Chapter 2
Greedy Algorithms

Algorithm Theory
WS 2018/19

Fabian Kuhn

Greedy Algorithms

- No clear definition, but essentially:

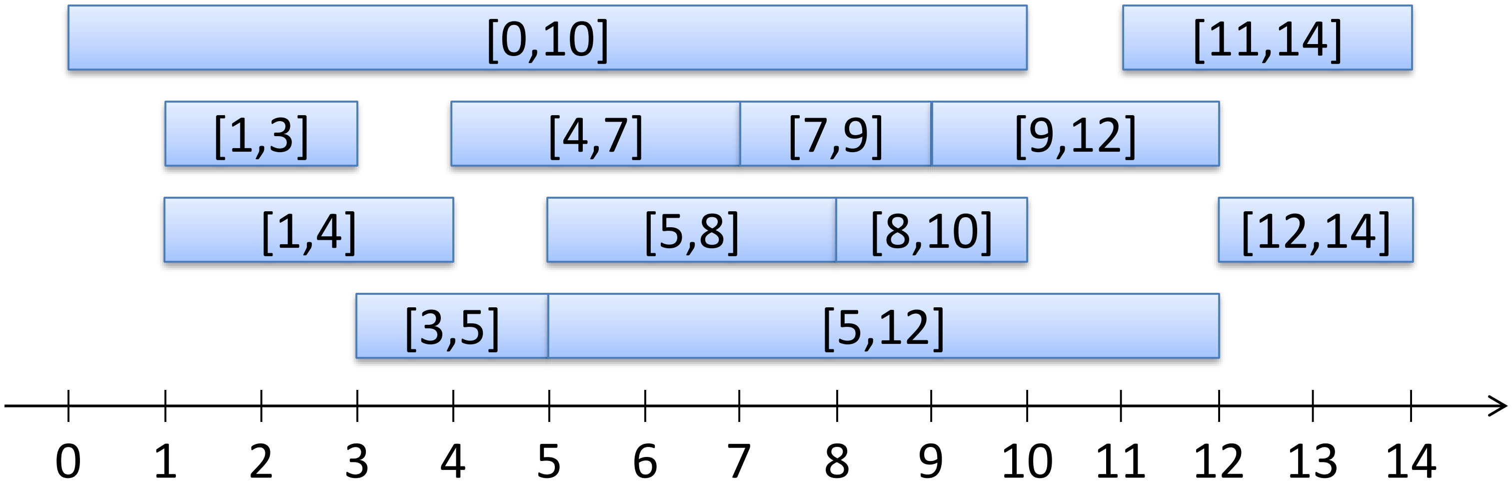
In each step make the choice that looks best at the moment!

- Depending on problem, greedy algorithms can give
  - Optimal solutions
  - Close to optimal solutions
  - No (reasonable) solutions at all
- If it works, very interesting approach!
  - And we might even learn something about the structure of the problem

Goal: Improve understanding where it works (mostly by examples)

Interval Scheduling

- **Given:** Set of intervals, e.g.
  [0,10],[1,3],[1,4],[3,5],[4,7],[5,8],[5,12],[7,9],[9,12],[8,10],[11,14],[12,14]



