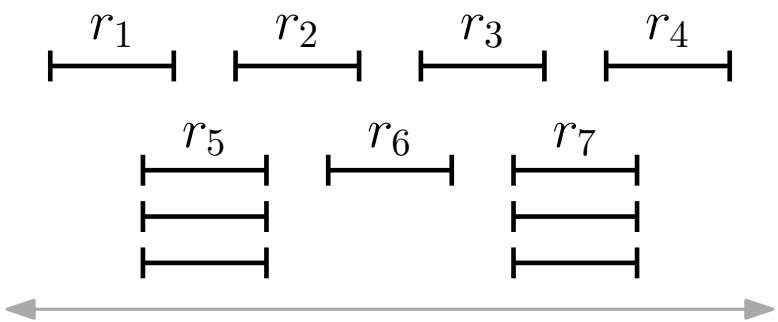
Algorithm Ouput is Optimal Solution

Interval Scheduling: Greedy Algorithm

Greedy Algorithm: Earliest finishing request first

▷ Idea is to make resource free as soon as possible

Optimal?

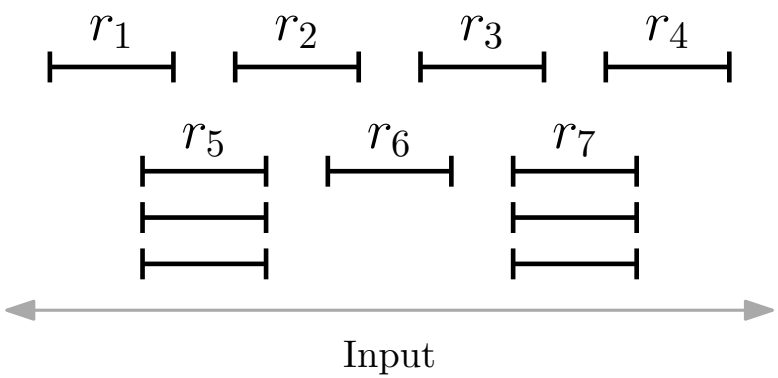


Interval Scheduling: Greedy Algorithm

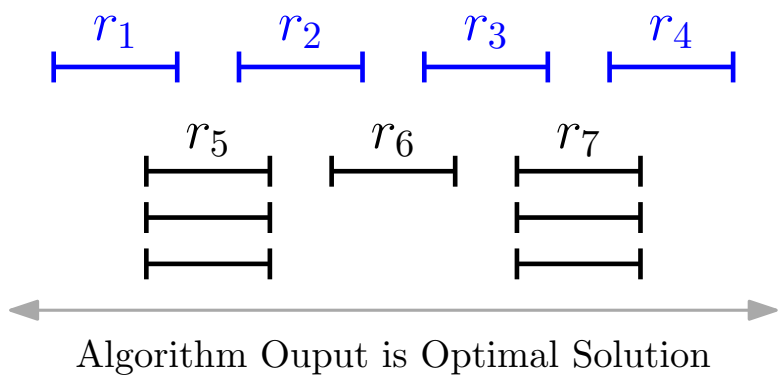
Greedy Algorithm: Earliest finishing request first

▷ Idea is to make resource free as soon as possible

Optimal?



Input



Algorithm Ouput is Optimal Solution