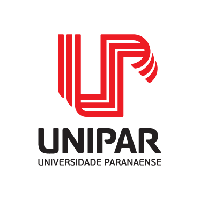
|  |  |  |
| --- | --- | --- |
| |  | | --- | | UNIPAR - Universidade Paranaense Analise e Desenvolvimento de Sistemas | |  | |
|  |
|  |
|  |
|  |
|  |

*Bruno Medeiros de Athaide - RA 00241269*

*Gabriel Sales De Lima – RA 00227249*

**RELATÓRIO** **DE ORDENAÇÃO**

Toledo – Paraná

2023

O **Insertion sort** é um algoritmo eficiente em listas menores e simples, ele executar da esquerda para a direita à medida que ele vai percorrendo a lista vai deixando os elementos à esquerda na ordenação.

Insertion Sort, simples e eficiente quando em pequenas listas.

A medida que o algoritmo avança vai deixando os elementos mais à esquerda ordenados.

Funciona da mesma forma que normalmente utilizamos para ordenar um baralho.

A **Selection Sort** já é um algoritmo que se baseia em passar o menor valor para a primeira posição ou dependendo da ordem a ser seguida, a selection começa pelos menores valores e vem substituindo a sua direita. Se o elemento da posição i for maior que elemento da 2 posição eles trocam de lado, seguindo assim o da posição 2 for mais ou menor que da 3 posição eles vão se alterando até chegar no final do vetor.

Selection Sort sempre vai passar o menor valor para a primeira posição, depois o segundo menor valor e assim sucessivamente;

A partir da primeira posição, ele escolhe qual é o menor número e coloca na primeira posição;

A mesma comparação é feita com o próximo item da lista;

Caso não encontre nenhum número menor, não altera nada.

**Bubble Sort** é o algoritmo mais simples, mas também o menos eficiente, ele funciona da sequente maneira. O elemento da posição i acaba por ser comparado com o outro elemento da posição i + 1. O elemento da posição 2 será comparado ao elemento da posição 3, caso o elemento 2 seja maior que o elemento 2 eles trocam de posição e assim por diante. Assim o vetor terá que percorrer a sequência quantas vezes forem necessárias, por conta dessa sequência acaba tornando o algoritmo ineficiente se for usado em listas muito grandes, tendo em vista que ordenação é simplificar e agilizar o próprio processo.

Bubble Sort é o algoritmo mais simples, e também o menos eficiente;

Cada elemento da posição [i] será comparado com o elemento da posição [i + 1];

Caso o elemento da posição [i] seja maior que o elemento da posição [i + 1], eles trocam de lugar, e assim sucessivamente.

No início do programa é pedido ao usuário a quantidade de vetores que ele deseja implementar, em seguida ele escolhe para qual ordenação (Bubble Sort, Insertion Sort, Selection Sort.) ele deseja ordenar os vetores, para cada um dos métodos é exibido a mensagem ordenada e desordenada, no código por escolha da dupla, foi implementado também quantidade de movimento e quantidade de trocas seguindo com o tempo de execução do código.   
  
Podemos concluir que dentre as três operações de ordenação cada algoritmo funciona de um jeito sendo o bubble Sort o mais simples e o menos eficaz, Insertion sort já é um algoritmo bem mais simples e eficiente quando se é aplicado em pequenas listas, A Selection Sort já é um algoritmo que se baseia em passar o menor valor para a primeira posição ou dependendo da ordem a ser seguida já fazendo uma busca linear no vetor, sendo um dos melhores para se fazer uma busca em caso de já ter uma lista quase ou já ordenada, em listas pequenas a Bubble Sort resolve, até certo ponto, a Insertion e a Selection Sort já são umas das melhores para serem executadas pela velocidade e numero tem termos de troca e quantidade.

já são umas das melhores para serem executadas pela velocidade e numero tem termos de troca e quantidade.