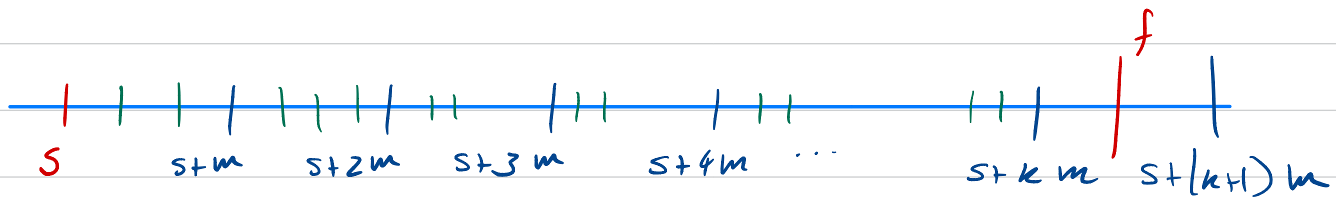
Práctica 8

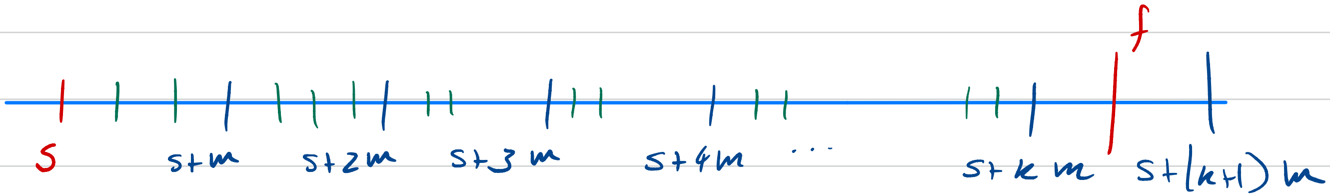
16.2-4



- A cada m milhas o Prof. Gekko deve reabastecer-se de água.
  Os postos de abastecimento estão indicados em verde.

- Problema: Em q postos de abastecimento deve o Prof. Gekko parar de modo a minimizar o nº de paragens?

16.2-4

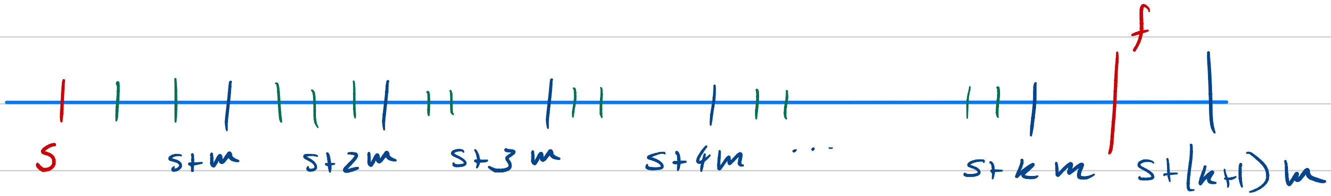


• A cada m milhas o Prof. Gekko deve reabastecer-se de água.
Os postos de abastecimento estão indicados em verde.

• Problema: Em q̃ postos de abastecimento deve o Prof. Gekko parar de modo a minimizar o nº de paragens?

• Formalização:
- Input: • m – distância máxima que o Prof. Gekko pode percorrer sem abastecer
  • $\vec{x}[1..n]$ – distâncias dos postos de abastecimento
  • s e f – distâncias de início e de fim
- Output – $\vec{j}[1..j]$ – distâncias dos postos de abastecimento escolhidos

16.2-4



- Input: · m - distância máxima que o Prof. Gekko pode percorrer sem abastecer
  · $\vec{x}[1..n]$ - distâncias dos postos de abastecimento
  · s e f - distâncias de início e de fim

- Output - $\vec{y}[1..k]$ - distâncias dos postos de abastecimento escolhidos

Escolha Greedy

$$y^* = \max\{\vec{x}[k] \mid \vec{x}[k] \le m\}$$

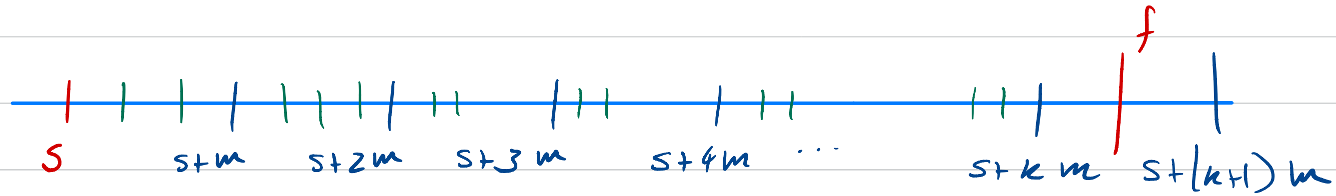
Subproblema:

