COMP 3711

Design and Analysis of Algorithms

Greedy Algorithms

Arman Haghighi

Review: Greedy algorithm

- A greedy algorithm is an algorithm that produces the solution one step at a time by choosing the **"locally best"** option at **each step** **(at the moment)**.

  + Simple
  + Efficient

  - Not always easy to design
  - Not always easy to prove

- **Question:** Can Greedy approach always provides the globally optimal solution?
- **Answer:** **No, not always!**

If it can generate the globally optimal solution you **must** prove why

- **Question:** How to show a greedy algorithm does not gives the globally optimal solution?

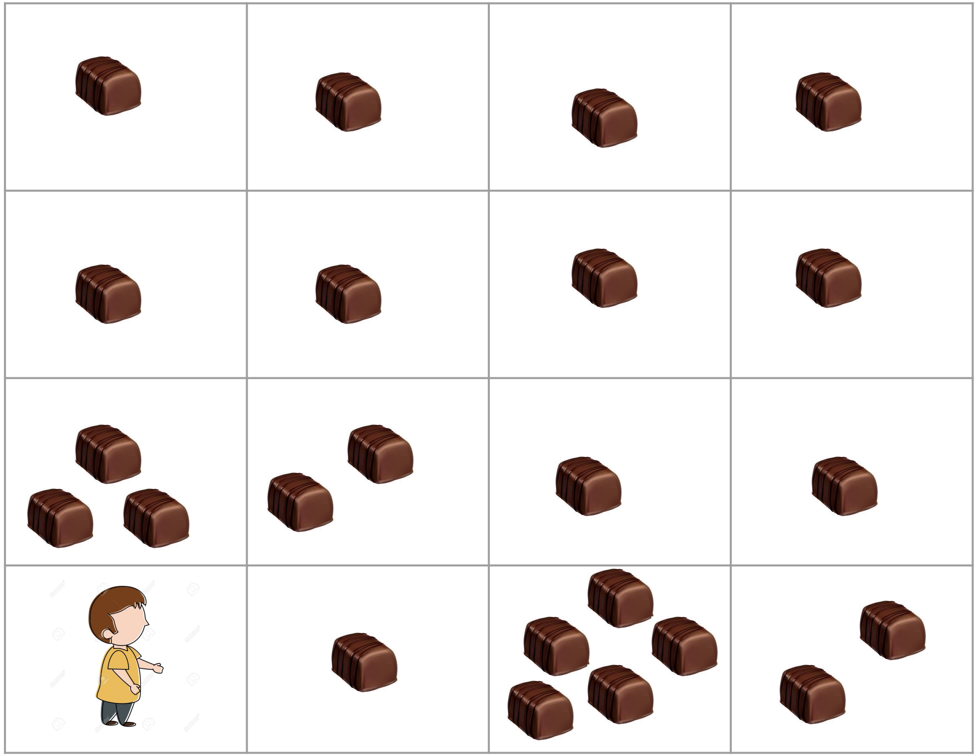
Provide a counterexample such that the globally optimal solution is better than the greedy algorithm output!

- **Question:** If a greedy approach does not provide globally optimal solution does it mean it is useless? **Maybe no! It might be a good approximate answer!**

Example 1: Bob and Chocolates!

- Bob Can go either **Up** or **Right**
- Bob wants to reach top-right cell
- Bob eats all the chocolates on the way

Question: Find the path with maximum chocolate



Greedy approach: at each step choose the side (between **Right** and **Up**) with <u>maximum</u> chocolate.

Question: Does this approach finds the best globally optimal solution?