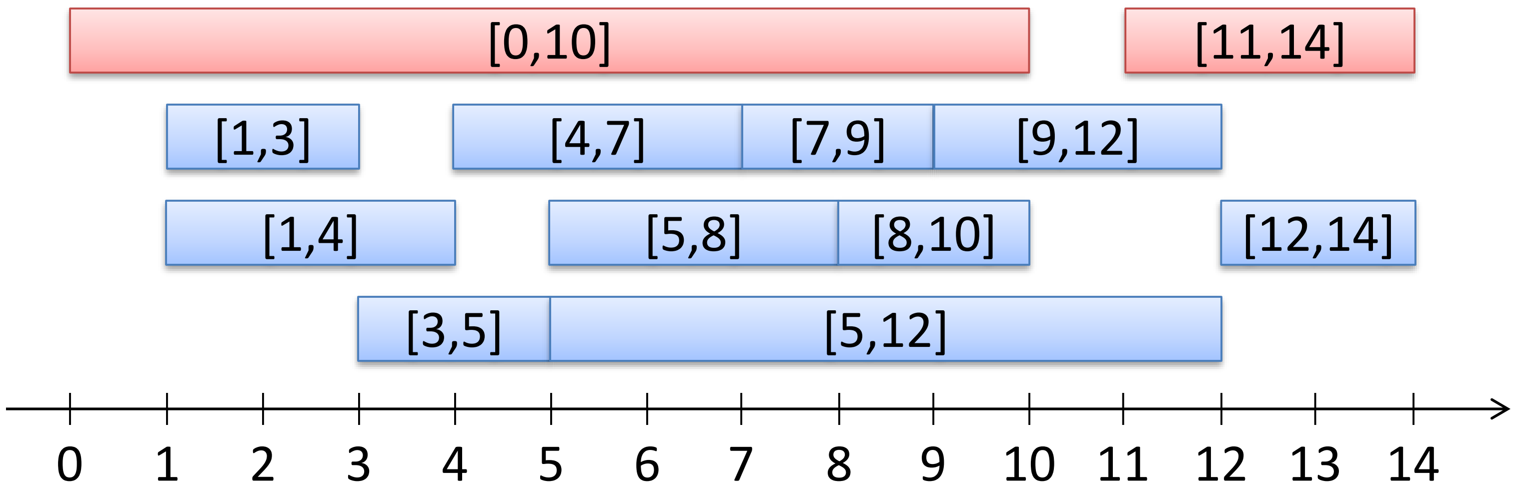
- **Goal:** Select largest possible non-overlapping set of intervals
  - For simplicity: overlap at boundary ok
    (i.e., [4,7] and [7,9] are non-overlapping)

- **Example:** Intervals are room requests; satisfy as many as possible

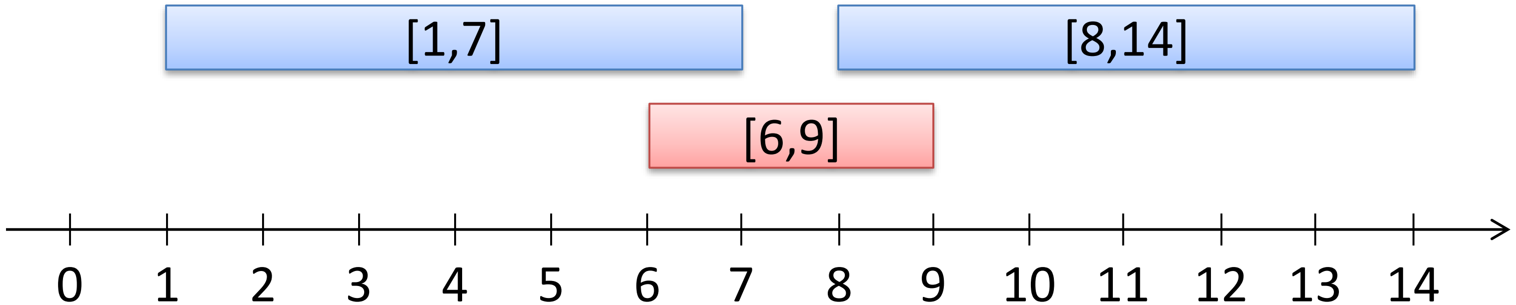
Greedy Algorithms

- Several possibilities…

Choose first available interval:

