

Projeto de algoritmos: uma visão prática

Aula 05

Rafael Melo
Instituto de Computação
Universidade Federal da Bahia



RELEMBRANDO: Elementos de algoritmos gulosos

De maneira mais geral, segue-se a sequência de passos:

1. converta o problema de otimização tal que ao se fazer uma escolha, resta apenas um subproblema a resolver;
2. prove que sempre existe uma solução ótima para o problema original que faz a escolha gulosa, então ela sempre será segura;
3. demonstre que, fazendo-se a escolha gulosa, o que resta é um subproblema cuja solução pode ser combinada à escolha gulosa de maneira a se chegar a uma solução ótima global.

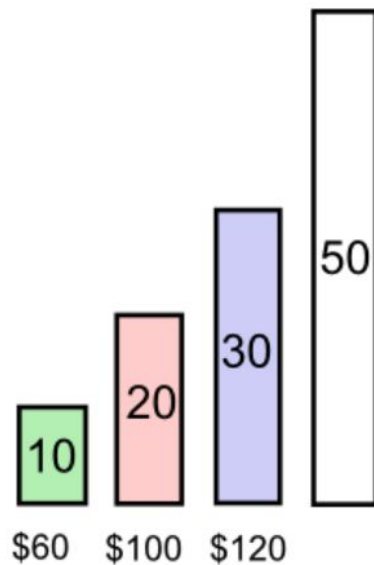
Problema da mochila fracionária

Problema da mochila fracionária

- **Entrada:** um conjunto $I = \{1, \dots, n\}$ de items, uma utilidade c_i e um peso a_i para cada $i \in I$, e uma capacidade b da mochila.
- **Saída:** selecionar a quantidade de cada item em I a ser carregado na mochila maximizando a soma total das utilidades de forma que a soma dos pesos não ultrapasse a capacidade da mochila.

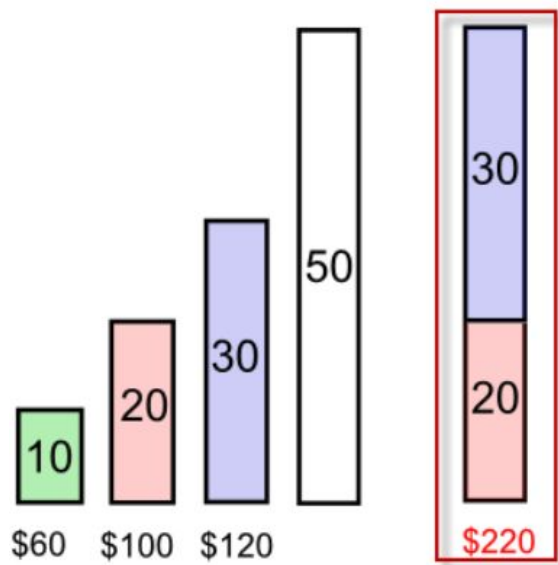
Exemplo

- Exemplo com 3 itens e mochila de capacidade 50.



Exemplo

- Solução viável 1



- Solução viável **ótima**



Algoritmo

```
1  def guloso_mochilafracionaria(itens, capacidade):
2      #itens recebidos como lista de triplas [(id,peso,utilidade)]
3
4      itens.sort(key=lambda x: x[2] / x[1], reverse=True)
5      utilidade_total = 0
6      peso_residual = capacidade
7
8      itens_selecionados = []
9
10     for id, peso, utilidade in itens:
11         if peso_residual >= peso:
12             utilidade_total += utilidade
13             peso_residual -= peso
14             itens_selecionados.append((id, 1.0))
15         else:
16             fracao = peso_residual / peso
17             utilidade_total += fracao * utilidade
18             itens_selecionados.append((id, fracao))
19             break
20
21     itens_selecionados.sort(key=lambda x: x[0])
22     return itens_selecionados, utilidade_total
```

Tempo de execução

- Ordenação dos itens por fração *utilidade/peso*: $\Theta(n \log n)$
- Algoritmo guloso para selecionar os itens de forma gulosa, uma vez que já estão ordenados: $O(n)$

Atividades práticas

Contato:
rafael.melo@ufba.br
rafaelmelo.ufba@gmail.com

