

# Relatório Trabalho 3

## Análise de Algoritmo

### Alunos:

Guilherme Ponce (2011179)

João Victor Godinho W. (2011401)

### Tarefa 1: Desenho do algoritmo

*A equação de recorrência original do problema da mochila para itens distintos pode ser descrito por:*

$$\text{OPT}(i,b) = \max \{ \text{OPT}(i - 1, b) , v_i + \text{OPT}(i - 1, b - b_i) \}$$

*Na equação original, caso o item coubesse na mochila, então você deveria pegar o maior valor entre o item fazer parte da solução ótima e o item não fazer parte da solução ótima. O decremento do peso do item e a soma de seu valor representam, justamente, a adição dele na mochila: menos espaço para o próximo item e o incremento em seu valor total para aquela combinação de itens.*

*Para o caso particular desta tarefa, temos a equação com uma cara um pouco diferente:*

$$\text{OPT}(i,b) = \max \{ \text{OPT}(i - 1, b) , (v_i * n) + \text{OPT}(i - 1, b - (b_i * n)) \}$$

*Sabendo que  $n$  é quantidade de itens daquele tipo que você irá selecionar, perceba que a única diferença entre elas é justamente essa: quantos deste item você irá selecionar para sua solução ótima.*

*t*

### Tarefa 2: Desenho do algoritmo

#### 1) Parte principal do algoritmo

```

QTD_MAX = 10 #quantidade máxima de vezes que posso pegar daquele item

tam = len(valor) #quantidade de itens que eu tenho. Poderia ter pego o peso
também para representar

opt = [[0] * (B + 1) for i in range(tam + 1)]

for i in range(1, tam + 1): #pula o primeiro elemento pois, sem nenhum item na
mochila, o valor sempre será zero
    for j in range(B + 1):
        if(peso[i-1] > j): #não cabe na mochila
            opt[i][j] = opt[i-1][j] #pega o valor anterior para o mesmo peso de
mochila
        else:
            for n in range(1, QTD_MAX + 1):
                if n * peso[i-1] <= j:
                    opt[i][j] = max(opt[i-1][j], (valor[i-1] * n) + opt[i-1][j-
(n * peso[i-1])])

```

## 2) Saída demonstrando solução ótima, quantos, e quais elementos foram selecionados

Hello world!

=== INSTANCIA 1 ===

QTDx item (v,w)

9x item (10,4)

5x item (87,46)

peso final: 266/269

valor ótimo: 525

=== =====

=== INSTANCIA 2 ===

QTDx item (v,w)

9x item (46,4)

3x item (90,43)

10x item (78,32)

10x item (61,25)

10x item (75,14)

peso final: 875/878

valor ótimo: 2824

=== =====

=== INSTANCIA 3 ===

QTDx item (v,w)

4x item (11,5)

*peso final: 20/20*

*valor ótimo: 44*

=== ===== == ===

=== INSTANCIA 4 ===

*QTDx item (v,w)*

*5x item (6,2)*

*peso final: 10/11*

*valor ótimo: 30*

=== ===== == ===

=== INSTANCIA 5 ===

*QTDx item (v,w)*

*4x item (5,5)*

*10x item (3,2)*

*10x item (1,1)*

*10x item (1,1)*

*peso final: 60/60*

*valor ótimo: 70*

=== ===== == ===

=== INSTANCIA 6 ===

*QTDx item (v,w)*

*1x item (70,31)*

*3x item (10,6)*

*peso final: 49/50*

*valor ótimo: 100*

=== ===== == ===

=== INSTANCIA 7 ===

*QTDx item (v,w)*

*10x item (37,8)*

*peso final: 80/80*

*valor ótimo: 370*

=== ===== == ===

=== INSTANCIA 8 ===

*QTDx item (v,w)*

*3x item (90,43)*

*10x item (46,4)*

*10x item (61,25)*

*10x item (75,14)*

*10x item (78,32)*

*peso final: 879/879*

*valor ótimo: 2870*

=== ===== == ===