# Lista de Ordenação

Estrutura de Dados

Prof. Roberto Cabral

20 de outubro de 2022

1. Uma amigo lhe disse que é capaz de ordenar qualquer conjunto de três números com no máximo 2 comparações. O seu amigo está falando a verdade ou mentindo? Justifique sua resposta.

2. Uma amigo lhe disse que é capaz de ordenar qualquer conjunto de quatro números com no máximo 5 comparações. O seu amigo está falando a verdade ou mentindo? Justifique sua resposta.

3. Um vetor $v[p..r]$ está "arrumado" se existe $j \in [p, r]$ tal que $v[p..j - a] \leq v[j] < v[j + 1..r]$. Escreva um algoritmo que decida se $v[p..r]$ está arrumado. Em caso afirmativo, seu algoritmo deve devolver o valor de $j$.

4. Discuta como a escolha do pivô pode influenciar no desempenho do método `quicksort`. Proponha estratégias para a escolha do pivô, visando melhorar seu desempenho.

5. Dada a sequência de números: 13 7 11 2 5 17 7 13 4 6 7 3 7 10 54 13, ordene-a em ordem crescente segundo cado um dos algoritmo estudados em sala. Para cada algoritmo, mostre o número de comparações e trocas que realizam na ordenação de sequências.

6. Dos algoritmos estudados, quais são estáveis? Utilize a questão anterior para apoiar sua resposta.

7. Considere a ordenação de $n$ números armazenados no arranjo A, localizando primeiro o menor elemento de A e permutando esse elemento contido em A[1]. Em seguida, encontre o segundo menor elemento de A e o troque pelo elemento A[2]. Continue dessa maneira para os primeiros $n - 1$ elementos de A. Implemente esse algoritmo que é conhecido como ordenação por seleção. Qual invariante do laço esse algoritmo mantém? Por que ele só precisa ser executado para os primeiros $n - 1$ elementos, e não para todos os elementos? Forneça os tempos de execução do melhor caso e do pior caso da ordenação por seleção em notação $O$.

8. Dado um conjunto de $n$ inteiros distintos e um inteiro positivo $k \leq n$:

   (a) Proponha um algoritmo que imprime os $k$ menores elementos do conjunto (em qualquer ordem) em tempo O($n$).

(b) Suponha agora que queremos imprimir os $k$ menores elementos em ordem crescente. É ainda possível fazer isso em tempo $O(n)$ para quais quer valores de $k \leq n$?

9. Modifique a função `partition`(partição) do QuickSort de modo que o valor do meio (mediano) de x[menor], x[maior] e x[meio] (onde meio = (maior + meio) / 2) seja usado para particionar o vetor. Em que casos o Quicksort usará esse método com mais eficiência do que a versão apresentada em aula? Em que casos ele será menos eficiente?

10. Implemente um algoritmo que ordena um vetor de inteiros e retorna a quantidade de inversões que ocorreram.

11. Para cada um dos cinco algoritmos de ordenação estudados em aula, responda as seguintes perguntas:

    (a) Explique, resumidamente, o funcionamento do algoritmo.

    (b) Qual a complexidade de melhor caso?

    (c) Qual a complexidade de pior caso?

12. Seja $v$ um vetor de inteiros de tamanho $n$. Faça uma função que ordena o vetor $v$ em ordem decrescente usando uma versão recursiva do algoritmo de ordenação por inserção.

13. O algoritmo Mergesort (decrescente) usa uma função auxiliar, chamada intercala, que recebe como entrada dois vetores em ordem decrescente $v[p \ldots q-1]$ e $v[q \ldots r-1]$ e rearranja $v[p \ldots r-1]$ em ordem decrescente. Escreva a função intercala para o Mergesort decrescente.

14. Reescrever o procedimento de partição do QuickSort (Separa) tomando como referência (pivô) o primeiro elemento (na implementação mais usual, toma-se o último).

15. Usando cada um dos algoritmos de ordenação estudado em sala de aula, implemente uma função que ordena um vetor de inteiros da seguinte forma: os números pares são ordenados em ordem crescente no início do vetor e os ímpares são ordenados em ordem decrescente no final do vetor.