

**UFMG – Engenharia de Sistemas**

**Lista de Exercício 3**

**Algoritmos de Ordenação e Seleção**

|  |  |  |
| --- | --- | --- |
| **DISCIPLINA:** | ESTRUTURA DE DADOS | |
| **PROFESSOR:** | [Carlos Henrique de Carvalho Teixeira](https://virtual.ufmg.br/20192/user/view.php?id=23049&course=9470) | |
| **ALUNO:** | Antonio Carlos da Anunciação | 2018019443 |

**1.** Mostre um exemplo que mostre que o Método de Seleção não é estável.

*void Selecao(int vetor[], int n) {*

int Min;

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

Min = i;

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

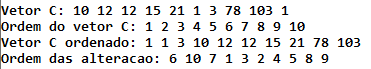
if (vetor[j] < vetor[Min]) Min = j;

Troca(vetor, i, Min);

}

}

*Resultados:*

**

*Podemos perceber de o método de seleção inverteu a ordem das posições originais do item 12, que era 2, 3 e passou a ser 3 e 2.*

**2.** É possível criar uma versão do Método de seleção estável? Justifique sua resposta e em caso de você achar que é possível, mostre como o algoritmo deve ser implementado.

*Sim, para isso devemos guardar a informação da posição original dos itens no vetor e salva-las na posição nova que ela ocupa, depois comparar se houve permutação das posições dos elementos que são iguais e destroca-los, caso houve permutação.*

void SelecaoEstavel(int vetor1[], int vetor2[], int n) {

int Min;

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

Min = i;

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

if (vetor1[j] < vetor1[Min]) Min = j;

Troca(vetor1, i, Min);

Troca(vetor2, i, Min);

}

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

if ( (vetor1[i] == vetor1[i+1]) && (vetor2[i] > vetor2[i+1]) )

Troca(vetor1, i, i+1);

}

**3.** Implemente o Método da Bolha recursivo.

void Troca(int vetor[], int i, int j) {

int c = vetor[i];

vetor[i] = vetor[j];

vetor[j] = c;

}

void Bolha(int vetor[], int n) {

if(n == 1) return;

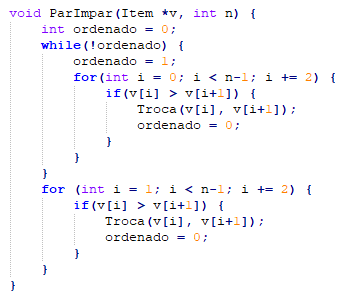
for(int i = 0; i < n - 1; i++)

if(vetor[i] > vetor[i+1]) Troca(vetor, i, i+1);

Bolha(vetor, n-1);

}

**4.** Considere o algoritmo de ordenação apresentado abaixo. Calcule o custo e a ordem de complexidade deste método em termos de comparação e movimentações, avaliando o melhor e o pior caso.



*Melhor caso é o vetor organizado, assim:*

*F(n) = n/2 + n/2 - 1*

*F(n) = n – 1 com uma complexidade de O(n)*

*Neste caso não haverá nenhuma troca e o programa termina sua execução sem executar o laço “while”.*

*Em termos de comparações o pior caso é quando o programa entrar todas as vezes no no “if” da Troca( ), esse caso vai acontecer para o vetor organizado de maneira oposta ao que o programa se propõe a fazer. Deste modo o while será executado (n/2), assim teremos:*

*F(n) = n/2(n – 1)*

*F(n) = (n² - n)/2, nos dando uma complexidade O(n²)*