$(\vec{x}, m, s', f)$ onde $s' = y^*$

Solução Óptima

$$y^* \underbrace{::}_{\text{prepended}} \text{WaterFill}(\vec{x}, m, s', f)$$

16.2-4



Input: $(\vec{x}, m, s, f)$

Output: $\vec{y} \subseteq \vec{x}$

Escolha Greedy

$$y^* = \max \{ \vec{x}[k] \mid \vec{x}[k] \leq m \}$$

- Há que provar que y^* faz parte de uma solução óptima.
  Seja $\vec{y}$ uma solução óptima; temos dois casos a considerar:
  - $y^* \in \vec{y}$ (nada provar)
  - $y^* \notin \vec{y}$

- Suponhamos que $y^* \notin \vec{y}$. Temos de construir a partir de $\vec{y}$ uma outra solução óptima $\vec{y}'$ tal que $y^* \in \vec{y}'$.

16.2-4

Escolha Greedy

$$y^* = \max\{\vec{x}[k] \mid \vec{x}[k] \leq m\}$$

- Há que provar que y^* faz parte de uma solução óptima.
Seja $\vec{y}$ uma solução óptima; temos dois casos a considerar:
  - $y^* \in \vec{y}$ (nada provar)
  - $y^* \notin \vec{y}$

- Suponhamos que $y^* \notin \vec{y}$. Temos de construir a partir de $\vec{y}$ uma outra solução óptima $\vec{y}'$ tal que $y^* \in \vec{y}'$.
Para tal basta trocar o 1º elemento de $\vec{y}$ por y^*.
Formalmente:

$$\vec{y} = \langle y_1, y_2, \dots, y_j \rangle$$
$$\Downarrow$$
$$\vec{y}' = \langle y^*, y_2, \dots, y_j \rangle$$

16.2-4

- Suponhamos que $y^* \notin \vec{y}$. Temos de construir a partir de $\vec{y}$ uma outra solução ótima $\vec{y}'$ tal que $y^* \in \vec{y}'$.
  Para tal basta trocar o 1º elemento de $\vec{y}$ por y^*.
  Formalmente:

$$\vec{y} = \langle y_1, y_2, \cdots, y_j \rangle$$
$$\Downarrow$$
$$\vec{y}' = \langle y^*, y_2, \cdots, y_j \rangle$$

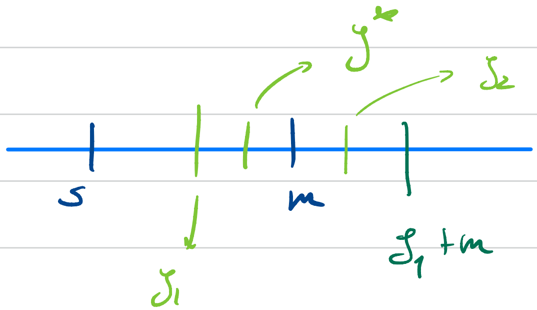
- Há ainda de provar que:

  - $|\vec{y}| = |\vec{y}'|$ ✓ (por construção)

  - $\vec{y}'$ é solução factível: $\forall\, 1 \leq i \leq l-1.\ \vec{y}'[i+1] \leq \vec{y}'[i] + m$

$\boxed{i \geq 2}$ $\vec{y}'[i+1] = \vec{y}[i+1] \leq \vec{y}[i] + m = \vec{y}'[i] + m$ porque $\vec{y}$ é factível

$\boxed{i = 1}$ $\vec{y}'[2] = \vec{y}[2] \geq \vec{y}[1] + m \geq y^* + m = \vec{y}'[1] + m$



