

Universidade Federal de Mato Grosso do Sul

Campus Ponta Porã  
**Análise de Algoritmos I**

**Trabalho Prático II**  
**Mochila Booleana**

Aluno: Daniel de Leon Bailo da Silva  
RGA: 2017.1805.021-6  
Professor: Eduardo Theodoro Bogue

Maio  
2019

# Sumário

<b>1</b>	<b>Resumo</b>	<b>1</b>
<b>2</b>	<b>Knapsack Top-Down</b>	<b>2</b>
<b>3</b>	<b>Knapsack Bottom-Up</b>	<b>3</b>
<b>4</b>	<b>Resultados Obtidos</b>	<b>4</b>

# 1 Resumo

Este trabalho consiste em mostrar os resultados obtidos a partir da execução do algoritmo da *Mochilha Booleana* ou *Knapsack 0/1*, em suas versões dinâmicas. Feito isso, dada as instâncias para realizar os testes, foi comparado o tempo de execução do algoritmo em questão, nas suas duas versões dinâmicas, *Top-Down* e *Bottom-Up*.

Este trabalho possui um repositório online para melhor controle do versionamento e testes conforme a mudança do programa final.

## 2 Knapsack Top-Down

```
def topDown(number_items, weight_max, values_items, weight_items):  
    if number_items == 0 or weight_max == 0: return 0  
    if weight_items[number_items-1] > weight_max:  
        return topDown(number_items-1, weight_max, values_items, weight_items)  
    if mem[number_items][weight_max] is not False:  
        return mem[number_items][weight_max]  
  
    temp = max(topDown(number_items-1,  
weight_max-weight_items[number_items-1], values_items, weight_items)  
+values_items[number_items-1],  
topDown(number_items-1, weight_max, values_items, weight_items))  
  
    mem[number_items][weight_max] = temp  
    return temp
```

### 3 Knapsack Bottom-Up

```
def bottomUp(number_items, weight_max, values_items, weight_items):
    K = [[0 for x in range(weight_max + 1)] for x in range(number_items + 1)]
    for i in range(number_items + 1):
        for w in range(weight_max + 1):
            if i == 0 or w == 0: K[i][w] = 0
            elif weight_items[i-1] <= w: K[i][w] = max(values_items[i-1] +
                K[i-1][w-weight_items[i-1]], K[i-1][w])
            else: K[i][w] = K[i-1][w]
    return K[number_items][weight_max]
```

## 4 Resultados Obtidos