

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:

- a) 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;

}

- b) Foi – se utilizado como contadores as variáveis inteiras movimentações, comparações e chamadas Recursivas, 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:

```
Informe o total de registros a serem cadastrados: 8

Nome do registrado 1: Clever Farias Brito Albuquerque Josefino Estevao Da Silva Pereira Barreto Santos Meneguel
Idade do registrado 1: 95

Nome do registrado 2: Tinos
Idade do registrado 2: 45

Nome do registrado 3: Jhon Steven McQueen Da Silva Pereira
Idade do registrado 3: 19

Nome do registrado 4: Bernardo Bondoso
Idade do registrado 4: 12

Nome do registrado 5: Robson De Carvalho Melo Santos Pereira
Idade do registrado 5: 43

Nome do registrado 6: Zhang
Idade do registrado 6: 67

Nome do registrado 7: Mariana Andrade Andy Adaminheguebe
Idade do registrado 7: 8

Nome do registrado 8: Marcos Alonso Cosmo
Idade do registrado 8: 56
```

```
Impressao dos dados ordenados, via QuickSort:
Nome do registrado 1: Mariana Andrade Andy Adaminhegube
Idade do registrado 1: 8

Nome do registrado 2: Bernardo Bondoso
Idade do registrado 2: 12

Nome do registrado 3: Jhon Steven McQueen Da Silva Pereira
Idade do registrado 3: 19

Nome do registrado 4: Robson De Carvalho Melo Santos Pereira
Idade do registrado 4: 43

Nome do registrado 5: Tinos
Idade do registrado 5: 45

Nome do registrado 6: Marcos Alonso Cosmo
Idade do registrado 6: 56

Nome do registrado 7: Zhang
Idade do registrado 7: 67

Nome do registrado 8: Clever Farias Brito Albuquerque Josefino Estevao Da Silva Pereira Barreto Santos Meneguel
Idade do registrado 8: 95

Movimentacoes: 31
Comparacoes: 33
Chamadas recursivas: 6
```

c) Tem - se diferentes dados para os diferentes vetores solicitados. Logo, vejamos caso a caso:

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

Informe o total de registros a serem cadastrados: 8

Idade do registrado 1: 45

Idade do registrado 2: 56

Idade do registrado 3: 12

Idade do registrado 4: 43

Idade do registrado 5: 95

Idade do registrado 6: 19

Idade do registrado 7: 8

Idade do registrado 8: 67

Impressao dos dados ordenados, via QuickSort:

Nome do registrado 1:

Idade do registrado 1: 8

Nome do registrado 2:

Idade do registrado 2: 12

Nome do registrado 3:

Idade do registrado 3: 19

Nome do registrado 4:

Idade do registrado 4: 43

Nome do registrado 5:

Idade do registrado 5: 45

Nome do registrado 6:

Idade do registrado 6: 56

Nome do registrado 7:

Idade do registrado 7: 67

Nome do registrado 8:

Idade do registrado 8: 95

Movimentacoes: 29

Comparacoes: 22

Chamadas recursivas: 4

(2) Vetor de idades: [8, 12, 19, 43, 45, 56, 67, 95]:

Informe o total de registros a serem cadastrados: 8

Idade do registrado 1: 8

Idade do registrado 2: 12

Idade do registrado 3: 19

Idade do registrado 4: 43

Idade do registrado 5: 45

Idade do registrado 6: 56

Idade do registrado 7: 67

Idade do registrado 8: 95

```
Impressao dos dados ordenados, via QuickSort:
Nome do registrado 1:
Idade do registrado 1: 8

Nome do registrado 2:
Idade do registrado 2: 12

Nome do registrado 3:
Idade do registrado 3: 19

Nome do registrado 4:
Idade do registrado 4: 43

Nome do registrado 5:
Idade do registrado 5: 45

Nome do registrado 6:
Idade do registrado 6: 56

Nome do registrado 7:
Idade do registrado 7: 67

Nome do registrado 8:
Idade do registrado 8: 95

Movimentacoes: 16
Comparacoes: 21
Chamadas recursivas: 3
```

