

Template para Apresentações

Prof. Dr. George H. G. Fonseca

Nome da matéria

Nome do curso/departamento

Universidade Federal de Ouro Preto

Março de 2023



UFOP

Universidade Federal
de Ouro Preto

1. Texto em Tópicos

Texto em Tópicos

- Esse slide apresenta o texto organizado em tópicos e sub-tópicos.
- O restante do modelo apresenta os seguintes exemplos:
 - Figuras.
 - Tabelas.
 - Algoritmos.
 - Código Fonte.
 - Modelos Matemáticos.
 - Referências.

Texto em Tópicos - *overlays*

- É possível quebrar um slide em vários usando *overlays*:
 - Figuras.

Texto em Tópicos - *overlays*

- É possível quebrar um slide em vários usando *overlays*:
 - Figuras.
 - Tabelas.

Texto em Tópicos - *overlays*

- É possível quebrar um slide em vários usando *overlays*:
 - Figuras.
 - Tabelas.
 - Algoritmos.

Texto em Tópicos - *overlays*

- É possível quebrar um slide em vários usando *overlays*:
 - Figuras.
 - Tabelas.
 - Algoritmos.
 - Código Fonte.

Texto em Tópicos - *overlays*

- É possível quebrar um slide em vários usando *overlays*:
 - Figuras.
 - Tabelas.
 - Algoritmos.
 - Código Fonte.
 - Modelos Matemáticos.

Texto em Tópicos - *overlays*

- É possível quebrar um slide em vários usando *overlays*:
 - Figuras.
 - Tabelas.
 - Algoritmos.
 - Código Fonte.
 - Modelos Matemáticos.
 - Referências.

2. Figuras

Figuras

- Figuras, tabelas e elementos não textuais são trabalhados exatamente como em documentos comuns LaTeX.
- Pode ser interessante incluir um espaçamento entre o texto e a imagem pelo comando `\vspace{3mm}`



Figura: É possível inserir legendas também.

Duas colunas: figura e texto

- Deste lado fica o texto, do outro a figura:

- item 1
- item 2
- item 3
- item 4
- item 5



3. Tabelas

Tabelas

- Uma excelente ferramenta para a criação de tabelas é o <https://www.tablesgenerator.com/>
- Um exemplo de tabela é dado abaixo

Tabela: É possível inserir legendas também.

Instância	Algoritmo A	Algoritmo B	Algoritmo C
<i>Arquivo 1</i>	1.00 ± 0.7	1.90 ± 0.5	0.90 ± 0.3
<i>Arquivo 2</i>	6.80 ± 0.5	8.60 ± 1.4	5.90 ± 0.2
<i>Arquivo 3</i>	7.90 ± 1.2	13.25 ± 3.1	8.50 ± 1.7
<i>Arquivo 4</i>	8.00 ± 0.0	12.70 ± 0.2	6.10 ± 1.0

4. Algoritmos

Algoritmos

- Siga as convenções para pseudocódigos conforme <https://link.springer.com/content/pdf/bbm%3A978-1-4471-5173-9%2F1.pdf>
- Para trocar o idioma das palavra-chave altere o idioma de *portuguese* para *english* na linha:
`\usepackage[portuguese, ...]{algorithm2e}`

Algoritmos

- Pseudocódigo do algoritmo de ordenação Bubble-Sort:

Algoritmo 1: BUBBLE-SORT.

Entrada: (i) Um vetor V .

Saída: (i) O vetor V ordenado crescentemente.

```
1 para  $i = 0$  até  $|V| - 1$  faça
2   trocou  $\leftarrow$  false;
3   para  $j = 0$  até  $|V| - 1$  faça
4     se  $V[j + 1] > V[j]$  então
5       aux  $\leftarrow V[j]$ ;  $\triangleright$  Troca as posições  $i$  e  $i + 1$  de  $V$ 
6        $V[j] \leftarrow V[j + 1]$ ;
7        $V[j + 1] \leftarrow$  aux;
8     trocou  $\leftarrow$  true;
9   se trocou então
10    break;  $\triangleright V$  já está ordenado
11 retorna  $V$ ;
```

5. Código-Fonte

Código-Fonte

- Segue um exemplo de implementação do algoritmo Bubble-Sort na linguagem Python

```
1 def bubble_sort(V): #Complexidade  $O(N^2)$ 
2     for i in range(len(V) - 1):
3         trocou = False
4         for j in range(len(V) - 1):
5             if V[j + 1] < V[j]:
6                 aux = V[j]
7                 V[j] = V[j + 1]
8                 V[j + 1] = aux
9                 trocou = True
10        if not trocou:
11            break
12    return V
```

6. Modelos Matemáticos

Modelos Matemáticos

Problema da mochila

- Conjuntos e parâmetros

\mathcal{I} Itens

w_i Peso do item i

b_i Benefício do item i

C Capacidade da mochila

- Variáveis

$$x_i = \begin{cases} 1 & \text{se o item } i \in \mathcal{I} \text{ está na mochila.} \\ 0 & \text{caso contrário.} \end{cases}$$

Modelos Matemáticos

- Restrições

- ① Capacidade da mochila:

$$\sum_{i \in \mathcal{I}} w_i \times x_i \leq C \quad (1)$$

- Função objetivo


$$\max \sum_{i \in \mathcal{I}} b_i \times x_i \quad (2)$$


7. Referências

Referências

- Sempre que se utilizar do trabalho de outrem é necessário incluir uma citação, que pode fazer parte do texto ou não.
 - Textual** Fonseca et al. (2017) apresenta uma nova formulação para o problema.
 - Não-textual** O problema de programação de horários é classificado como NP-Difícil (EVEN; ITAI; SHAMIR, 1975).

Referências I

 EVEN, S.; ITAI, A.; SHAMIR, A. On the complexity of time table and multi-commodity flow problems. In: *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*. [S.l.: s.n.], 1975. p. 184–193.

 FONSECA, G. H. et al. Integer programming techniques for educational timetabling. *European Journal of Operational Research*, v. 262, n. 1, p. 28 – 39, 2017.