16.2-4

WaterFill($\vec{x}$, m, s, f)

let L = emptyList();
let next = s + m;

for i = 1 to $\vec{x}$.size()
    if ($\vec{x}$[i] > next)
        L.add($\vec{x}$[i-1])
        next = $\vec{x}$[i-1] + m

return L

$y^* = \max \{ \vec{x}[k] \mid \vec{x}[k] \leq m \}$

Subproblema:
($\vec{x}$, m, s', f) onde $s' = y^*$

Solução Óptima: $y^* :: \text{WaterFill}(\vec{x}, m, s', f)$

prepended

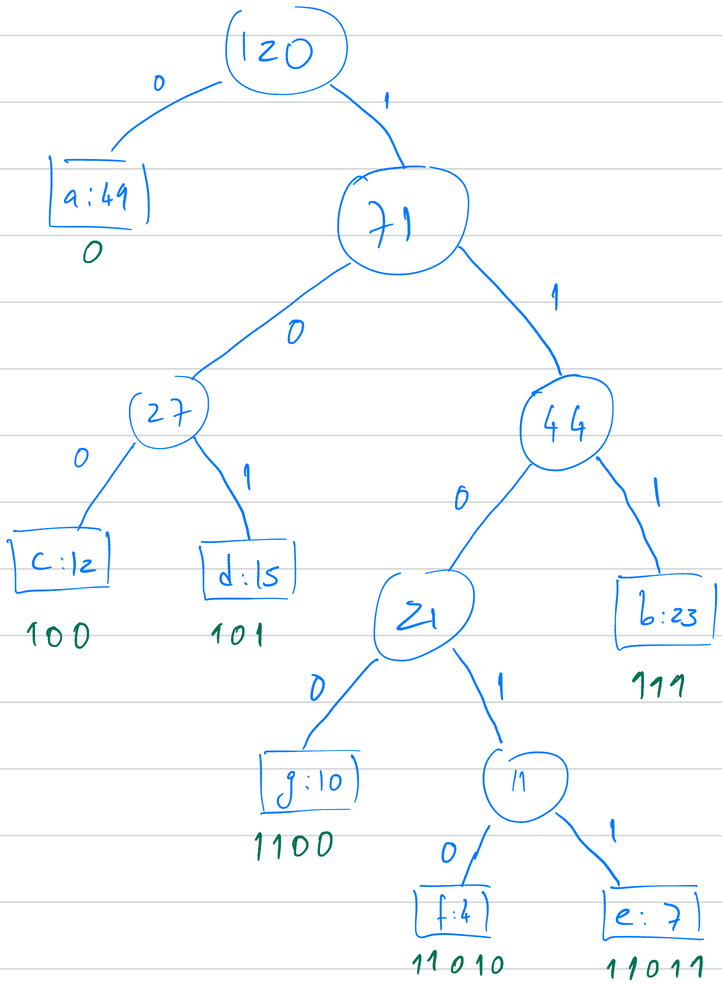
Complexidade: $\Theta(n)$

T2 08/09 II.3

a: 49, b: 23, c: 12, d: 15, e: 7, f: 4, g: 10

TZ 08/09 II.3

a:49, b:23, c:12, d:15, e:7, f:4, g:10



Rz 08/09 II.3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a_i	00	06	10	13	17	20	23	25	28	31	33	36	38	41	43	50	53
d_i	5	5	3	6	1	3	2	6	2	4	5	3	3	4	4	6	2

Rz 08/09 II.3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a_i	00	06	10	13	17	20	23	25	28	31	33	36	38	41	43	50	53
d_i	5	5	3	6	1	3	2	6	2	4	5	3	3	4	4	6	2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a_i	00	06	10	13	17	20	23	25	28	31	33	36	38	41	43	50	53
d_i	5	5	3	6	1	3	2	6	2	4	5	3	3	4	4	6	2
f_i	5	11	13	19	18	23	25	31	30	35	38	39	41	45	47	56	55

	1	2	3	5	4	6	7	9	8	10	11	12	13	14	15	17	16
a_i	00	06	10	17	13	20	23	28	25	31	33	36	38	41	43	53	50
d_i	5	5	3	1	6	3	2	2	6	4	5	3	3	4	4	2	6
f_i	5	11	13	18	19	23	25	30	31	35	38	39	41	45	47	55	56

$X = \{1, 2, 5, 6, 7, 9, 10, 12, 14, 17\}$

16.2.5

- Input: $\vec{x}[1..n]$ n pontos na recta real
- Output: $\vec{y}[1..k]$ k pontos na recta real tais que:

  $\forall 1 \le i \le n.\ \exists 1 \le j \le k.\ \vec{x}[i] \in [\vec{y}[j], \vec{y}[j]+1]$

  Queremos encontrar uma solução $\vec{y}$ tão pequena quanto possível.

