FAMEPI

Faculdade do Amazonas de Ensino, Pesquisa e Inovação

FUNDAÇÃO MATIAS MACHLINE

Algoritmos e Estrutura de Dados 2

Prof Pedro Corrêa

Objetivo da aula

Utilizar o método de ordenação Selection Sort

Sumário

Sumário

Introdução

A ordenação Selection Sort é a 2ª ordenação mais básica

- A 1ª é a Bubble Sort

Compara cada elemento com os outros, visando encontrar o menor

A complexidade deste algoritmo será sempre $O(n^2)$

Sumário

Sumário

Método Selection Sort (Seleção)

Técnica básica:

- Ache o menor elemento a partir da posição 0.
  Troque então este elemento com o elemento da posição 0.
- Ache o menor elemento a partir da posição 1.
  Troque então este elemento com o elemento da posição 1.
- Ache o menor elemento a partir da posição 2.
  Troque então este elemento com o elemento da posição 2.
- E assim sucessivamente...

Método Selection Sort (Seleção)

Técnica básica:

Exemplo: (5,3,2,1,90,6).

Iteração 0. Acha menor: (5,3,2,1,90,6). Faz troca: (1,3,2,5,90,6).

Iteração 1. Acha menor: (1,3,2,5,90,6). Faz troca: (1,2,3,5,90,6).

Iteração 2. Acha menor: (1,2,3,5,90,6). Faz troca: (1,2,3,5,90,6).

Iteração 3. Acha menor: (1,2,3,5,90,6). Faz troca: (1,2,3,5,90,6).

Iteração 4: Acha menor: (1,2,3,5,90,6). Faz troca: (1,2,3,5,6,90).

Método Selection Sort (Seleção)

Vantagens

Algoritmo simples de ser implementado em comparação aos demais

Não necessita de um vetor auxiliar (in-place)

Por não usar um vetor auxiliar para realizar a ordenação, ele ocupa menos memória

É um dos mais velozes para vetores de tamanhos pequenos

Desvantagens

É um dos mais lentos para vetores de tamanhos grandes

Não é estável

Faz sempre $(n^2 - n) / 2$ comparações, independentemente do vetor estar ordenado ou não

```c
#include <locale.h>
#include <stdlib.h>// para malloc()

void inserir(int *vet, int n){

    int i;

    for( i = 0 ; i < n ; i++){

        printf("\n\n Digite o %dº nr: ", i + 1);

        scanf("%d", &vet[i]);
    }
} // fim inserir

void mostrar(int *vet, int n){

    int i;

    printf("\n\n Conteúdo do vetor: ")

    for( i = 0 ; i < n ; i++)

        printf("%d ", vet[i]);

} // fim mostrar
```

```c
void selectionSort(int *vet, int n){

    int i, j, min, aux;

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

        min = i;

        for ( j = i + 1 ; j < n ; j++ )

            if(vet[j] < vet[min])

                min = j; // acha o índice do menor do vetor

        if (i != min) { // evita trocar na mesma posição

            aux = vet[i];

            vet[i] = vet[min];

            vet[min] = aux;
        }

    }
} // fim selectionSort
```

```c
int main( ) {
    setlocale(LC_ALL, "");

    int n, *vet;

    printf("\n\n Digite o tamanho do vetor:

    scanf("%d", &n);

    // vetor definido dinamicamente:
    vet = (int *) malloc(sizeof(int));

    inserir(vet, n);

    printf("\n\n Vetor Original: ");

    mostrar(vet, n);

    selectionSort(vet, n);

    printf("\n\n Vetor Ordenado: ");

    mostrar(vet, n);

    return(0);
}
```