

## Análise Visual dos Algoritmos

- Para criar gráficos ilustrando como os algoritmos de ordenação **Merge Sort**, **Quick Sort** e **Selection Sort** reorganizam os elementos a cada etapa, utilizaremos o Python com a biblioteca **matplotlib** para visualizar o processo de ordenação.
- **1. Merge Sort**
- O Merge Sort segue a estratégia **dividir para conquistar**. Ele divide recursivamente a lista ao meio até que cada sublista tenha apenas um elemento. Depois, ele começa a combinar essas sublistas ordenadas.
- **2. Quick Sort**
- O Quick Sort escolhe um pivô e particiona a lista em elementos menores e maiores que o pivô. A ordenação é recursiva e ocorre em torno do pivô escolhido.
- **3. Selection Sort**
- O Selection Sort encontra o menor elemento em cada iteração e o coloca na posição correta, repetindo o processo para o restante da lista.

## Como Funciona a Visualização:

- Para cada um dos algoritmos de ordenação, após cada passo significativo (como fusão no Merge Sort, troca no Quick Sort e Selection Sort), o estado atual da lista é exibido como um gráfico de barras.
- O eixo X representa o índice dos elementos e o eixo Y representa o valor do elemento na lista.
- As barras são atualizadas a cada iteração para mostrar a transformação da lista ao longo do tempo.

## Resultados Esperados:

Ao rodar o código, você verá uma série de gráficos para cada algoritmo, mostrando o progresso da ordenação da lista. Cada algoritmo terá uma série de gráficos, cada um representando o estado da lista após uma etapa importante de ordenação.

## Exemplo de Saída (Merge Sort):

1. **Passo 1:** Divisão inicial
2. **Passo 2:** Fusão das sublistas
3. **Passo Final:** Lista completamente ordenada