16.2.5

Find Intervals($\vec{x}$)

- Input: $\vec{x}[1..n]$ n pontos na recta real
- Output: $\vec{y}[1..k]$ k pontos na recta real tais que:

$\forall 1 \le i \le n.\ \exists 1 \le j \le k.\ \vec{x}[i] \in [\vec{y}[j], \vec{y}[j]+1]$

Queremos encontrar uma solução $\vec{y}$ tão pequena quanto possível.

Escolha Greedy

$$y^* = \min\{\vec{x}[i] \mid 1 \le i \le n\}$$

Subproblema: Find Intervals($\vec{x}'$)

com $\vec{x}'$ vector q contém todos os elementos de $\vec{x}$ maiores q y^*+1

16.2.5

Escolha Greedy

$$y^* = \min \{ \vec{x}[i] \mid 1 \leq i \leq n \}$$

Subproblema: Find Intervals $(\vec{x}')$
com $\vec{x}'$ vector q contém todos os elementos de $\vec{x}$ maiores q $y^* + 1$

- Há que provar que y^* faz parte de uma solução óptima.
  Seja $\vec{J}$ uma solução óptima; temos dois casos a considerar:
  - $y^* \in \vec{J}$ (nada provar)
  - $y^* \notin \vec{J}$

- Suponhamos que $y^* \notin \vec{J}$. Temos de construir a partir de $\vec{J}$ uma outra solução óptima $\vec{J}'$ tal que $y^* \in \vec{J}'$.
  Para tal basta trocar o 1º elemento de $\vec{J}$ por y^*.
  Formalmente:

$$\vec{J} = \langle J_1, J_2, \cdots, J_j \rangle$$

$$\Downarrow$$

$$\vec{J}' = \langle y^*, J_2, \cdots, J_j \rangle$$

16.2.5

- Há ainda q provar q:

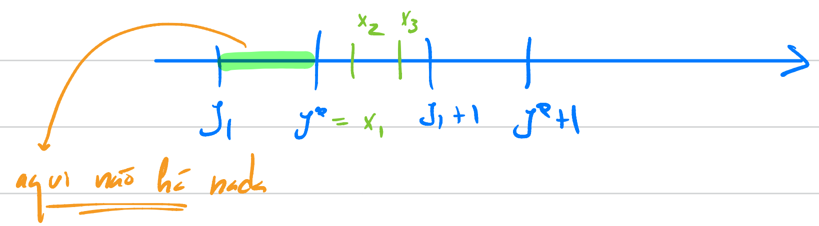
  - $|\vec{y}| = |\vec{y}'|$ ✓ (por construção)

  - $\vec{y}'$ é uma solução factível:

$$\forall 1 \leq i \leq n.\ \exists 1 \leq j \leq k.\ \ \vec{x}[i] \in [\vec{y}[j], \vec{y}[j]+1]$$

- Trocámos o intervalo $[y_1, y_1+1]$ pelo intervalo $[y^*, y^*+1]$

- Como $[y_1, y_1+1]$ tem de incluir o elemento y^*, concluímos que: $y_1 < y^*$
  Além disso, o intervalo $[x_1, y^*[$ não inclui nenhum elemento de $\vec{x}$.
  Concluímos portanto que ao trocar $[y_1, y_1+1]$ por $[y^*, y^*+1]$ não deixamos de incluir nenhum ponto de $\vec{x}$.



16.2.5

```
ComputeIntervals(x⃗)
  let L = emptyList();
  let last = x⃗[1]
  for i=2 to x⃗.size()
    if (x⃗[i] > last+1)
      L.add(x⃗[i])
      last := x⃗[i]
  return L
```

$\Theta(n)$

onde $n = \vec{x}.\text{size}()$

Tz 14/15 II.b

1. Ordenamos os n objectos por ordem decrescente ($\leq$) de valor por unidade de peso — $O(n \lg n)$

2. Seguindo a ordem estabelecida, vamos colocando 50% de cada objecto em cada uma das caixas

3. Seja i o índice do objecto tal que $p_i/2$ excede a capacidade residual das duas caixas.
Quando atingimos o objecto i, colocamos em cada caixa apenas apenas a fracção de i suficiente para completar a capacidade da caixa.

