

Universidade Federal de Ouro Preto
PCC104 - Projeto e Análise de Algoritmos
Dividir e Conquistar

Prof. Rodrigo Silva

April 25, 2022

Dicas

- 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 4 - *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 Atividades

1. Implemente o algoritmo MergeSort.
2. Mostre como o MergeSort ordena a lista E, X, A, M, P, L, E em ordem alfabética.
3. O MergeSort é um algoritmo estável? Explique.
4. Implemente um algoritmo de divisão e conquista para encontrar a posição do maior elemento de um arranjo de n elementos. Responda também:
 - (a) Qual será a saída do método quando se vários elementos do arranjo tiverem o maior valor?
 - (b) Defina e resolva a relação de recorrência para o método proposto.
 - (c) Como este algoritmo se compara com uma solução força bruta?
5. Implemente um algoritmo de divisão e conquista para encontrar a posição do maior e do menor elemento de um arranjo de n elementos. Responda também:

- (a) Defina e resolva a relação de recorrência para o método proposto.
- (b) Como este algoritmo se compara com uma solução força bruta?
- 6. Implemente o algoritmo QuickSort.
- 7. Mostre como o QuickSort ordena a lista E, X, A,M, P, L, E em ordem alfabética. Desenhe a árvore de chamadas feitas.
- 8. Dê um exemplo que mostre que o QuickSort não é um algoritmo estável.
- 9. Implemente a estrutura de dados Árvore Binária.
- 10. Implemente um algoritmo recursivo que encontre o tamanho de uma árvore binária.
- 11. Mostre o resultado dos caminhamentos *preorder*, *postorder* e *inorder* para a seguinte árvore binária.

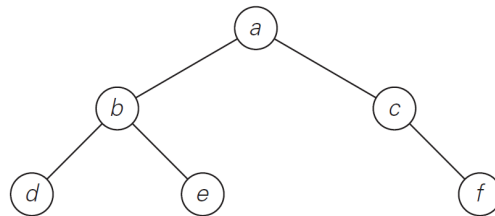


Figure 1: Exemplo de árvore binária

- 12. Implemente os caminhamentos *preorder*, *postorder* e *inorder* para árvores binárias.
- 13. Implemente o algoritmo de Strassen para multiplicação de matrizes? Como este algoritmo se compara com o algoritmo de força bruta?
- 14. Implemente o algoritmo QuickHull que usa o paradigma de divisão e conquista para encontrar o ConvexHull de um conjunto de pontos. Como o seu custo se compara com o algoritmo de força bruta?
- 15. Apresente o Teorema Mestre.