

Estruturas de dados

Ordenação

Marco A L Barbosa

malbarbo.pro.br

Introdução

- 1) Descreva o problema de ordenação.
- 2) O que é uma ordenação in-loco?
- 3) O que é uma ordenação estável?
- 4) Qual é a ideia geral de um algoritmo incremental?
- 5) Qual é a ideia geral de um algoritmo de divisão e conquista?

Começando

- 6) Crie um exemplo de arranjo que mostre que a ordenação por seleção não é estável.
- 7) Explique qual é a relação entre os algoritmos de ordenação por inserção, seleção e heap.
- 8) O arranjo [23, 17, 14, 6, 13, 10, 1, 5, 7, 12] representa um heap máximo?
- 9) Mostre passo a passo a construção de um heap mínimo a partir do arranjo [5, 1, 7, 4, 9, 1, 3, 4, 8, 6].
- 10) Crie um exemplo de arranjo que mostre que a ordenação por heap não é estável.
- 11) Explique qual é a relação entre os algoritmos de ordenação por intercalação e particionamento.

Praticando

- 12) Faça uma implementação iterativa da função `concerta_heap`.
- 13) Um outro algoritmo de ordenação é a ordenação por flutuação (do inglês, *bubble-sort*). A ideia do algoritmo é percorrer os elementos da sequência diversas vezes e fazer o maior elemento “flutuar” para o final. Isto pode ser feito da seguinte forma: a cada passagem pelos elementos, cada par de elementos adjacentes e comparado e trocando de lugar se o primeiro for maior que o segundo. O algoritmo para quando uma passagem por todos os elementos não gerar nenhuma troca (você consegue ver porque o algoritmo funciona?). Implemente o algoritmo de ordenação por flutuação para arranjos e listas encadeadas. Determine a complexidade de tempo, se ele é in-loco e se é estável.
- 14) Partindo do arquivo `ordena_encadeamento.py`, implemente o algoritmo de ordenação por seleção para listas encadeadas. Faça uma versão in-loco e outra que devolve uma nova lista. Qual você achou mais simples?

Avançando

- 15) Projete a função `intercala(lst: list[int], ini: int, meio: int, fim: int)` que intercala os elementos de `lst[ini:meio]` (que está ordenado) com `lst[meio:fim]` (que está ordenado) de maneira que `lst[ini:fim]` fique ordenado. Dica: cria arranjos auxiliares.
- 16) Projete uma função que implemente o esquema de particionamento de Hoare.