

Trabalho 3 - Análise de Algoritmos

Bernardo Vieira Santos - 2220502

Prof.: Marco Molinaro

Tarefa 1: Desenho do algoritmo

Equação de recorrência:

$\text{OPT}(i, b)$ é o valor máximo que podemos obter usando os itens $1..i$ com a capacidade B de mochila. Ao escolher um item, podemos escolher k unidades ($k \leq 10$) e precisamos respeitar o limite de peso da mochila, $k \cdot w_i \leq B$. Escolhendo k unidades subtraímos $k \cdot w_i$ de espaço mas ganhamos $k \cdot v_i$ de valor. Restam $B - k \cdot w_i$ de capacidade para ser preenchida com itens de 1 a $i - 1$, que vale $\text{OPT}(i - 1, b - k \cdot w_i)$. Então, para um k fixo, o valor total é:
$$\text{OPT}(i-1, b-k \cdot w_i) + k \cdot v_i$$

Dessa forma, pegando o melhor k possível temos a seguinte recorrência:

$$\text{OPT}(i, b) = \max_{k \leq 10} \{ \text{OPT}(i - 1, b - k \cdot w_i) + k \cdot v_i \}$$

Tarefa 2: Implementação do algoritmo

1. Trecho de código com o loop que preenche a tabela de memoização

```
# dp[i][b] = OPT(i, b): usando itens 1..i e capacidade b
dp = [[0] * (B + 1) for _ in range(n + 1)]

# choice[i][b] = número de unidades do item i escolhidas na solução
ótima de (i, b)
choice = [[0] * (B + 1) for _ in range(n + 1)]

for i in range(1, n + 1):
    w_i = weights[i - 1]
    v_i = values[i - 1]
    for b in range(B + 1):
        best_value = 0
        best_k = 0
        for k in range(0, 11):
            total_weight = k * w_i
            if total_weight > b:
                break
            value_candidate = dp[i - 1][b - total_weight] + k * v_i
            if value_candidate > best_value:
                best_value = value_candidate
                best_k = k
        dp[i][b] = best_value
        choice[i][b] = best_k
```

2. Relatório gerado via código com valor da solução ótima e quais/quantos itens foram selecionados:

```
==== Instância inst1 ====
n = 10, B = 269
Valor ótimo: 525
Itens usados (item: quantidade, valor, peso):
Item 2: 9 unidades (v=10, w=4)
Item 10: 5 unidades (v=87, w=46)
```

```
==== Instância inst2 ====
n = 20, B = 878
Valor ótimo: 2880
Itens usados (item: quantidade, valor, peso):
Item 2: 10 unidades (v=46, w=4)
Item 11: 10 unidades (v=78, w=32)
Item 12: 7 unidades (v=40, w=18)
Item 15: 10 unidades (v=61, w=25)
```

Item 19: 10 unidades ($v=75$, $w=14$)

==== Instância inst3 ===

$n = 4$, $B = 20$

Valor ótimo: 44

Itens usados (item: quantidade, valor, peso):

Item 2: 4 unidades ($v=11$, $w=5$)

==== Instância inst4 ===

$n = 4$, $B = 11$

Valor ótimo: 30

Itens usados (item: quantidade, valor, peso):

Item 1: 5 unidades ($v=6$, $w=2$)

==== Instância inst5 ===

$n = 10$, $B = 60$

Valor ótimo: 70

Itens usados (item: quantidade, valor, peso):

Item 7: 8 unidades ($v=5$, $w=5$)

Item 8: 10 unidades ($v=3$, $w=2$)

==== Instância inst6 ===

$n = 7$, $B = 50$

Valor ótimo: 107

Itens usados (item: quantidade, valor, peso):

Item 1: 1 unidades ($v=70$, $w=31$)

Item 4: 1 unidades ($v=37$, $w=19$)

==== Instância inst7 ===

$n = 5$, $B = 80$

Valor ótimo: 370

Itens usados (item: quantidade, valor, peso):

Item 4: 10 unidades ($v=37$, $w=8$)

==== Instância inst8 ===

$n = 20$, $B = 879$

Valor ótimo: 2880

Itens usados (item: quantidade, valor, peso):

Item 4: 10 unidades ($v=46$, $w=4$)

Item 9: 10 unidades ($v=61$, $w=25$)

Item 12: 7 unidades ($v=40$, $w=18$)

Item 14: 10 unidades ($v=75$, $w=14$)

Item 18: 10 unidades ($v=78$, $w=32$)