

Selection Sort e Insertion Sort

Ciência da Computação
Laboratório de Ordenação e Pesquisa
Prof. M.Sc. Elias Gonçalves

Atividades

- Utilizando **selection sort** e **Insertion sort**:
 - Preencha um vetor com o nome de 10 pessoas. Crie um programa para ordená-lo e mostrá-lo em ordem crescente e decrescente;
 - Preencha um vetor com salários de 10 pessoas. Ordene-o e mostre-o em ordem crescente e decrescente;

Atividades

```
void selection_sort(char v[][TAM_STR], int tam){  
    int i;  
    for(i=0; i<tam; i++){  
        int min = minimo_indice(v, i, tam-1);  
        if(min != i)  
            troca_elemento(v, i, min);  
    }  
}
```

Atividades

```
int minimo_indice(char v[][TAM_STR], int i, int tam){
    int min = i;
    int aux;
    char min_str[TAM_STR];
    strcpy(min_str, v[i]);

    for(aux = i+1; aux <= tam; aux++){
        if(strcmp(min_str, v[aux]) > 0){
            strcpy(min_str, v[aux]);
            min = aux;
        }
    }

    return min;
}

void troca_elemento(char v[][TAM_STR], int i, int min){
    char aux[TAM_STR];
    strcpy(aux, v[i]);
    strcpy(v[i], v[min]);
    strcpy(v[min], aux);
}
```

Atividades

```
void selection_sort(float v[], int tam){  
    int i;  
    for(i=0; i<tam; i++){  
        int min = minimo_indice(v, i, tam-1);  
        if(min != i)  
            troca_elemento(v, i, min);  
    }  
}
```

Atividades

```
int minimo_indice(float v[], int i, int tam){  
    int min = i;  
    int aux;  
  
    for(aux = i+1; aux <= tam; aux++){  
        if(v[min] > v[aux])  
            min = aux;  
    }  
  
    return min;  
}  
  
void troca_elemento(float v[], int i, int min){  
    float aux = v[i];  
    v[i] = v[min];  
    v[min] = aux;  
}
```

Atividades

```
void insertion_sort(char v[][TAM_STR], int tam){
    int i;
    for(i=1; i<tam; i++){
        char elemento[TAM_STR];
        strcpy(elemento, v[i]);
        int aux = i-1;
        while( aux >= 0 && (strcmp(v[aux], elemento) > 0) ){
            strcpy(v[aux+1], v[aux]);
            aux = aux-1;
        }
        strcpy(v[aux+1], elemento);
    }
}
```


Atividades

```
void insertion_sort(float v[], int tam){  
    int i;  
    for(i=1; i<tam; i++){  
        float elemento = v[i];  
        int aux = i-1;  
        while(aux >= 0 && v[aux] > elemento){  
            v[aux+1] = v[aux];  
            aux = aux-1;  
        }  
        v[aux+1] = elemento;  
    }  
}
```


Atividades

- Utilizando **selection sort** e **Insertion sort**:
 - Utilize uma função para calcular o tempo de execução de cada método de ordenação gastou.
 - Implemente uma função para contar o número de comparações que cada método realizou.
 - Faça uma alteração na função que ordena String em cada método de ordenação de modo que possa receber o nome e o sobrenome de cada pessoa e oferecer a opção de ordenar por nome e por sobrenome.

Biblioteca Virtual

CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas.
Introdução a Estruturas de Dados com Técnicas de Programação em C (Capítulo 11);

DROZDEK, Adam. **Estrutura de dados e algoritmos em c++**
(Capítulo 9);

MARKENZON, Lilian; SZWARCFITER, Jorge Luiz. **Estruturas de Dados e seus Algoritmos** (Capítulos: 7, 11 e 12);

PINTO, Rafael Albuquerque. **Estrutura de Dados** (Páginas 155 a 177).