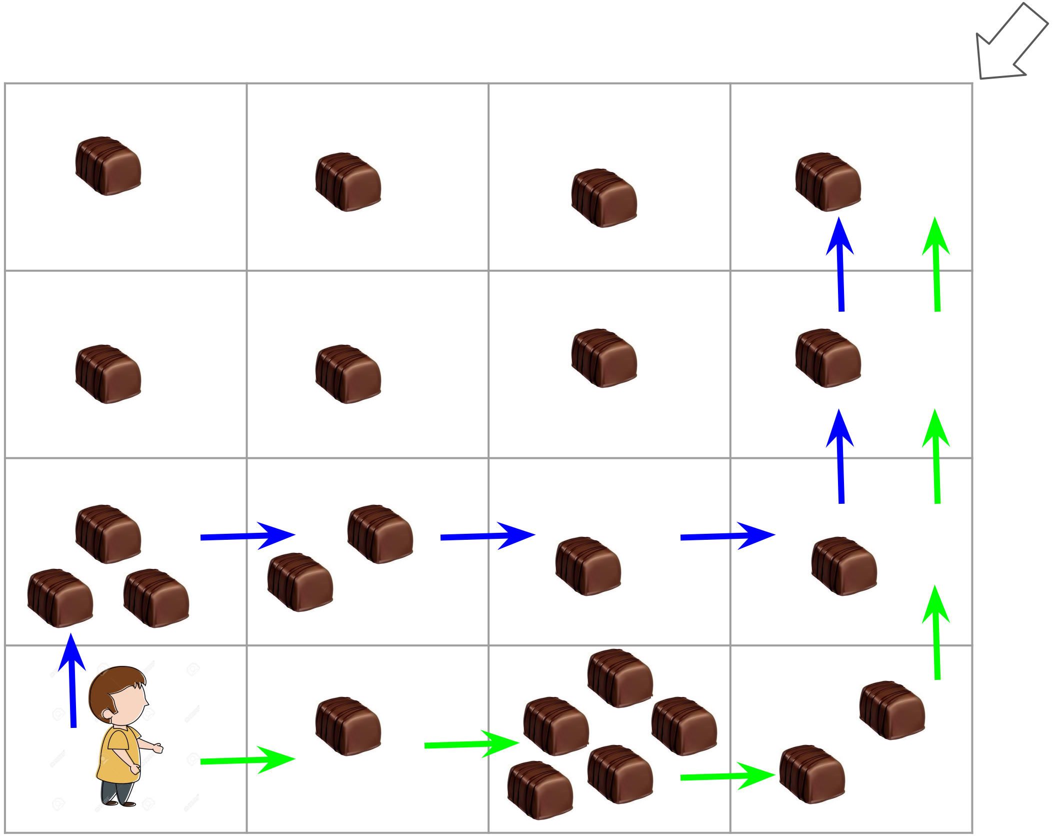
Example 1(cont'd): Bob and Chocolates!

Greedy approach: at each step choose the side (between **Right** and **Up**) with maximum number of chocolates.

Greedy approach:
U R R R U U
 # chocolates: 9

Globally optimal solution:
R R R U U U
 # chocolates: 11

This example shows that this approach does **not** find the globally optimal solution!



Example 2

A *unit-length closed interval* on the real line is an interval $[x, 1 + x]$. Describe an $O(n)$ algorithm that, given input set $X = \{x_1, x_2, \ldots, x_n\}$, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct. You should assume that $x_1 < x_2 < \cdots < x_n$.

As an example the points above are given on a line and you are given the length of a 1-unit interval. Show how to place a minimum number of such intervals to cover the points

Example 2: Solution

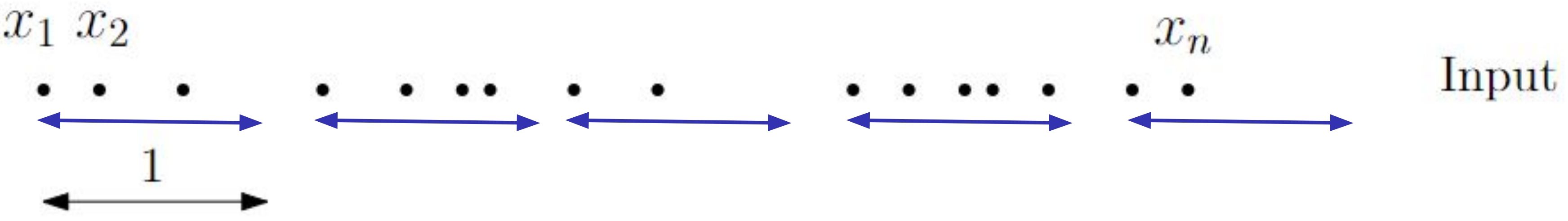
(a) *Set* $x = x_1$.

