

Nome: Matheus Goulart Ranzani
RA: 800278

Algoritmos de Ordenação

Código do arquivo ordenacoes.h

```
// Selection Sort (ordenação por seleção)
void selection_sort(int vetor[], int tamanho) {
    int i, j, menor, aux;

    for (i = 0; i < tamanho - 1; i++) {
        menor = i;

        for (j = i + 1; j < tamanho; j++) {
            if (vetor[j] < vetor[menor]) {
                menor = j;
            }
        }

        aux = vetor[i];
        vetor[i] = vetor[menor];
        vetor[menor] = aux;
    }
}

// Insertion Sort (ordenação por inserção)
void insertion_sort(int vetor[], int tamanho) {
    int i, j, aux;

    for (j = 1; j < tamanho; j++) {
        aux = vetor[j];

        for (i = j; (i > 0) && (aux < vetor[i - 1]); i--) {
            vetor[i] = vetor[i - 1];
        }

        vetor[i] = aux;
    }
}
```

Código do arquivo usa_ordenacoes.c

```
#include "ordenacoes.h"

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define TAMANHO 10

void inicia_vetor_sorteado(int vetor[]) {
    srand(time(NULL));

    for (int i = 0; i < TAMANHO; i++) {
        vetor[i] = rand() % 10 + 1;
    }
}

void imprime_vetor(int vetor[]) {
    for (int i = 0; i < TAMANHO; i++) {
        printf("%d ", vetor[i]);
    }

    printf("\n");
}

void copia_vetor(int vetor_copia[], int vetor_original[]) {
    for (int i = 0; i < TAMANHO; i++) {
        vetor_copia[i] = vetor_original[i];
    }
}

int main() {
    int vetor[TAMANHO], vetor_copia[TAMANHO];

    inicia_vetor_sorteado(vetor);
    copia_vetor(vetor_copia, vetor);

    printf("Vetor gerado aleatoriamente de tamanho 10:\n");
    imprime_vetor(vetor);
    selection_sort(vetor, TAMANHO);
    printf("Vetor ordenado utilizando Selection Sort:\n");
    imprime_vetor(vetor);

    printf("\nVetor copia do vetor original:\n");
    imprime_vetor(vetor_copia);

    printf("Vetor copia ordenado utilizando Insertion Sort:\n");
    insertion_sort(vetor_copia, TAMANHO);
}
```

```
    imprime_vetor(vetor_copia);  
  
    return 0;  
}
```

Comparação de desempenho

Tanto a ordenação por seleção (Selection Sort) quanto a ordenação por inserção (Insertion Sort) são capazes de ordenar os elementos de um vetor. Na busca por seleção, em qualquer caso, seu desempenho é da ordem de $O(n^2)$. Já na busca por inserção, no melhor caso, seu desempenho é da ordem de $O(n)$, e, no pior caso, da ordem de $O(n^2)$.

Imagem da execução do programa

```
C:\Users\mathe\OneDrive\Área de Trabalho\UFSCar\ENPE 4\AED1\Frequências\F11  
>usa_ordenacoes  
Vetor gerado aleatoriamente de tamanho 10:  
9 3 9 2 6 1 7 2 7 10  
Vetor ordenado utilizando Selection Sort:  
1 2 2 3 6 7 7 9 9 10  
  
Vetor copia do vetor original:  
9 3 9 2 6 1 7 2 7 10  
Vetor copia ordenado utilizando Insertion Sort:  
1 2 2 3 6 7 7 9 9 10
```