

**CESED – CENTRO DE ENSINO SUPERIOR E DESENVOLVIMENTO
CENTRO UNIVERSITÁRIO UNIFACISA**

**APLICAR ALGORITMOS DE APOIO À PESQUISA OPERACIONAL
PROF. JONHNANTHAN OLIVEIRA**

**ALGORITMO UTILIZADO PARA ORDENAÇÃO DA BIBLIOTECA
BUBBLE SORT**

**ALUNOS DO GRUPO
GUSTAVO TOMIO MAGALHÃES KUBO
SÉRGIO MAGNO CASTOR PINHEIRO
THIAGO LIMEIRA DE ALENCAR**

**CAMPINA GRANDE – PB
2024.2**

Bubble Sort: Algoritmo utilizado para ordenação

O **Bubble Sort** (ou Ordenação por Flutuação) é um algoritmo de ordenação muito simples e intuitivo. Ele é chamado assim porque funciona de forma semelhante ao modo como as bolhas sobem até a superfície da água. Em cada rodada do algoritmo, o maior elemento vai "subindo" até o fim da lista, enquanto os menores elementos vão se ajustando antes dele.

Passo a Passo do Funcionamento:

O Bubble Sort percorre repetidamente a lista, comparando elementos adjacentes e trocando-os de posição se estiverem fora de ordem. Vamos entender como ele funciona em detalhes:

1. **Comparação de Pares:** O algoritmo começa comparando o primeiro elemento com o segundo. Se o primeiro é maior que o segundo, eles trocam de lugar. Em caso contrário, permanecem onde estão.
2. **Movimento Progressivo:** Depois de comparar o primeiro e o segundo, o algoritmo avança para o segundo e o terceiro elementos, e assim por diante, até alcançar o fim da lista.
3. **Repetição:** O processo de comparação e troca é repetido várias vezes em toda a lista. Cada vez que o algoritmo passa pela lista, o maior elemento "flutua" para o fim, ficando na posição correta.
4. **Parada do Algoritmo:** A ordenação termina quando o algoritmo passa pela lista sem precisar fazer nenhuma troca. Isso significa que todos os elementos já estão ordenados.

Exemplo

Vamos ver um exemplo rápido de como o Bubble Sort funcionaria em uma lista com alguns números. Suponha que temos a lista [5, 3, 8, 4, 2] e queremos ordená-la em ordem crescente.

Primeira Passagem

- Comparação entre 5 e 3 → trocam de lugar → lista fica: [3, 5, 8, 4, 2]
- Comparação entre 5 e 8 → sem troca → lista permanece: [3, 5, 8, 4, 2]
- Comparação entre 8 e 4 → trocam de lugar → lista fica: [3, 5, 4, 8, 2]
- Comparação entre 8 e 2 → trocam de lugar → lista fica: [3, 5, 4, 2, 8]

Aqui, o maior número (8) já chegou à posição final, que é o último elemento.

Segunda Passagem

- Comparação entre 3 e 5 → sem troca → lista permanece: [3, 5, 4, 2, 8]
- Comparação entre 5 e 4 → trocam de lugar → lista fica: [3, 4, 5, 2, 8]
- Comparação entre 5 e 2 → trocam de lugar → lista fica: [3, 4, 2, 5, 8]

Após essa segunda passagem, o segundo maior número (5) já está na posição correta, e assim por diante. Esse processo se repete até que toda a lista esteja ordenada.

Análise de Eficiência:

O Bubble Sort é eficiente para listas pequenas ou que já estão quase ordenadas. No entanto, para listas maiores, ele não é tão eficiente porque precisa comparar cada par de elementos várias vezes, o que o torna mais lento em comparação com outros algoritmos de ordenação.

A complexidade de tempo do Bubble Sort é:

- **Melhor Caso:** $O(n)$ - quando a lista já está ordenada e não são necessárias trocas.
- **Caso Médio e Pior Caso:** $O(n^2)$ - quando a lista está completamente desordenada.

Vantagens e Desvantagens:

Vantagens

Simplicidade: fácil de entender e implementar.

Utilidade para listas pequenas ou quase ordenadas.

Desvantagens

Ineficiente para listas grandes, especialmente em comparação com algoritmos mais avançados, como o Merge Sort ou o Quick Sort.

Maior número de comparações e trocas.

Conclusão:

O Bubble Sort é um algoritmo de ordenação básico que pode ser usado para organizar pequenas listas de forma fácil e rápida. Ele foi escolhido neste projeto pela simplicidade e adequação para listas pequenas, como a coleção de livros na nossa aplicação. Por ser um algoritmo fácil de entender, ele também facilita o aprendizado dos conceitos básicos de ordenação.