**PRÁTICA 7 – Trabalho 2 Exercício 7: Ordenação por QuickSort**

**Nome:** Davi Gabriel Domingues

**Número USP:** 15447497

A atividade solicitou a ordenação de um registro (“struct” /estrutura) com os campos: nome (string) e idade (int), considerando cada item com suas demasiadas solicitações. Dessa forma, tem – se as seguintes respostas apropriadas para cada item:

1. O programa, em C++, a ser ofertado foi utilizado tanto para responder este item, como também aos demais, uma vez que cada um deles solicita diferentes aplicações do mesmo em diferentes contextos desejados pela atividade em questão. O código final é este:

#include <iostream>

#include <string>

using namespace std;

int movimentacoes = 0, comparacoes = 0, chamadasRecursivas = 0;

struct Registro{

string nome;

int idade;

};

void preencherRegistro(int limite, Registro \*registro){

for (int i = 0; i < limite; i++){

cout<<"Nome do registrado "<<(i+1)<<": ";

getline(cin, registro[i].nome);

while (registro[i].nome.empty()){ // verificação de dados.

cout<<"Informe um nome valido: ";

getline(cin, registro[i].nome);

}

cout<<"Idade do registrado "<<(i+1)<<": ";

cin>>registro[i].idade;

cin.ignore();

while (registro[i].idade <= 0){ // verificação de dados.

cout<<endl<<"Informe um numero inteiro valido!"<<endl;

cout<<"Idade do registrado "<<(i+1)<<": ";

cin>>registro[i].idade;

cin.ignore();

}

cout<<endl;

}

}

void ordenacaoRegistroViaQuickSort(int esquerda, int direita, Registro \*registro){ // limite = direita, nas execuções recursivas

int i = esquerda, j = direita;

Registro x = registro[(i + j)/2];

Registro aux;

movimentacoes++;

do{

comparacoes++;

while (x.idade < registro[j].idade){

j--;

comparacoes++;

}

comparacoes++;

while (x.idade > registro[i].idade){

i++;

comparacoes++;

}

if (i <= j){

aux = registro[i];

registro[i] = registro[j];

registro[j] = aux;

movimentacoes += 3;

i++;

j--;

}

} while (i <= j);

if (j > esquerda){

ordenacaoRegistroViaQuickSort(esquerda, j, registro);

chamadasRecursivas++;

}

if (i < direita){

ordenacaoRegistroViaQuickSort(i, direita, registro);

chamadasRecursivas++;

}

}

void impressaoRegistrosOrdenados(int limite, Registro \*registro){

cout<<endl;

cout<<"Impressao dos dados ordenados, via QuickSort:"<<endl;

for (int i = 0; i < limite; i++){

cout<<"Nome do registrado "<<(i+1)<<": "<<registro[i].nome<<endl;

cout<<"Idade do registrado "<<(i+1)<<": "<<registro[i].idade<<endl;

cout<<endl;

}

}

int main(){

int N;

cout<<"Informe o total de registros a serem cadastrados: ";

cin>>N;

cin.ignore();

cout<<endl;

Registro \*registro = new Registro[N];

preencherRegistro(N, registro);

ordenacaoRegistroViaQuickSort(0, N - 1, registro);

impressaoRegistrosOrdenados(N, registro);

cout<<"Movimentacoes: "<<movimentacoes<<endl;

cout<<"Comparacoes: "<<comparacoes<<endl;

cout<<"Chamadas recursivas: "<<chamadasRecursivas<<endl;

delete[] registro;

/\*

Para o vetor [45 56 12 43 95 19 8 67] (idades associadas), houve:

Movimentacoes: 29

Comparacoes: 22

Chamadas recursivas: 4

Para o vetor [8 12 19 43 45 56 67 95] (idade associadas), houve:

Movimentacoes: 16

Comparacoes: 21

Chamadas recursivas: 3

Para o vetor [95 67 56 45 43 19 12 8] (idades associadas), houve:

Movimentacoes: 29

Comparacoes: 24

Chamadas recursivas: 4

\*/

return 0;

}

1. Foi – se utilizado como contadores as variáveis inteiras movimentações, comparações e chamadasRecursivas, cujos nomes são, por si só, autoexplicativos. Sendo assim, tem – se esta aplicação como exemplo da correta execução do algoritmo de ordenação solicitado pela atividade:

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

1. Tem - se diferentes dados para os diferentes vetores solicitados. Logo, vejamos caso a caso:
2. **Vetor de idades: [45, 56, 12, 43, 95, 19, 8, 67]:**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

1. **Vetor de idades: [8, 12, 19, 43, 45, 56, 67, 95]:**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

1. **Vetor de idades: [95, 67, 56, 45, 43, 19, 12, 8]:**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

**Árvores recursivas associadas:**

Texto, Carta

Descrição gerada automaticamente