— $O(n)$

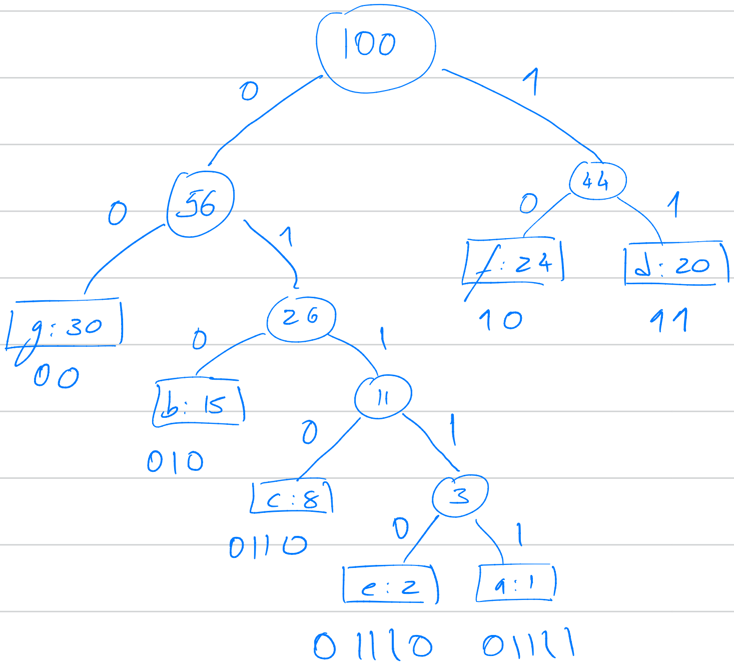
Total com Ord.: $O(n \lg n)$

<u>ℓ2 14/15 II.b</u>

$\boxed{a:1}$ $\boxed{b:15}$ $\boxed{c:8}$ $\boxed{d:20}$ $\boxed{e:2}$ $\boxed{f:24}$ $\boxed{g:30}$

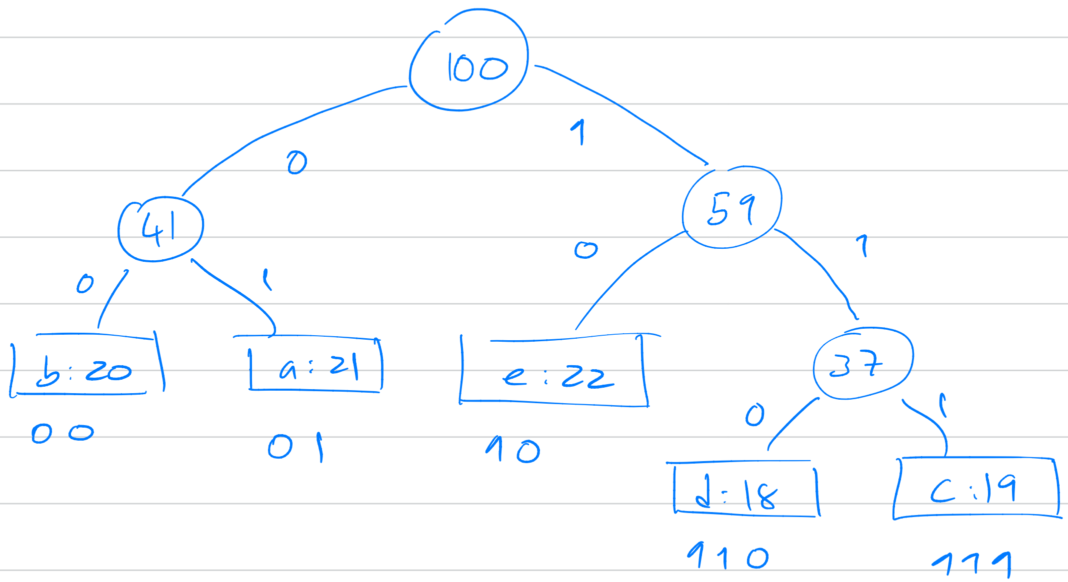
l2 14/15 II.b

a : 1 | b : 15 | c : 8 | d : 20 | e : 2 | f : 24 | g : 30



R2 16/17 I.a

a : 21 b : 20 c : 19 d : 18 e : 22



R2 15/16 - II.a

Input: $\underbrace{\langle d_1, \dots, d_n \rangle}_{\text{denominações por ordem crescente}}, \overset{\text{valor do troco}}{K}$

Formalização: $\min \sum_{i=1}^{n} x_i$

$\sum_{i=1}^{n} x_i . d_i = K$

Importante: $\boxed{\forall 1 \leq i < n . \; d_{i+1} \geq 2 \times d_i}$

R2 15/16 - II.a

Input: $\underbrace{\langle d_1, \ldots, d_n \rangle}_{\text{denominações por ordem crescente}}, \overset{\text{valor do troco}}{K}$

