The Greedy paradigm

1. Greedy - general ideas

- Used to solve optimization problems

- General strategy of Greedy:
  Small irreversible choices are made sequentially in a "greedy" manner to construct the solution
  At each step, greedy takes the best choice based on the currently known information
  (there are multiple ways of defining the "best choice" ...)

- Greedy usually leads to suboptimal solutions ... in some cases it actually leads to optimal solutions, solving the problem.

- One common way to prove that a Greedy algorithm actually solves the problem is the Exchange Argument.
  (If we use the Exchange Argument, we need to define two properties:
  * The Greedy Choice Property
  * The Optimal Substructure Property
)

2. Some examples of problems for which we can use Greedy (many more in the lecture ...)

2.1. The Minimum Coin Change Problem (Problema bancnotelor)

A version of the problem:

Input: $m \in \mathbb{N}$, m = sum to be paid

Output: numbers: $m_{500}, m_{200}, m_{100}, m_{50}, m_{20}, m_{10}, m_5, m_1$,
where m_i = the number of banknotes of i RON used, such that:

$S = \sum_{i \in \{500, 200, 100, 50, 20, 10, 5, 1\}} m_i$ is minimum and $\sum_{i \in \{500, 200, 100, 50, 20, 10, 5, 1\}} i \cdot m_i = m$

2.2. Interval Scheduling Problem (Problema selectiei activitatilor)

Input: $n \in \mathbb{N}$ (n = number of activities)
$s[0..n-1]$ – array retaining start time of each activity
$f[0..n-1]$ – array retaining finish time of each activity
such that $s[i] < f[i]$ for any $0 \le i \le n-1$
and f is in increasing order ($f[j] \le f[j+1]$, $\forall j = \overline{0, n-2}$)

(Obs: the ith activity is associated to $[s[i], f[i])$)

Output: $A \subseteq \{0, .., n-1\}$ such that: $|A|$ is maximal and
no two activities in A overlap ($[s[i], f[i]) \cap [s[j], f[j]) = \emptyset$, $\forall i, j \in A$)

An optimal strategy: choosing the activity with the earliest finish

3. Subproblems and Greedy Choice

"A subproblem is an instance of the problem, smaller-sized then the original instance and that can take part in solving the initial instance, based on some hierarchical structure with multiple levels. "

For the Interval Scheduling Problem:
Subproblem S: an instance of the problem in which we have to choose a subset of the activities from a subset of {0,1,..., n -1}
Greedy Choice x: at one step, we choose the activity in S that finishes earliest

(In this case, the greedy choice leads to optimal solution(s))

e.g.
Initial subproblem $\{0, 1, \dots, n-1\}$
Greedy Choice (choose activity 2)
Subproblem $\{0, 1, 3, \dots, n-1\}$
Greedy choice (choose activity 0)
Subproblem $\{1, 3, \dots, n-1\}$
...

The Exchange Argument for proving Greedy optimality

4. Proving that a Greedy Algorithm is optimal using the Exchange Argument

We need to define and prove for our problem:

The Greedy Choice Property: "There is an optimal solution that contains the Greedy Choice"

Let S be a subproblem and $x \in S$ a greedy choice

We want to show that $\exists A$, such that $\begin{cases} A \text{ is an optimal solution for } S \\ x \in A \end{cases}$

Proof: Let B an optimal solution for S, such that $x \notin B$.

Using B, we define A, such that $x \in A$

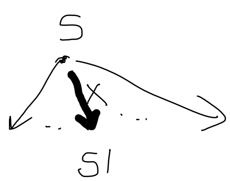
We need to show that A is also a solution for S and that A is optimal

(depending on the problem B may actually exist as an optimal solution or not)

This part of proof differs from problem to problem

The Optimal Substructure Property: "Let A be an optimal solution for the subproblem S. The subsolutions of A are optimal for the subproblems of S "

A



$B = A \setminus \{x\}$

Let S a subproblem (such that there are still choices to be made)

Let $x \in S$ the greedy choice.

Let A an optimal solution for S that contains x $(x \in A)$.

We want to show that $B = A \setminus \{x\}$ is an optimal solution for S' (the subproblem that we obtain by making the greedy choice x from S)

Proof:

We suppose there is a better solution than B (let it be C) for subproblem S'

Then, $C \cup \{x\}$ should be a better solution for S than A $\Rightarrow$ we contradict the optimality of A!

$\Downarrow$

there is no such solution C

$\Downarrow$

B is the optimal solution for S'