

Lista 01

Estrutura de Dados
Prof. Roberto Cabral
24 de Outubro de 2017

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. Uma amigo lhe disse que é capaz de ordenar qualquer conjunto de cinco números com no máximo 7 comparações. O seu amigo está falando a verdade ou mentindo? Justifique sua resposta.
4. Uma amigo lhe disse que é capaz de ordenar qualquer conjunto de seis números com no máximo 9 comparações. O seu amigo está falando a verdade ou mentindo? Justifique sua resposta.
5. No método **insertsort**, a cada passo, o menor elemento é procurado para que seja inserido na sequência já ordenada. Essa procura pode ser realizada sequencialmente ou por busca binária. Analise o desempenho de ambas as abordagens.
6. 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 .
7. Discuta como a escolha do pivô pode influenciar no desempenho do método **quicksort**. Proponha estratégias para a escolha do pivô, visando melhorar seu desempenho.
8. 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 cada um dos algoritmo estudados em sala. Para cada algoritmo, mostre o número de comparações e trocas que realizam na ordenação de sequências.
9. Dos algoritmos estudados, quais são estáveis? Utilize a questão anterior para apoiar sua resposta.

10. 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 .
11. 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 $k \leq n$?
12. Modifique a função `partition`(partição) do QuickSort de modo que o valor do meio (mediano) de $x[\text{menor}]$, $x[\text{maior}]$ e $x[\text{meio}]$ (onde $\text{meio} = (\text{maior} + \text{menor}) / 2$) 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?
13. Considere a seguinte ordenação por **seleção quadrática**: divida os n elementos do vetor em \sqrt{n} grupos de \sqrt{n} elementos cada. Encontre o maior elemento de cada grupo e insira-o num vetor auxiliar. Encontre o maior elemento nesse vetor auxiliar. Esse será o maior elemento do vetor. Em seguida, substitua esse elemento dentro do vetor pelo maior elemento seguinte do grupo a que ele pertencia. Ache novamente o maior elemento do vetor auxiliar. Esse será o segundo maior elemento do vetor. Repita o processo até que o arquivo esteja classificado. Escreva uma função para implementar uma ordenação por seleção quadrática o mais **eficiente possível**.
14. Implemente um algoritmo que ordena um vetor de inteiros e retorna a quantidade de inversões que ocorreram.
15. Para cada um dos seis 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?
16. 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.

17. O algoritmo Mergesort (decrecente) usa uma função auxiliar, chamada intercala, que recebe como entrada dois vetores em ordem decrescente $v[p \dots q - 1]$ e $v[q \dots r - 1]$ e rearranja $v[p \dots r - 1]$ em ordem decrescente. Escreva a função intercala para o Mergesort decrescente.
18. Reescrever o procedimento de partição do QuickSort (Separa) tomando como referência (pivô) o primeiro elemento (na implementação mais usual, toma-se o último).
19. Seja o vetor $[11, 13, 7, 3, 2, 5, 17]$. Mostre a ação do Heapsort sobre este vetor. Mostre primeiramente o passo a passo da construção da heap inicial. Na sequência, mostre o passo a passo da execução do algoritmo Heapsort.
20. Faça uma função que recebe como entrada um vetor crescente $v[e + 1 \dots d - 1]$ e um inteiro x tal que $v[e] < x \leq v[d]$ e devolve um índice j em $e + 1 \dots d$ tal que $v[j - 1] < x \leq v[j]$. Seu algoritmo deve ter complexidade $O(\log n)$.
21. Seja v uma lista de inteiros de tamanho n . Faça uma função que retorna uma lista v' contendo os elementos da lista v ordenados em ordem crescente. Sua função não deve alocar novas células na memória, mas reaproveitar as células de v . Nota: deve-se explicitar qual algoritmo de ordenação está sendo usando.
22. A ordenação por inserção bidirecional é uma modificação da ordenação por inserção simples da seguinte forma: um vetor de saída separado, de tamanho n , é reservado. Esse vetor de saída atua como uma estrutura circular. $x[0]$ é posicionado no elemento do meio do vetor. Assim que um grupo contíguo de elementos estiver no vetor, será aberto espaço para um novo elemento deslocando todos os elementos menores um passo para a esquerda ou todos os elementos maiores um passo a direita. A escolha do deslocamento é feita para provocar o menor número de movimentações. Escreva uma função para implementar essa técnica.
23. 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.
24. Implemente cada um dos algoritmos de ordenação estudados em sala de aula.
25. Implemente uma versão recursiva de cada um dos algoritmos de ordenação estudados em sala de aula.
26. Usando lista duplamente encadeada, implemente cada um dos algoritmos de ordenação estudados em sala de aula.
27. Usando lista duplamente encadeada, implemente uma versão recursiva de cada um dos algoritmos de ordenação estudados em sala de aula.