

Universidade Evangélica de Goiás – UniEVANGÉLICA
CURSO DE ENGENHARIA DE SOFTWARE

Atividade pré-aula (semana 03)

Antônio Claudio Ferreira Filho

Matrícula: 2110854

Anápolis - GO

2023

Antônio Claudio Ferreira Filho

Atividade pré-aula (semana 03)

Trabalho apresentado à disciplina de Árvores e grafos como requisito parcial para aprovação.

Anápolis – GO

2023

Faça uma comparação entre os métodos de ordenação BubbleSort, SelectionSort e InsertionSort, e escreva resumo sobre suas complexidades, no melhor e pior caso. Demonstre em quais situações seria mais ou menos pertinentes a utilização de um método específico. Utilize exemplos práticos:

Os algoritmos de ordenação Bubble Sort, Selection Sort e Insertion Sort são amplamente utilizados para ordenar listas em ordem crescente ou decrescente. Vamos comparar suas complexidades no melhor e pior caso e analisar em quais situações cada um pode ser mais ou menos eficiente.

1. Bubble Sort:

- a. O Bubble Sort consiste em percorrer a lista várias vezes, comparando elementos adjacentes e trocando suas posições se necessário. Ele é mais eficiente quando a lista já está quase ordenada ou quando a lista é pequena. Porém, quando a lista está completamente desordenada, o Bubble Sort é menos eficiente que outros algoritmos, como o Insertion Sort.
- b. Complexidade no melhor caso: $O(n)$
- c. Complexidade no pior caso: $O(n^2)$
- d. Exemplo prático: suponha que temos a seguinte lista: 3, 5, 7, 8, 10, 12, 15. Neste caso, o Bubble Sort teria complexidade $O(n)$ no melhor caso, pois a lista já está quase ordenada. Na primeira iteração, ele compararia os elementos 3 e 5, 5 e 7, 7 e 8, 8 e 10, 10 e 12, e 12 e 15, sem realizar nenhuma troca.

2. Selection Sort:

- a. O Selection Sort percorre a lista várias vezes em busca do menor elemento, que é colocado na posição correta. Ele é eficiente quando a lista é pequena, mas menos eficiente que outros algoritmos para listas grandes. Além disso, o Selection Sort é ineficiente para listas com muitos elementos repetidos, pois ele não é estável e pode trocar a posição de elementos iguais.
- b. Complexidade no melhor caso: $O(n^2)$
- c. Complexidade no pior caso: $O(n^2)$
- d. Exemplo prático: suponha que temos a seguinte lista: 10, 8, 6, 4, 2. Neste caso, o Selection Sort teria complexidade $O(n^2)$ no pior caso, pois teria que percorrer toda a lista para encontrar o menor elemento em cada iteração. Na primeira iteração, ele encontraria o elemento 2 e o colocaria na primeira posição. Na segunda iteração, encontraria o elemento 4 e o colocaria na segunda posição, e assim por diante, até que a lista estivesse completamente ordenada.

3. Insertion Sort:

- a. O Insertion Sort consiste em percorrer a lista e inserir cada elemento em sua posição correta, mantendo a lista parcialmente ordenada. Ele é mais eficiente para listas pequenas ou quase ordenadas, e é mais estável que o Selection Sort e o Bubble Sort.
- b. Complexidade no melhor caso: $O(n)$
- c. Complexidade no pior caso: $O(n^2)$