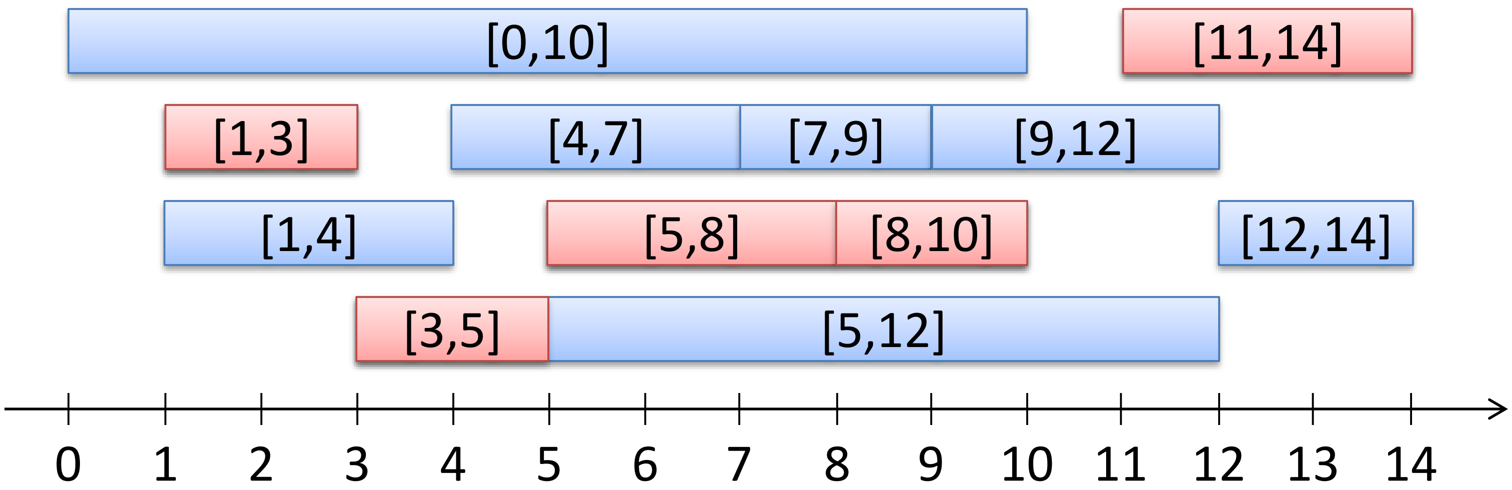


Choose shortest available interval:



Greedy Algorithms

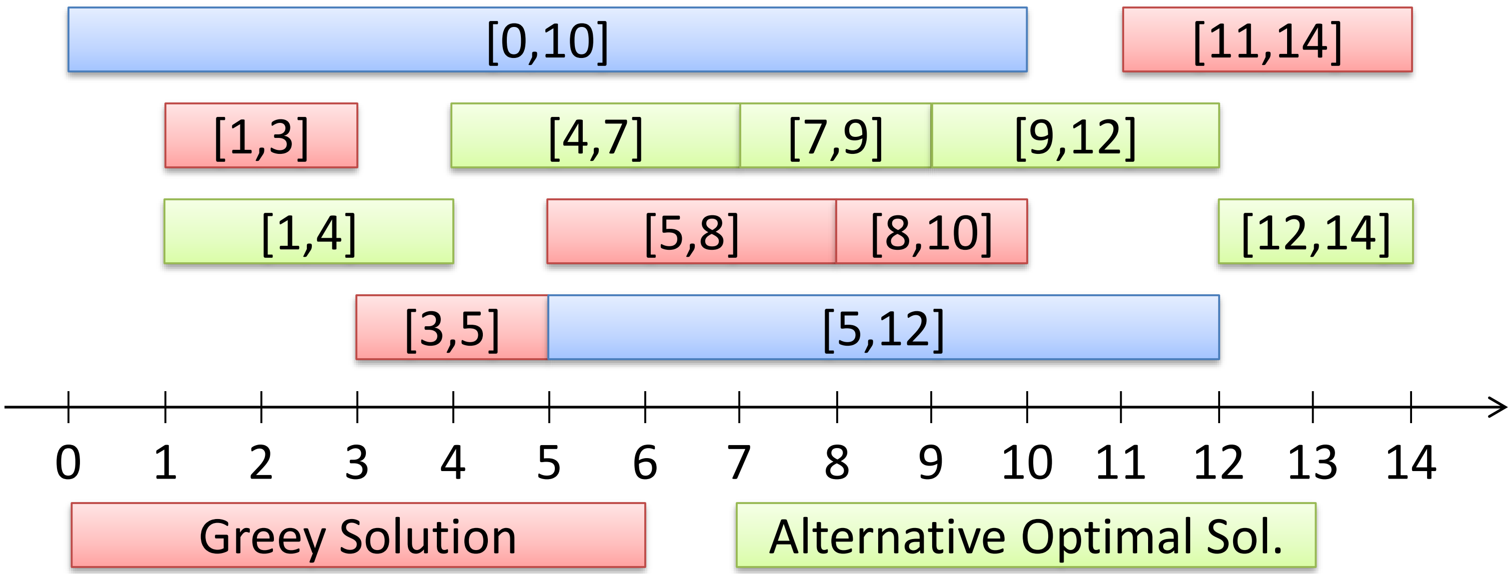
Choose available request with earliest finishing time:



R ≔ set of all requests; S ≔ empty set;
while R is not empty **do**
    choose $r \in R$ with smallest finishing time
    add r to S
    delete all requests from R that are not compatible with r
end // S is the solution

Earliest Finishing Time is Optimal

- Let O be the set of intervals of an optimal solution
- Can we show that $S = O$?
  - No…



- Show that $|S| = |O|$.

