CSI - 3105 Design & Analysis of Algorithms
Course 9

Jean-Lou De Carufel

Fall 2019

Greedy Algorithms

A *greedy algorithm* arrives at a solution by making a sequence of choices, each of which simply looks the best at the moment.

The hope is that locally-optimal choices will lead to a globally-optimal solution.

Making Change

How can a given amount of money be made with the least number of coins of a given denominations? (Assume that you have an unlimited amount of coins of each denomination (!))

Making Change

How can a given amount of money be made with the least number of coins of a given denominations? (Assume that you have an unlimited amount of coins of each denomination (!))

Algorithm *MakingChange*(x)

$Coins = \{5, 10, 25, 100, 200\}$
$Sum = 0$
while $Sum \neq x$ **do**
  Let $c \in Coins$ be the largest denomination such that $Sum + c \leq x$.
  if there is no such denomination **then**
    return "No Solution"
  end if
  Add c to the change that will be returned.
  $Sum = Sum + c$
end while

Question: In the Republik of Informatik, the only denominations available are 1, 4 and 6 cents. Does *MakingChange* produce an optimal solution for all possible inputs?

$X = 8$

Section 4.2: The $[0, 1]$-Knapsack Problem

Find a greedy algorithm to solve the following problem.

input:

- n objects such that object i $(1 \leq i \leq n)$ has a positive *value* v_i and a positive *weight* w_i.
- A maximum weight W.

output: A vector $X = (x_1, x_2, ..., x_n)$ such that

- $0 \leq x_i \leq 1$
- The total value $x_1 v_1 + x_2 v_2 + ... + x_n v_n$ is maximized.
- The total weight is not too heavy: $x_1 w_1 + x_2 w_2 + ... + x_n w_n \leq W$.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$
- $x_5 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$
- $x_5 = 1$
- $x_4 = 0.5$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$
- $x_5 = 1$
- $x_4 = 0.5$
- Total value: 146

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$
- $x_3 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$
- $x_3 = 1$
- $x_4 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$
- $x_3 = 1$
- $x_4 = 1$
- Total value: 156

This proves that greedy choices with respect to values **does not** lead to an optimal solution in general.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$
- Total value: 164

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$
- Total value: 164

This proves that greedy choices with respect to weights **does not** lead to an optimal solution in general.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$
- Total value: 164

This proves that greedy choices with respect to weights **does not** lead to an optimal solution in general.
What about greedy choices with respect to values per unit of weight.
Is this optimal ?!

Lemma

If the objects are selected in order of decreasing $\frac{v_i}{w_i}$, then the greedy choice finds an optimal solution.

Lemma

If the objects are selected in order of decreasing $\frac{v_i}{w_i}$, *then the greedy choice finds an optimal solution.*

PROOF:

So here is how to solve the problem:

Step 1 : Compute all the $\frac{v_i}{w_i}$'s.

Step 2 : Sort the objects in decreasing order of $\frac{v_i}{w_i}$ (use Merge Sort).

Step 3 : Build the solution by applying the greedy choice with respect to decreasing order of $\frac{v_i}{w_i}$.

How much time does it take?

So here is how to solve the problem:

Step 1 : Compute all the $\frac{v_i}{w_i}$'s.

Step 2 : Sort the objects in decreasing order of $\frac{v_i}{w_i}$ (use Merge Sort).

Step 3 : Build the solution by applying the greedy choice with respect to decreasing order of $\frac{v_i}{w_i}$.

How much time does it take?

Step 1 : $O(n)$ time

Step 2 : $O(n \log(n))$ time

Step 3 : $O(n)$ time

Total time : $O(n) + O(n \log(n)) + O(n) = O(n \log(n))$

Is there a faster algorithm?

Is there a faster algorithm?

What do we do if x_i must satisfy $x_i \in \{0, 1\}$ instead of $x_i \in [0, 1]$?

Is there a faster algorithm?

What do we do if x_i must satisfy $x_i \in \{0, 1\}$ instead of $x_i \in [0, 1]$?

Refer to Chapter 5.

Minimum Spanning Tree

We are given a graph $G = (V, E)$ that is undirected and connected. Each edge $\{u, v\} \in E$ has a weight $wt(u, v)$.

We want to compute a subgraph G' of G such that

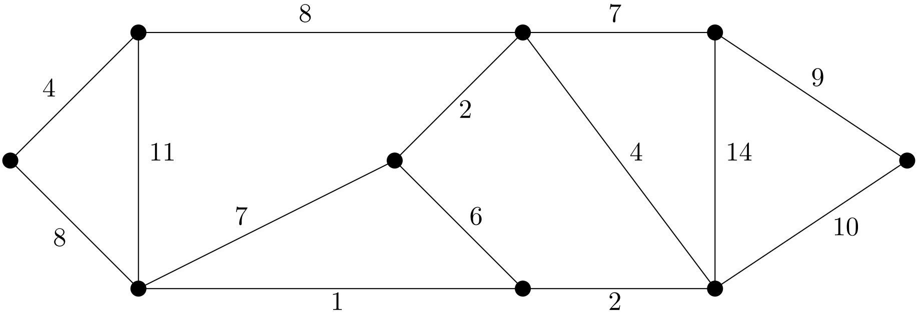
- The vertex set of G' is V,
- G' is connected,
- and $weight(G')$ is minimum, where

$$weight(G') = \text{sum of weights of edges in } G'.$$

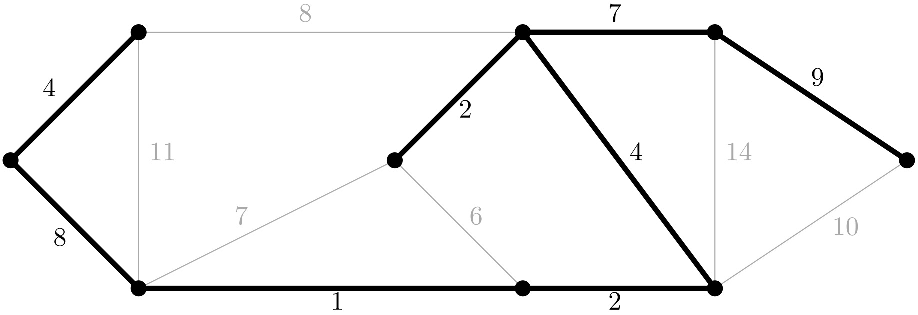
We can prove that G' must be a tree (connected and no cycles). Do you see why?

G' is called a *Minimum Spanning Tree of G* (MST of G).

Example:



Example:



Fundamental Lemma

Lemma

Let $G = (V, E)$ be an undirected and connected graph, where each edge $\{u, v\} \in E$ has a weight $wt(u, v)$.

Split V into A and B. Let $\{u, v\} \in E$ be a shortest edge connecting A and B. Then there is an MST of G that contains $\{u, v\}$.