

Algoritmos e Estrutura de dados 1 - Bubble Sort

Discentes:

Ana Santana Clementino de Araujo

Janieli Tainar da Silva

João Pedro Medeiros Nogueira Moura

Rafael Oliveira Freire

Método de ordenamento: Consiste em organizar uma sequência de números ou outras variáveis na ordem crescente, o algoritmo examina cada conjunto de elementos de um vetor, da esquerda para a direita, trocando suas posições se estiverem fora de ordem, repetindo esse processo até que não sobrem mais elementos para serem trocados, finalizando assim, a classificação dos elementos.

Método Utilizado: No nosso caso, estamos organizando alfabeticamente as strings de um vetor de acordo com o sistema ASCII, por meio do algoritmo Bubble Sort.

No início do código declaramos as bibliotecas "stdio", "string", "stdlib", "time" e o TAD "aluno", variáveis para a contagem do tempo de execução do algoritmo e para guardá-lo, abrimos uma pasta chamada "ListaDeAlunos.txt" em modo "w" que significa escrita, depois há uma verificação se o arquivo teve êxito ou não na sua criação e a seguir nós declaramos variáveis que vão guardar a quantidade de pessoas a serem cadastradas e a partir dessa quantidade definimos o tamanho da variável "p" do tipo "estrutura aluno", a qual foi renomeada apenas para "Aluno" que foi alocada na função "AlocaAlunos". Além disso, na função citada, insere-se as informações do aluno (Nome, matrícula e documento, que pode ser CPF ou RG).

A função que será executada em seguida será a "OrdenaAlunos", onde de fato acontece a implementação do algoritmo Bubble Sort. Nele existem 3 variáveis, duas de contagem usadas nos laços de repetição "for" e uma usada para auxiliar na ordenação de strings (que será explicado mais à frente). Em seguida, declaramos um vetor do tipo caractere com tamanho igual a 100 que servirá para guardar temporariamente a string. Os dois laços de repetição "for", um dentro do outro,

servem para acessar duas posições do vetor do tipo "Aluno" ao mesmo tempo. Então a partir daí entra a variável auxiliar "r" citada anteriormente, a qual vai guardar o resultado da função strcmp (que compara qual string acessada é maior em relação ao código ASCII), que pode ser 0 (caso os dois sejam iguais), 1 (caso o primeiro seja maior que o segundo) ou -1 (caso o segundo seja maior que o primeiro).

Após a execução da função "strcmp", vemos se o valor retornado é maior que 0 e, caso seja, ele será ordenado alfabeticamente pelo algoritmo em questão, utilizando a função strcpy, copiamos o conteúdo do segundo parâmetro para o primeiro, realizando esse processo 3 vezes, primeiro copiamos o conteúdo de "Vet[ i ].nome" para o vetor "temp", depois o conteúdo de "Vet[ i ].nome" é copiado para "Vet[ j ].nome", depois o conteúdo de "temp" é copiado para "Vet[ j ].nome".

A última função a ser executada é a "ImprimeAluno", que consiste basicamente em uma função que acessa cada uma das posições do vetor do tipo "Aluno" e imprime os dados dos alunos (nome, matrícula e documento, que pode ser CPF ou RG) tanto no terminal quanto no documento que foi criado.

Em seguida o arquivo é fechado, o espaço reservado para a estrutura "Aluno" é liberado e o tempo de execução é informado. Fim do programa.

Falando agora sobre a **complexidade do programa** tem-se que pelo fato do uso de dois laços de repetição no programa para a implementação do Bubble Sort isso faz com que a complexidade do algoritmo chegue ao  $O(n^2)$  o que não se caracteriza como um ótimo algoritmo em relação ao seu tempo de execução, mas ainda assim ele não se encontra na pior categoria dos algoritmos que seria  $O(n!)$ .

O algoritmo de Bubble Sort pode ser utilizado nos mais diversos tipos de casos e dados, como em sites, apps, entre outros, tendo uma fácil e simples implementação. Mas pode não ser uma boa escolha quando se lida com vários dados, principalmente se forem dados sigilosos de contas e senhas, devido a sua complexidade, podendo reduzir a velocidade do site ou app em que ele está sendo implementado.

Em conclusão, com o desenvolvimento deste projeto, toda a equipe aprendeu uma nova forma de ver a programação e, especificamente, diversas maneiras de manipular estruturas, variáveis e vetores na linguagem C, por fim, mas não menos importante aprendemos como trabalhar em conjunto.

Link do repositório: <https://github.com/classroom-ufersa/BubbleSort>