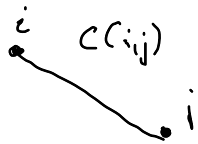
The traveling salesman problem (TSP)



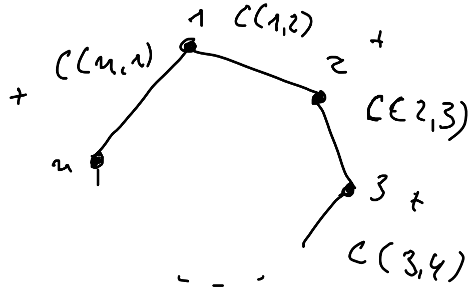
Given: n cities, costs $c(i,j)$ to travel from city i to city j

Goal: Find a cheapest round-trip route that visits each city exactly once and then returns to the starting city.

Formally: Find a permutation p of $\{1, 2, ..., n\}$, such that

$$c(\, p(1), p(2)\,) + \cdots + c(\, p(n-1), p(n)\,) + c(\, p(n), p(1)\,)$$

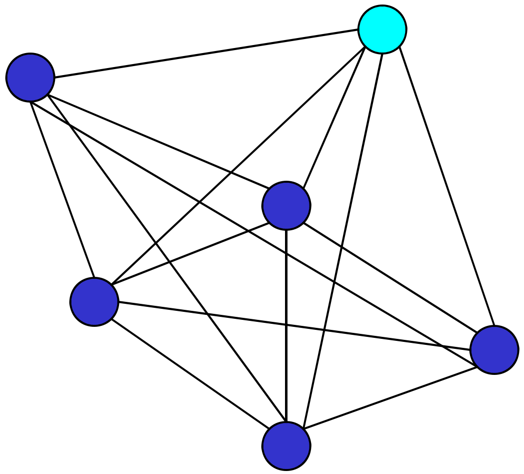
is minimzed.



The traveling salesman problem (TSP)

A greedy algorithm for solving the TSP

Starting from city 1, each time go to the nearest city not visited yet. Once all cities have been visited, return to the starting city 1.



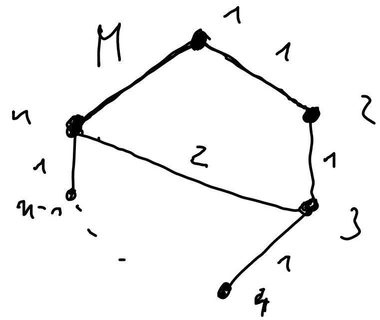
The traveling salesman problem (TSP)

Example

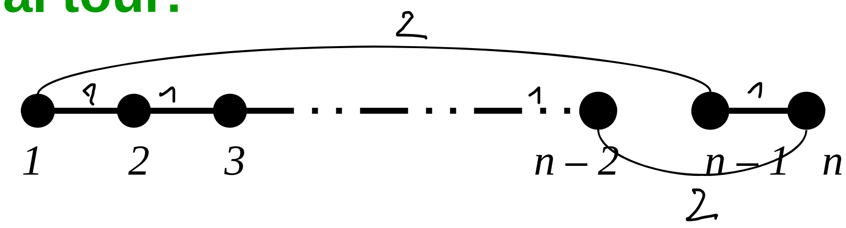
$c(i, i{+}1) = 1,\quad$ for $i = 1, ..., n - 1$

$c(n, 1) = M\quad$ (for some large number M)

$c(i,j) = 2,\quad$ otherwise

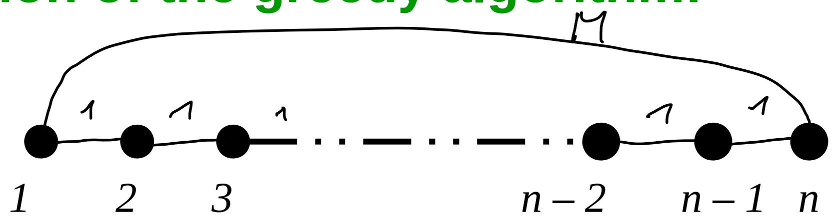


Optimal tour:



$OPT = n - 2 + 2 \cdot 2 = n + 2$

Solution of the greedy algorithm:



$GREEDY = n - 1 + M$

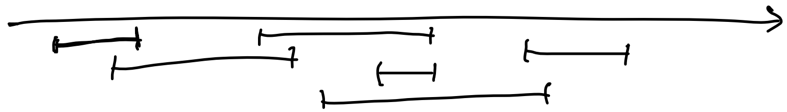
$$\frac{GREEDY}{OPT} = \frac{n-1+M}{n+2} \xrightarrow[n \to \infty]{} \infty$$

