Análise da Complexidade do Quick Sort

Complexidade de Tempo

O Quick Sort é um algoritmo de ordenação baseado no conceito de divisão e conquista. Ele divide o vetor em subvetores ao redor de um pivô e os ordena recursivamente.

1. Melhor Caso:
- O pivô divide o vetor de forma equilibrada, ou seja, os subvetores têm tamanhos aproximadamente iguais.
- O número de comparações em cada nível é O(n), e a profundidade da recursão é O(log n).
- Complexidade: $T(n) = 2T(n/2) + O(n) \rightarrow O(n \log n)$.

2. Caso Médio:
- Em média, o pivô divide o vetor de forma aceitável (não necessariamente equilibrada).
- A complexidade esperada é semelhante ao melhor caso, pois a divisão ocorre em níveis aproximadamente logarítmicos.
- Complexidade: O(n log n).

3. Pior Caso:
- O pivô escolhido é o maior ou o menor elemento, levando a divisões altamente desequilibradas.
- Nesse caso, o Quick Sort executa n - 1 comparações no primeiro nível, n - 2 no segundo, e assim por diante.
- Complexidade: $T(n) = T(n-1) + O(n) \rightarrow O(n^2)$.

Complexidade de Espaço

A complexidade de espaço do Quick Sort depende principalmente da pilha de recursão:

1. Melhor Caso:
- Quando o pivô divide o vetor de forma equilibrada, a profundidade máxima da pilha de recursão é O(log n).
- Espaço adicional para variáveis temporárias: O(1).

2. Caso Médio:
- Semelhante ao melhor caso, a profundidade da pilha de recursão é O(log n).
- Espaço adicional: O(1).

3. Pior Caso:
- Quando o pivô divide o vetor de forma altamente desequilibrada, a pilha de recursão alcança O(n) níveis.
- Espaço adicional: O(1).