***Insertion Sort:***

O Insertion Sort percorre o vetor da esquerda para a direita, construindo a parte ordenada à medida que avança.

Vantagens:

* Simples de implementar e entender.
* Eficiente para conjuntos de dados pequenos ou quase ordenados.
* Requer menos comparações do que outros algoritmos, como o Selection Sort e o Bubble Sort.

Desvantagens:

* Não é eficiente para grandes conjuntos de dados ou quando o vetor está em ordem inversa.
* Possui uma complexidade de tempo quadrática no pior caso (O(n^2)).

***Selection*** ***Sort***:

O Selection Sort seleciona o menor elemento não ordenado e o coloca na posição correta.

Vantagens:

* Simples de implementar e entender.
* Requer um número fixo de trocas, independentemente do estado do vetor.

Desvantagens:

* Possui uma complexidade de tempo quadrática em todos os casos (O(n^2)).
* Não é eficiente para grandes conjuntos de dados, mesmo quando o vetor já está ordenado.

***Bubble*** ***Sort***:

O Bubble Sort compara pares adjacentes e troca os elementos se estiverem na ordem errada.

Vantagens:

* Simples de implementar e entender.
* Requer um número fixo de trocas, independentemente do estado do vetor.

Desvantagens:

* Possui uma complexidade de tempo quadrática em todos os casos (O(n^2)).
* Não é eficiente para grandes conjuntos de dados, mesmo quando o vetor já está ordenado.
* Realiza muitas trocas, mesmo quando o vetor já está ordenado.

Em resumo, Insert Sort mostrou ser mais eficientes em todos os testes realizados.