Greedy Stays Ahead

- Greedy solution S:

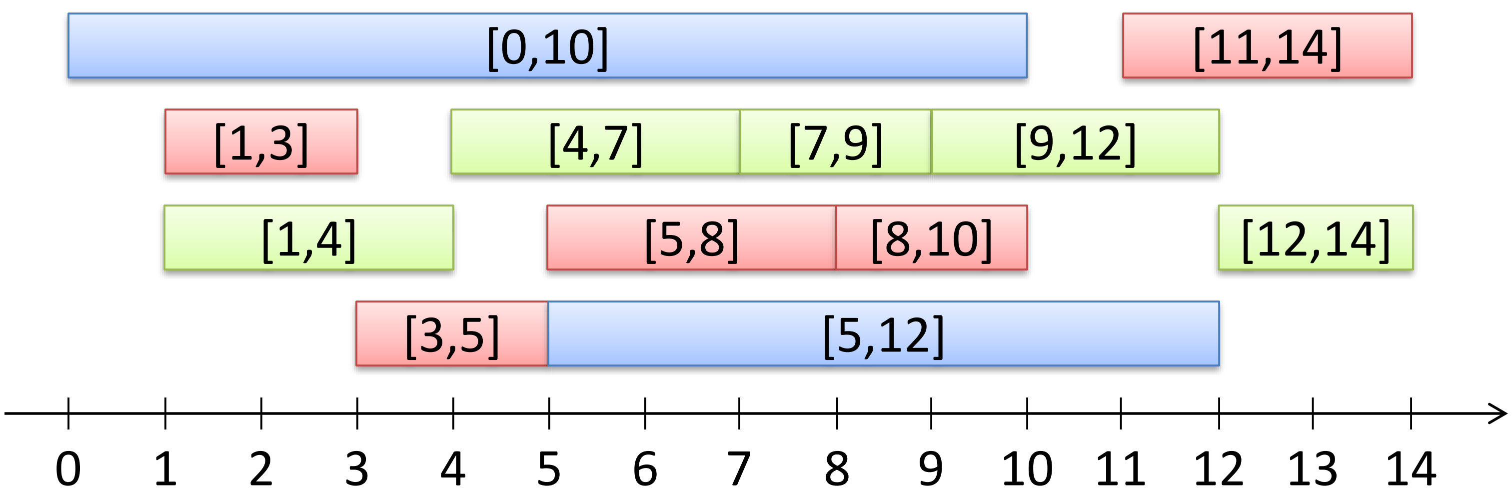
  $$[a_1, b_1], [a_2, b_2], \dots, [a_{|S|}, b_{|S|}], \qquad \text{where } b_i \leq a_{i+1}$$

- Some optimal solution O:

  $$[a_1^*, b_1^*], [a_2^*, b_2^*], \dots, [a_{|O|}^*, b_{|O|}^*], \qquad \text{where } b_i^* \leq a_{i+1}^*$$

- Definde $b_i := \infty$ for $i > |S|$ and $b_i^* := \infty$ for $i > |O|$

Claim: For all $i \geq 1$, $b_i \leq b_i^*$



Greedy Stays Ahead

Claim: For all $i \geq 1$, $b_i \leq b_i^*$

Proof (by induction on i):

Corollary: Earliest finishing time algorithm is optimal.

Weighted Interval Scheduling

Weighted version of the problem:

- Each interval has a weight
- Goal: Non-overlapping set with maximum total weight

Earliest finishing time greedy algorithm fails:

- Algorithm needs to look at weights
- Else, the selected sets could be the ones with smallest weight…

No simple greedy algorithm:

- We will see an algorithm using another design technique later.