

1)

O método de "divisão e conquista" é um algoritmo de solução de problemas que divide um problema maior em subproblemas menores e mais simples, e então resolve cada subproblema separadamente. O resultado obtido para cada subproblema é combinado para formar a solução final para o problema original.

Divisão: o problema original é dividido em subproblemas menores e mais simples. Essa divisão deve ser feita de forma que cada subproblema possa ser resolvido independentemente e sem sobreposição.

Conquista: cada subproblema é resolvido separadamente usando o mesmo algoritmo de "divisão e conquista". Isso geralmente envolve recursividade, ou seja, a aplicação do mesmo algoritmo aos subproblemas gerados na etapa anterior.

Combinar: os resultados obtidos para cada subproblema são combinados para formar a solução final para o problema original. Essa combinação deve ser feita de forma a garantir que a solução final seja correta e completa.

2)

O MergeSort e o QuickSort são algoritmos de ordenação baseados em "divisão e conquista". A principal diferença é que o MergeSort divide a lista em duas partes iguais e as combina ordenadas, enquanto o QuickSort usa um pivô para dividir a lista em elementos menores e maiores que ele, ordenando cada parte recursivamente. Outras diferenças incluem a estabilidade, desempenho e uso recomendado para diferentes tipos de dados.

3)

O ShellSort é uma adaptação do método de insertion sort. A melhoria proposta pelo ShellSort é a ideia de dividir a lista em subgrupos menores e aplicar o método de inserção direta em cada subgrupo, reduzindo assim o número de comparações e trocas necessárias para ordenar a lista como um todo. A escolha do tamanho e da quantidade de subgrupos é determinada por uma sequência específica de gaps, que são usadas para definir os subgrupos e reduzir o tamanho deles a cada iteração do algoritmo.

4)

a)

Divisão em subvetores menores:

[7, 8, 3, 4, 2, 1, 5]

[7, 8, 3, 4] [2, 1, 5]

[7, 8] [3, 4] [2, 1] [5]

[7] [8] [3] [4] [2] [1] [5]

Ordenação de cada subvetor:

[7] [8] [3] [4] [2] [1] [5]

[7, 8] [3, 4] [1, 2] [5]

Mesclagem dos subvetores ordenados:

[3, 4, 7, 8] [1, 2, 5]

[1, 2, 3, 4, 5, 7, 8]

b)

Escolha do pivô:

[7, 8, 3, 4, 2, 1, 5]

Pivô = 5

Divisão em subvetores menores:

[3, 4, 2, 1] [5] [7, 8]

Ordenação de cada subvetor:

[3, 4, 2, 1]

Pivô = 1

[1] [3, 4, 2]

Pivô = 2

[1] [2] [3, 4]

[3, 4]

Pivô = 4

[3] [4]

[1, 2, 3, 4]

[7, 8]

Mesclagem dos subvetores ordenados:

[1, 2, 3, 4] [5] [7, 8]

Resultado final:

[1, 2, 3, 4, 5, 7, 8]

c)

Definição dos gaps:

[7, 8, 3, 4, 2, 1, 5]

gap = 3

Ordenação dos subvetores com gap = 3:

[7, 4, 1] [8, 2, 5] [3]

[1, 4, 7] [2, 5, 8] [3]

Definição dos gaps:

[1, 4, 7, 2, 5, 8, 3]

gap = 1

Ordenação dos subvetores com gap = 1:

[1, 4, 7, 2, 5, 8, 3]

[1, 2, 3, 4, 5, 7, 8]

Resultado final:

[1, 2, 3, 4, 5, 7, 8]