

Universidade Federal de Ouro Preto  
PCC104 - Projeto e Análise de Algoritmos  
Força Bruta e Busca Exaustiva

Prof. Rodrigo Silva

March 28, 2022

## Instruções

- Implementar as atividades práticas em C++ é altamente recomendado.
- Utilize ao máximo os algoritmos e estruturas de dados da biblioteca STL. <https://www.geeksforgeeks.org/the-c-standard-template-library-stl/>.
- Evite ao máximo a utilização de ponteiros, mas se precisar, utilizar ponteiros inteligentes <https://alandefreitas.github.io/moderncpp/basic-syntax/pointers/smart-pointers/>.
- Quando precisar de uma estrutura de dados linear sempre avalie primeiro a utilização da classe `vector` (<https://en.cppreference.com/w/cpp/container/vector>)

## 1 Leitura Recomendada

- Capítulo 3 - *Introduction to the Design and Analysis of Algorithms (3rd Edition)* - Anany Levitin
- Livro - *Introdução à programação* - Alan de Freitas (disponível em <http://www.decom.ufop.br/alan/bcc702/livrocpp.pdf>)
- Livro - *Problem Solving with Algorithms and Data Structures using C++* (disponível em: <https://runestone.academy/runestone/books/published/cppds/index.html#>)

## 2 Vídeos Recomendados

- Confira a playlist de C++ do Prof. Alan de Freitas (UFOP) - <https://www.youtube.com/watch?v=jes0Z6i-3DA&list=PLIUc9-A-aPpqrzY3YuWDU0yQL0BCb5lck>

## 3 Questões Teóricas

1. Apresente uma descrição da classe `list` (<https://en.cppreference.com/w/cpp/container/list>) apresentando o custo computacional de cada um de suas operações.
2. Apresente uma descrição da classe `set` (<https://en.cppreference.com/w/cpp/container/set>) apresentando o custo computacional de cada um de suas operações.
3. Apresente uma descrição da classe `vector` (<https://en.cppreference.com/w/cpp/container/vector>) apresentando o custo computacional de cada um de suas operações.

## 4 Atividades Prática

1. Implementar o algoritmo *Bubble Sort*
2. Implementar o algoritmo *Selection Sort*
3. Implementar o algoritmo *SequentialSearch2* apresentado na Seção 3.2 do Capítulo 3<sup>1</sup>
4. Implementar o algoritmo *BruteForceStringMatch* apresentado na Seção 3.2 do Capítulo 3<sup>1</sup>
5. Implementar o algoritmo *BruteForceClosestPair* apresentado na Seção 3.2 do Capítulo 3<sup>1</sup>
6. Implementar um algoritmo que dado um conjunto de pontos  $S$  calcule sua casca convexa (*convex hull*) (Ver Seção 3.3 do Capítulo 3<sup>1</sup>).
7. Implemente uma solução baseada em busca exaustiva para o problema do Caixeiro Viajante (Traveling Salesman Problem).
8. Implemente uma solução baseada em busca exaustiva para o problema da Mochila (Knapsack Problem).
9. Implemente o algoritmo de busca em largura.
10. Implemente o algoritmo de busca em profundidade.

---

<sup>1</sup>Livro - *Introduction to the Design and Analysis of Algorithms (3rd Edition)*