The activity-selection problem

Given:

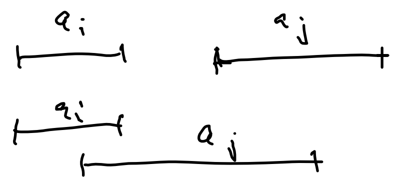
A set $S = \{a_1,...,a_n\}$ of n activities that wish to use a resource, e.g. a lecture hall.

activity a_i: start time s_i , finish time f_i $\quad a_i = [s_i, f_i)$



Activities a_i and a_j are compatible if

$$[s_i, f_i) \cap [s_j, f_j) = \varnothing$$



Goal:

Select a maximum-size subset of mutually compatible activities.

Assumption:

Activities are sorted in non-decreasing order of finish time:

$$\boxed{f_1 \leq f_2 \leq f_3 \leq ... \leq f_n}$$

Greedy strategy for solving the activity-selection problem:

> Always choose the activity with the earliest finish time that is compatible with all previously selected activities!

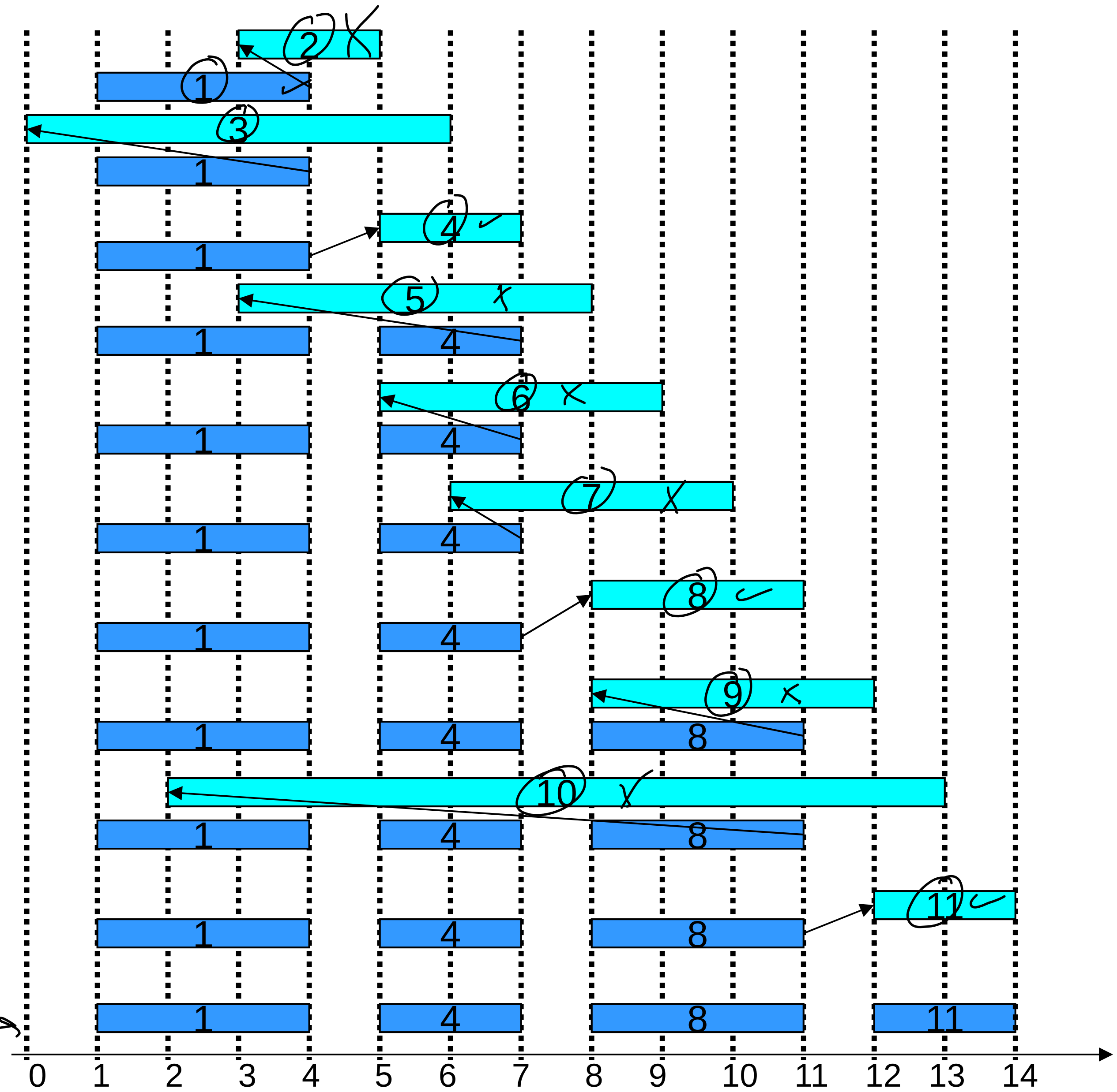
In particular, the activity chosen first is the one with the earliest finish time.