(b) *Walk through the points in increasing order until finding the first* j *such that* $x_j > x + 1$. *(if no such point then stop)*

(c) *Output* $[x, 1 + x]$

(d) *If there was no such* j *in (b) then stop. Otherwise, set* $x = x_j$.

(e) *Go to Step (b)*



Example 2: Correctness

How can we prove correctness of a greedy approach?

High-level idea: Assume the optimal solution is different from the greedy solution. Show that the greedy solution can not be worse than the optimal solution.

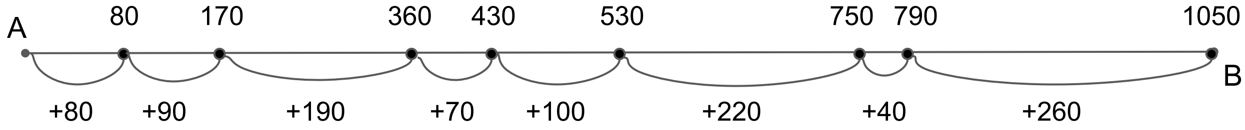
Hint for this question: Use induction on number of points

Example 3

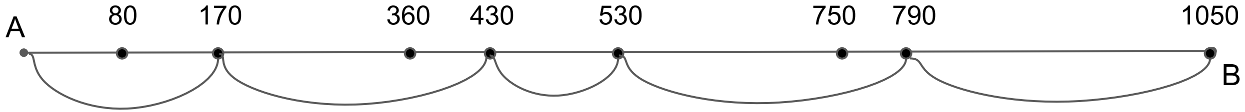
(CLRS–16.2-4) Professor Midas drives an automobile from Newark to Reno along Interstate 80. His car's gas tank, when full, holds enough gas to travel m miles, and his map gives the distance between gas stations on his route. The professor wishes to make as few gas stops as possible along the way. Give an efficient method by which Professor Midas can determine at which gas stations he should stop and prove that your algorithm yields an optimal solution.

Example 3: Solution

Example: Distance with full tank: 300 km



- **Approach 1:** Fill your tank as soon as you see a gas station!
  # stops: 7
  Definitely a bad approach! We can do better!
- **Approach 2:** Refill at the farthest reachable gas station!



 # stops: 5

This is the globally optimal solution! But how do we prove it?

Example 3: Correctness

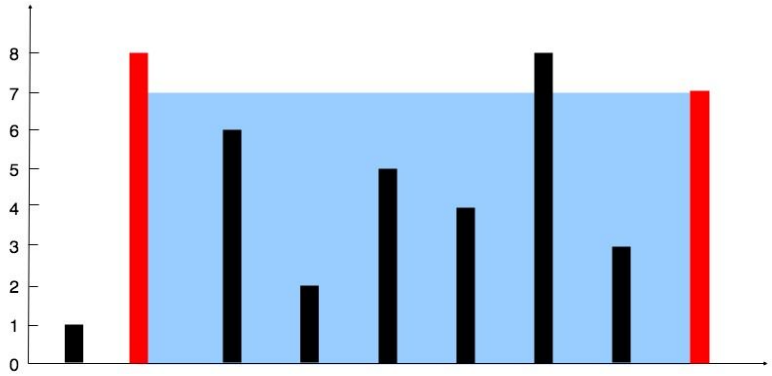
Hint for this question: Use induction on the number of cities

Example 4: Container with most water

Question: You are given an integer array **height** of length **n**. There are **n** vertical lines drawn such that the two endpoints of the **i_th** line are (i,0) and (i,height(i)).

Find two lines that together with the x-axis form a container, such that the container contains the most water and return ***the maximum amount of water a container can store*** . (you may not slant the container)

Example:



Globally optimal solution : max area of water in container = 49

- **Question 1:** How to find these two lines?
- **Question 2:** How to prove it is the globally optimal solution?

Example 5: Distributing Candies

Question: Consider a given array with **N** integers, where each element represents the ratings of **N** children standing in a line. We have to distribute **minimum** candies in to these children in such a way that:

- Children with higher ratings have more candies than their neighbors.
- Each child must have at least one candy.

Example:

Rating:	6	4	5	2	1

#candy:	2	1	3	2	1

Globally optimal solution : Total number of candies = 9

- **Question 1:** How to distribute candies?
- **Question 2:** How to prove it is the globally optimal solution?