(3) Vetor de idades: [95, 67, 56, 45, 43, 19, 12, 8]:

```
Informe o total de registros a serem cadastrados: 8

Idade do registrado 1: 95

Idade do registrado 2: 67

Idade do registrado 3: 56

Idade do registrado 4: 45

Idade do registrado 5: 43

Idade do registrado 6: 19

Idade do registrado 7: 12
```

Impressao dos dados ordenados, via QuickSort:

Nome do registrado 1:

Idade do registrado 1: 8

Nome do registrado 2:

Idade do registrado 2: 12

Nome do registrado 3:

Idade do registrado 3: 19

Nome do registrado 4:

Idade do registrado 4: 43

Nome do registrado 5:

Idade do registrado 5: 45

Nome do registrado 6:

Idade do registrado 6: 56

Nome do registrado 7:

Idade do registrado 7: 67

Nome do registrado 8:

Idade do registrado 8: 95

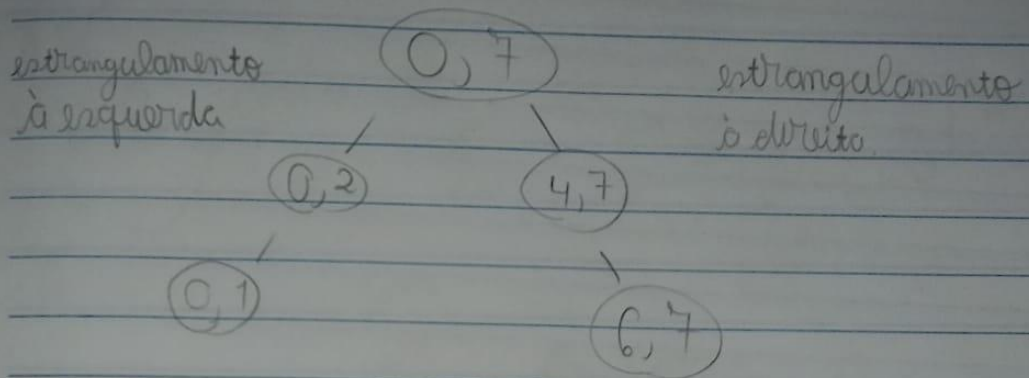
Movimentacoes: 29

Comparacoes: 24

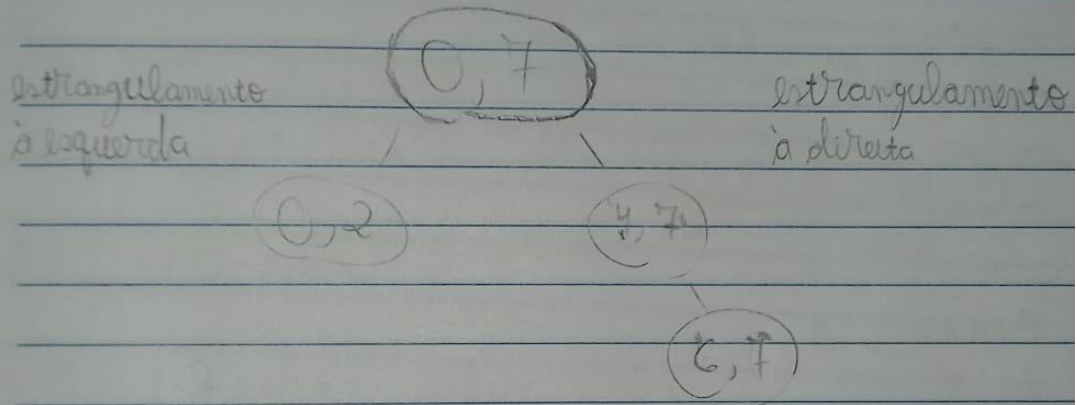
Chamadas recursivas: 4

Árvores recursivas associadas:

(1) Vetor: (45, 56, 12, 43, 95, 19, 8, 67)



(2) Vetor: (8, 12, 19, 43, 45, 56, 67, 95)



(3) Vetor: (95, 67, 56, 45, 43, 19, 12, 8)