Theorem

The greedy strategy for selecting activities yields an optimal solution to the activity-selection problem.

The activity-selection problem

	s_i	f_i
1	1	4
2	~~3~~	~~5~~
3	~~0~~	~~6~~
4	5	7
5	~~3~~	~~8~~
6	~~5~~	~~9~~
7	~~6~~	~~10~~
8	8	11
9	~~8~~	~~12~~
10	~~2~~	~~13~~
11	12	14

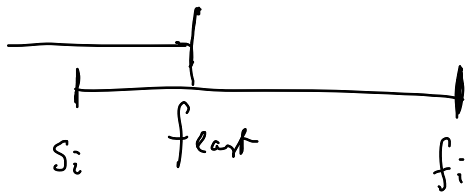


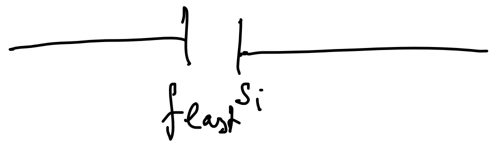
Activity-selection

Algorithm *Greedy-Activity-Selector*
Input: n activities represented by intervals $[s_i, f_i)$, $1 \le i \le n$, where $f_i \le f_{i+1}$
Output: a maximum-size set of mutually compatible activities

```
A_1 = {a_1}
last = 1          /* last indexes the activity added most recently */
for i = 2 to n do
   if s_i < f_last
     then A_i = A_{i-1}
     else  /* s_i ≥ f_last */
           A_i = A_{i-1} ∪ {a_i}
           last = i
return A_n
```

f_{last}





Running time: $O(n)$ provided that activities are sorted $f_1 \le \dots \le f_n$.

Optimality of the greedy algorithm

Theorem

The greedy algorithm yields an optimal solution.

Proof Show that for all $1 \leq i \leq n$ the following holds:

There exists an optimal solution A* with

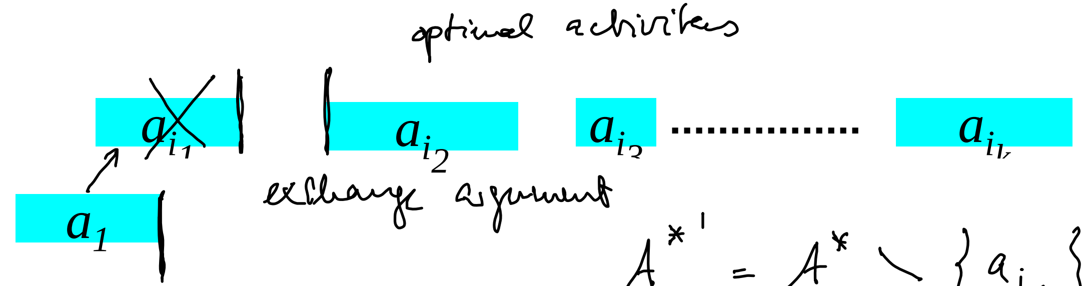
$$\boxed{A^* \cap \{a_1, \ldots, a_i\} = A_i}$$

set of activities greed has chosen among the sets $\{a_1, \ldots, a_i\}$

i = 1:

Let $A^* \subseteq \{a_1, \ldots, a_n\}$ be some optimal solution, $A^* = \{a_{i_1}, \ldots, a_{i_k}\}$

A^* =



$A^{*\prime} = A^* \setminus \{a_{i_1}\} \cup \{a_1\}$ is optimal

Optimality of the greedy algorithm

$i - 1 \rightarrow i$:

Let $A^* \subseteq \{a_1,\ldots,a_n\}$ be some optimal solution with $A^* \cap \{a_1,\ldots,a_{i-1}\} = A_{i-1}$.
Consider $R = A^* \setminus A_{i-1}$.

