

Relatório do Projeto

Nome: Domingos Neto

RA: 318117865

O projeto implica na aplicação e verificação de 5 métodos de ordenação, sendo eles: Bubble Sort, Insert Sort, Selection sort, Merge Sort e Quick Sort, que veremos abaixo a definição resumida deles.

O bubble sort, ou ordenação por flutuação (literalmente "por bolha"), é um [algoritmo de ordenação](#) dos mais simples. A ideia é percorrer o [vector](#) diversas vezes, e a cada passagem fazer flutuar para o topo o maior elemento da sequência. Essa movimentação lembra a forma como as bolhas em um [tanque](#) de água procuram seu próprio nível, e disso vem o nome do algoritmo.

Insertion Sort, ou ordenação por inserção, é o algoritmo de ordenação que, dado uma estrutura (array, lista) constrói uma matriz final com um elemento de cada vez, uma inserção por vez. Assim como algoritmos de ordenação quadrática, é bastante eficiente para problemas com pequenas entradas, sendo o mais eficiente entre os algoritmos desta ordem de classificação.

A ordenação por seleção (do inglês, selection sort) é um [algoritmo de ordenação](#) baseado em se passar sempre o menor valor do vetor para a primeira posição (ou o maior dependendo da ordem requerida), depois o de segundo menor valor para a segunda posição, e assim é feito sucessivamente com os $n - 1$ elementos restantes, até os últimos dois elementos.

O merge sort, ou [ordenação por mistura](#), é um exemplo de [algoritmo de ordenação por comparação](#) do tipo [dividir-para-conquistar](#).

Sua ideia básica consiste em Dividir (o problema em vários subproblemas e resolver esses subproblemas através da recursividade) e Conquistar (após todos os subproblemas terem sido resolvidos ocorre a conquista que é a união das resoluções dos subproblemas). Como o algoritmo Merge Sort usa a recursividade, há um alto consumo de memória e tempo de execução, tornando esta técnica não muito eficiente em alguns problemas.

O [algoritmo quicksort](#) é um método de ordenação muito rápido e eficiente, inventado por [C.A.R. Hoare](#) em 1960[1], quando visitou a [Universidade de Moscovo](#) como estudante. Naquela época, Hoare trabalhou em um projeto de [tradução de máquina](#) para o [National Physical Laboratory](#). Ele criou o quicksort ao tentar traduzir um dicionário de inglês para russo, ordenando as palavras, tendo como objetivo reduzir o problema original em subproblemas que possam ser resolvidos mais fácil e rápido. Foi publicado em 1962 após uma série de refinamentos.[2]

O quicksort é um algoritmo de [ordenação por comparação não-estável](#).

Fonte : https://pt.wikipedia.org/wiki/Wikip%C3%A9dia:P%C3%A1gina_principal

A implementação do projeto foi baseada na criação dos vetores crescente, decrescente e aleatório alocados em cada método de ordenação para exibir o tempo de resposta de cada um, feito com o comando stopwatch().

Abaixo o informativo com o retorno no terminal de cada um dos métodos:

Vetor ordenado pelo insertion sort

Tempo de execução do Insertion Sort vetor decrescente : 00:00:00.3992357

Tempo de execução do Insertion Sort vetor crescente : 00:00:00.0001390

Tempo de execução do Insertion Sort vetor randomico : 00:00:00.2149480

Aperte ENTER para continuar

Vetor ordenado pelo Selection sort

Tempo de execução do Selection Sort vetor decrescente : 00:00:00.2833513

Tempo de execução do Selection Sort vetor crescente : 00:00:06.6146223

Tempo de execução do Selection Sort vetor randomico : 00:00:00.2644056

Aperte ENTER para continuar

Vetor ordenado pelo Bubble sort

Tempo de execução do Bubble Sort vetor decrescente : 00:00:00.6988123

Tempo de execução do Bubble Sort vetor crescente : 00:00:00.3756318

Tempo de execução do Bubble Sort vetor randomico : 00:00:00.7288446

Aperte ENTER para continuar

Vetor ordenado pelo Quick sort

Tempo de execução do Quick Sort vetor decrescente : 00:00:00.5203523

Tempo de execução do Quick Sort vetor crescente : 00:00:00.5824210

Tempo de execução do Quick Sort vetor randomico : 00:00:00.0021488

Aperte ENTER para continuar

Vetor ordenado pelo

Merge sort

Tempo de execução do Merge Sort vetor decrescente: 00:00:00.0058617

Tempo de execução do Merge Sort vetor crescente: 00:00:00.0032967

Tempo de execução do Merge Sort vetor randomico: 00:00:00.0050666

Aperte ENTER para continuar

Conclusão

O projeto serviu para conhecer melhor esses métodos e saber diferenciar cada um deles. Também me proporcionou um aprofundamento na criação de vetores e do uso do FOR.