Formalização: $\min \sum_{i=1}^{n} x_i$

$\sum_{i=1}^{n} x_i \cdot d_i = K$

Importante: $\boxed{\forall_{1 \leq i < n} \cdot \; d_{i+1} \geq 2 \times d_i}$

Escolha Greedy

$$x_n = \left\lfloor \frac{k}{d_n} \right\rfloor$$

Subproblema: $\left(\langle d_1, \ldots, d_{n-1} \rangle, k - \left\lfloor k/d_n \right\rfloor \right)$

R2 15/16 - II.a

Input: $\underbrace{\langle d_1, \dots, d_n \rangle}_{\text{denominações por ordem crescente}}, \; k$ ← valor do troco

Formalização: $\min \sum_{i=1}^{n} x_i$

$\sum_{i=1}^{n} x_i \cdot d_i = k$

$\forall_{1 \leq i < n} . \; d_{i+1} \geq 2 \times d_i$

Escolha Greedy

$$x_n = \left\lfloor \frac{k}{d_n} \right\rfloor$$

Subproblema: $\left(\langle d_1, \dots, d_{n-1} \rangle, \; k - \left\lfloor k/d_n \right\rfloor \right)$

Provar que: x é ótimo $\longrightarrow x_n = \left\lfloor k/d_n \right\rfloor$

- Provamos o resultado por contradição.
  Suponhamos que x é ótimo e $x_n \neq \left\lfloor k/d_n \right\rfloor$.
  Há dois casos a considerar:

(I) $x_n > \left\lfloor k/d_n \right\rfloor$

$x_n > \left\lfloor \frac{k}{d_i} \right\rfloor \Rightarrow x_n \cdot d_n > k$ ↯

↓

$\underline{x \text{ não é solução}}$

(II) $x_n < \left\lfloor k/d_n \right\rfloor$

$\Longleftrightarrow x_n \leq k/d_n - 1$

$\Longleftrightarrow x_n \cdot d_n \leq k - d_n$

- Isto significa que há pelo menos d_n unidades q têm de ser pagas com denominações inferiores a d_n. Ora, usando a denominação d_n eu posso pagar d_n unidades com uma única moeda. Usando uma denominação inferior a d_n, tenho de utilizar pelo menos <u>duas</u> moedas; de onde concluímos a contradição

↯

Rz 15/16 - II. a

ComputeChange($\vec{d}$, k)
  let $\vec{x}$ be a new array of size $n = \vec{d}.size$
  for $i = 1$ to n
    $\vec{x}[i] = \lfloor k / \vec{d}[i] \rfloor$
    $k = k - \vec{x}[i] * \vec{d}[i]$
  return $\vec{x}$

RZ 15/16 - II.a

Input: $\underbrace{\langle d_1, \dots, d_n \rangle}_{\text{denominações por ordem crescente}}, K$ → valor do troco

Formalização:

$$\min \sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} x_i \cdot d_i = K$$

$$\forall_{1 \leq i < n} . \; d_{i+1} \geq 2 \times d_i$$

Escolha Greedy

$$x_n = \left\lfloor \frac{k}{d_n} \right\rfloor$$

Subproblema: $\left(\langle d_1, \dots, d_{n-1} \rangle, \; k - \left\lfloor k/d_n \right\rfloor \right)$

Contra-exemplo

$$\forall_{1 \leq i < n} . \; d_{i+1} \ngeq 2 \times d_i$$

RZ 15/16 - II.a

Input: $\underbrace{\langle d_1, \ldots, d_n \rangle}_{\text{denominações por ordem crescente}}$, K → Valor do troco

Formalização: $\min \sum_{i=1}^{n} x_i$

$\sum_{i=1}^{n} x_i \cdot d_i = K$

$\forall_{1 \leq i < n} \cdot\ d_{i+1} \geq 2 \times d_i$

Escolha Greedy

$$x_n = \left\lfloor \frac{k}{d_n} \right\rfloor$$

Subproblema: $\left(\langle d_1, \ldots, d_{n-1} \rangle, k - \left\lfloor k/d_n \right\rfloor \right)$

Contra-exemplo

$\forall_{1 \leq i < n} \cdot\ d_{i+1} \ngeq 2 \times d_i$

$\vec{d} = \langle 1, 7, 8 \rangle$

$k = 14$

↳ $1 \times 8 + 6 \times 1 \Rightarrow 7$ moedas

$2 \times 7 \Rightarrow 2$ moedas