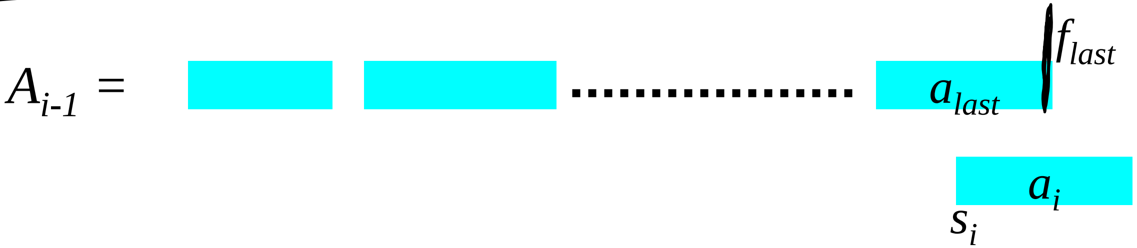
Observation:

R is an optimal solution to the problem of finding a maximum-size set of activities in $\{a_i,\ldots,a_n\}$ that are compatible with all activities in A_{i-1}.

Optimality of the greedy algorithm

$$A^* \cap \{a_1, \dots, a_{i-1}\} = A_{i-1} \subseteq A^*$$

Case 1: $s_i < f_{last}$



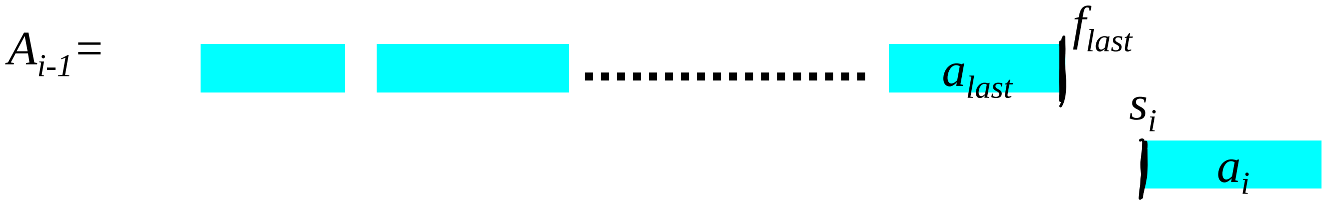
a_i is not compatible with A_{i-1} $\Rightarrow$ a_i is not in A_i $\Rightarrow$ $A_i = A_{i-1}$.

a_i is not contained in A^*

$\rightarrow$ $A^* \cap \{a_1,\dots,a_i\} = A_{i-1} = A_i$ ✔

Optimality of the greedy algorithm

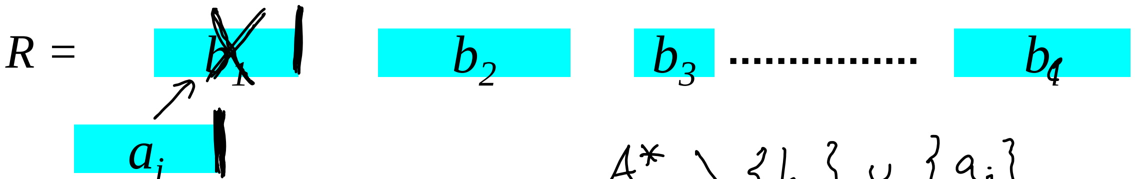
Case 2: $s_i \geq f_{last}$



a_i is compatible with A_{i-1} $\Rightarrow A_i = A_{i-1} \cup \{a_i\}$

$R = A^* \setminus A_{i-1}$

There is: $R \subseteq \{a_i, \ldots, a_n\}$



$= A^* \setminus \{b_1\} \cup \{a_i\}$

$B^* = \underline{A_{i-1}} \cup \underline{(R \setminus \{b_1\})} \cup \underline{\{a_i\}}$ is <u>optimal</u>

$B^* \cap \{a_1, \ldots, a_i\} = A_{i-1} \cup \{a_i\} = A_i$ ✓

Greedy algorithms

Greedy-choice property:

A globally optimal solution can be attained by a series of locally optimal (greedy) choices.

Optimal substructure property:

An optimal solution to the problem contains optimal solutions to its subproblems.

→ After making a locally optimal choice a new problem, analogous to the original one, arises.