

Algoritmos Gulosos

Prof. Luciano Antonio Digiampietri

Problema da Mochila Binária

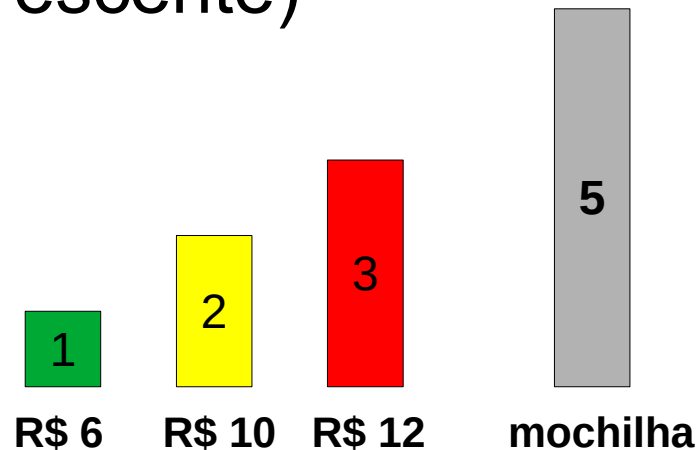
- Tentativa de solução gulosa:

Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 1) Ordenar os objetos pela relação valor/peso (de forma decrescente)

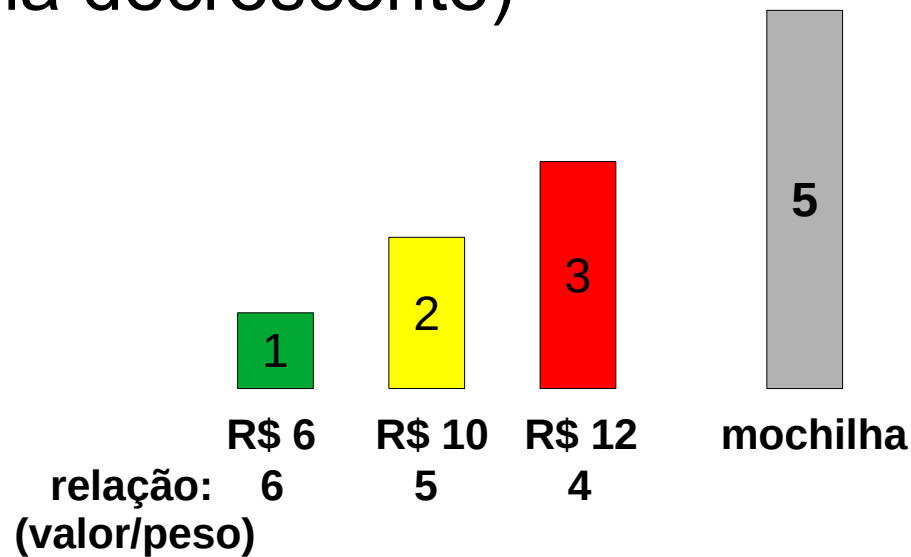
Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 1) Ordenar os objetos pela relação valor/peso (de forma decrescente)



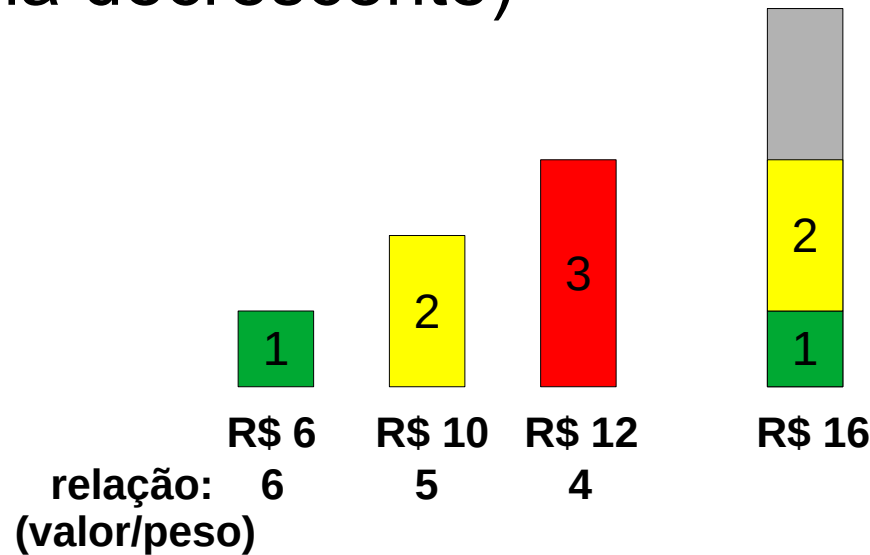
Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 1) Ordenar os objetos pela relação valor/peso (de forma decrescente)



Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 1) Ordenar os objetos pela relação valor/peso (de forma decrescente)

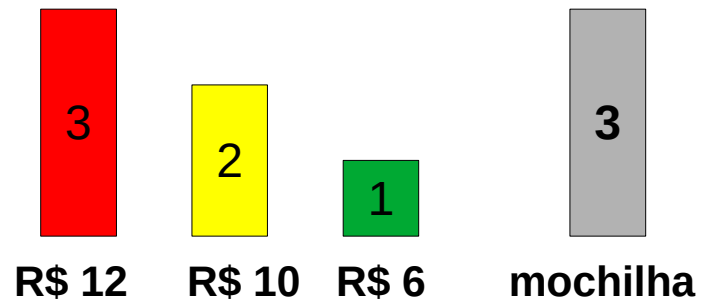


Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 2) Ordenar os objetos pelo maior valor (de forma decrescente)

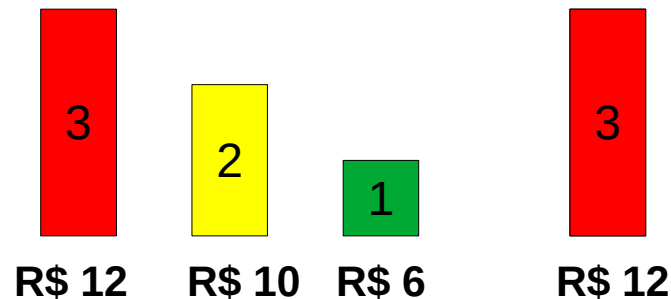
Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 1) Ordenar os objetos pelo maior valor por peso (de forma decrescente)
 - 2) Ordenar os objetos pelo maior valor (de forma decrescente)



Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 1) Ordenar os objetos pelo maior valor por peso (de forma decrescente)
 - 2) Ordenar os objetos pelo maior valor (de forma decrescente)

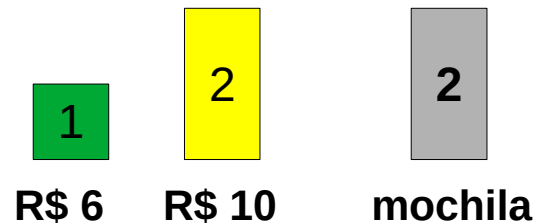


Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 3) Ordenar os objetos pelo peso

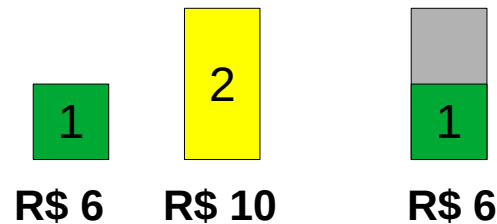
Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 3) Ordenar os objetos pelo peso



Problema da Mochila Binária

- Tentativa de solução gulosa:
3) Ordenar os objetos pelo peso



Problema da Mochila Binária

- Tentativa de solução gulosa:
 - 1) Ordenar os objetos pela relação valor/peso (de forma decrescente)
 - 2) Ordenar os objetos pelo maior valor (de forma decrescente)
 - 3) Ordenar os objetos pelo peso
- **Não há solução gulosa que sempre encontre a solução ótima para o problema da Mochila Binária**

Tentativa e Erro

```
class Objeto{
    int peso;
    int valor;
    Objeto(int p, int v){
        valor = v;
        peso = p;
    }
}

public class MochilaBinaria {
    static int melhorSolucao;
    static void resolveMochilaBinaria(Objeto[] objetos, int atual, int pesoDisponivel,
                                      int valorAtual){
        if (atual >= objetos.length) {
            if (valorAtual > melhorSolucao) melhorSolucao = valorAtual;
            return;
        }
        // NAO COLOCAR OBJETO ATUAL
        resolveMochilaBinaria(objetos, atual+1, pesoDisponivel, valorAtual);
        // SE POSSIVEL, COLOCAR OBJETO ATUAL
        Objeto obj = objetos[atual];
        if (obj.peso <= pesoDisponivel){
            resolveMochilaBinaria(objetos, atual+1, pesoDisponivel-obj.peso,
                                  valorAtual + obj.valor);
        }
    }
}
```