|  |  |  |  |
| --- | --- | --- | --- |
| Instituto Federal Goiano – Campus Urutaí | | | C:\Users\Phael\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.png |
| Aluno Raphael P. Policena Rosa | Sistemas de Informação  4º Período  Matutino | |
| Docente Júnio César de Lima | | Estrutura de Dados II |

Implementar os seguintes métodos de ordenação:

1. BobbleSort;

2. SelectionSort;

3. InsertionSort;

4. ShellSort.

Criar três vetores de 20 elementos, sendo:

•. Um vetor ordenado de forma crescente;

•. Um vetor em ordem inversa (ordem decrescente);

•. Um vetor aleatório.

Cada método de ordenação deve receber os três vetores como entrada e devolver o vetor ordenado. Para cada execução contar o número de trocas e o número de comparações;

Vetores com 20 elementos;

Vetor Desordenado Aleatório:

18, 4, 10, 18, 10, 10, 17, 0, 8, 19, 7, 14, 2, 14, 1, 0, 7, 15, 16, 10

Vetor Desordenado Decrescente:

20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

Vetor Desordenado Crescente:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19

***Bubble Sort***

|  |  |  |  |
| --- | --- | --- | --- |
|  | Aleatório | Crescente | Decrescente |
| Comparações | 189 | 19 | 190 |
| Trocas | 303 | 0 | 570 |

***Selection Sort***

|  |  |  |  |
| --- | --- | --- | --- |
|  | Aleatório | Crescente | Decrescente |
| Comparações | 190 | 190 | 190 |
| Trocas | 115 | 76 | 176 |

***Insertion Sort***

|  |  |  |  |
| --- | --- | --- | --- |
|  | Aleatório | Crescente | Decrescente |
| Comparações | 190 | 190 | 190 |
| Trocas | 222 | 20 | 400 |

***Shell Sort***

|  |  |  |  |
| --- | --- | --- | --- |
|  | Aleatório | Crescente | Decrescente |
| Comparações | 87 | 42 | 72 |
| Trocas | 84 | 84 | 84 |

***Bubble Sort :***

É lento: trocas ocorrem apenas entre itens adjacentes e ainda executa mais trocas que os algoritmos Insertion Sort, Selection Sort e Shell Sort.

Se o menor item está no final da tabela, serão precisos (n) passos para o colocar na posição correta

***Shell Sort:***

Acelera o algoritmo permitindo trocas entre elementos que estão afastados porem sua usabilidade é melhor em arquivos de tamanho moderado pois sua implementação é simples.

***Insertion sort:***

Insertion Sort usa aproximadamente (N2/4) comparações e (N2/4) pseudo-trocas (movimentos) no caso médio e o dobro destes valores no pior caso.

Quando a sequência a ordenar se encontra parcialmente ordenada, tem um desempenho aproximadamente linear.

***Selection Sort:***

O Algoritmo não é instável e sempre faz N2 comparações, mesmo se o vetor estiver ordenado, no qual é o caso do vetor crescente ali mostrado na tabela e também é muito lento ao ordenar vetores de grande tamanho.