

made for free at coggle.it

Eficiência e complexidade: complexidade de tempo O(n log n), tornando-o eficiente para grandes conjuntos de dados.

Pode degradar para O(n^2) no **pior caso**, como quando a lista já está ordenada ou quase ordenada.

recursiva, o Quicksort também pode ser implementado de **forma iterativa usando uma pilha**.

Divide o problema em subproblemas menores e resolve-os de forma recursiva

Embora geralmente seja implementado de forma

Escolha do pivô: seleciona um elemento da lista como pivô. Geralmente, o pivô é escolhido como o elemento do meio ou aleatoriamente

Elementos rearranjados: elementos menores que o pivô: à esquerda e os elementos maiores: à direita. O pivô é colocado em sua posição final.

Processo de partição é repetido recursivamente para as sublistas à esquerda e à direita do pivô até que todas as partições tenham apenas um elemento.

Otimização: existem várias otimizações possíveis para o Quicksort, como a escolha inteligente do pivô (por exemplo, o pivô de mediana de três), que visam melhorar o desempenho e evitar o pior caso.

Algoritmo eficiente de ordenação: estratégia de

divisão e conquista para classificar elementos em

uma lista

Quicksort

Amplamente utilizado em **sistemas de classificação e bancos de dados** devido à sua eficiência em média e à sua implementação simples.