Merge Sort

A classificação de mesclagem é outro algoritmo de classificação que se enquadra na categoria da técnica de [divisão e conquista](https://www.javatpoint.com/divide-and-conquer-introduction) . É uma das melhores técnicas de classificação que constrói com sucesso um algoritmo recursivo.

## Estratégia de divisão e conquista

Nesta técnica, segmentamos um problema em duas metades e os resolvemos individualmente. Depois de encontrar a solução de cada metade, nós as mesclamos de volta para representar a solução do problema principal.

Suponha que tenhamos um array **A** , de modo que nossa principal preocupação seja classificar a subseção, que começa no índice **pe** termina no índice **r** , representado por **A [p..r]** .

**Dividir**

Se assumido que **q** é o ponto central em algum lugar entre **p** e **r** , então iremos fragmentar o subarray **A [p..r]** em dois arrays **A [p..q]** e **A [q + 1, r]** .

**Conquistar**

Depois de dividir as matrizes em duas metades, a próxima etapa é conquistar. Nesta etapa, classificamos individualmente ambos os subarrays **A [p..q]** e **A [q + 1, r]** . Caso não tenhamos atingido a situação de base, seguimos novamente o mesmo procedimento, ou seja, segmentamos ainda mais esses subarranjos e, em seguida, classificá-los separadamente.

**Combinar**

Como quando a etapa de base é adquirida pela etapa de conquista, obtemos com sucesso nossas **submatrizes** ordenadas **A [p..q]** e **A [q + 1, r]** , após o que as **fundimos de** volta para formar uma nova matriz ordenada **[p. .r]** .

## Algoritmo de mesclagem de classificação

A função MergeSort continua dividindo um array em duas metades até que uma condição seja satisfeita onde tentamos realizar MergeSort em um subarray de tamanho 1, ou seja, **p == r** .

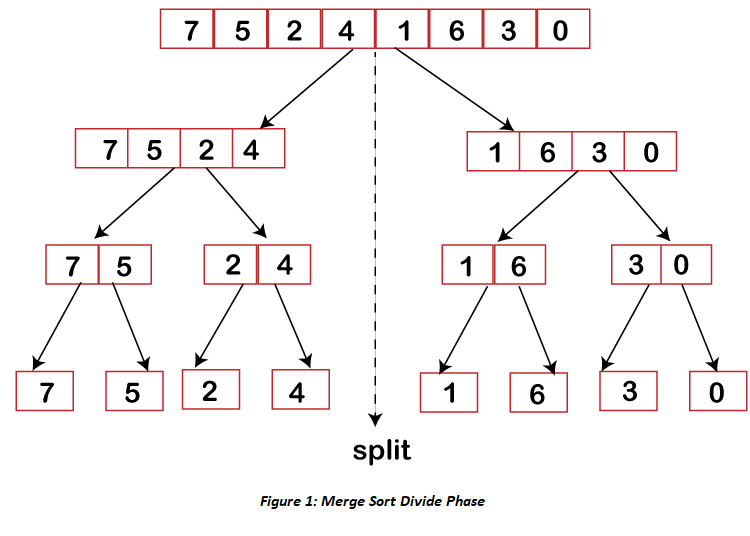
E então, ele combina as submatrizes classificadas individualmente em matrizes maiores até que toda a matriz seja mesclada.

1. ALGORITHM-MERGE SORT
2. 1. Se p **< r**
3. 2. Então q → (p + r) / 2
4. 3. MERGE-SORT (A, p, q)
5. 4. MERGE-SORT (A, q + 1, r)
6. 5. FUSÃO (A, p, q, r)

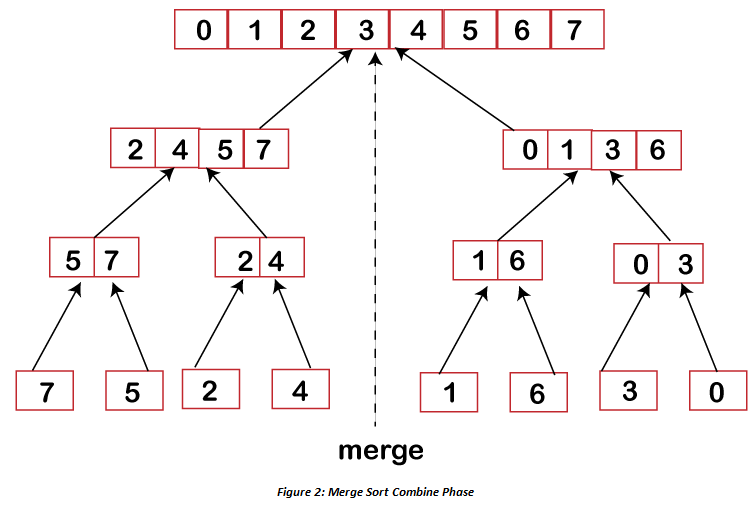
Aqui chamamos **MergeSort (A, 0, length (A) -1)** para classificar o array completo.

Como você pode ver na imagem abaixo, o algoritmo de classificação por mesclagem divide recursivamente o array em metades até que a condição básica seja satisfeita, onde ficamos com apenas 1 elemento no array. E então, a função de mesclagem seleciona os submatrizes classificados e os mescla de volta para classificar o array inteiro.

A figura a seguir ilustra o procedimento de divisão (divisão).



1. FUNÇÕES: MERGE (A, p, q, r)
3. 1. n  1  =  q -p + 1
4. 2. n  2 =  r -q
5. 3. criar matrizes [1 ..... n 1 + 1] e R [1 ..... n 2 +1]
6. 4. para i ← 1 a n 1
7. 5. faça [i] ← A [p + i-1]
8. 6. para j ← 1 a n2
9. 7. faça R [j] ← A [q + j]
10. 8. L [n 1+ 1] ← ∞
11. 9. R [n 2+ 1] ← ∞
12. 10. I ← 1
13. 11. J ← 1
14. 12. Para k ← p para r
15. 13. Faça se L [i] ≤ R [j]
16. 14. então A [k] ← L [i]
17. 15. i ← i +1
18. 16. mais A [k] ← R [j]
19. 17. j ← j + 1



### A etapa de mesclagem da classificação de mesclagem

Principalmente, o algoritmo recursivo depende de um caso base, bem como de sua capacidade de mesclar de volta os resultados derivados dos casos base. A ordenação de mesclagem não é um algoritmo diferente, apenas o fato de que a etapa de **mesclagem** possui mais importância.

Para qualquer problema, a etapa de mesclagem é uma solução que combina as duas listas classificadas individualmente (matrizes) para construir uma grande lista classificada (matriz).

O algoritmo de classificação por mesclagem mantém três ponteiros, ou seja, um para os dois arrays e o outro para preservar o índice atual do array ordenado final.

1. Você alcançou o final da matriz?
2. Não:
3. Em primeiro lugar, comece comparando os elementos atuais de ambas as matrizes.
4. Em seguida, copie o elemento menor na matriz classificada.
5. Por último, mova o ponteiro do elemento que contém um elemento menor.
6. Sim:
7. Simplesmente copie o resto dos elementos do array não vazio

## Função Merge () explicada passo a passo

Considere o seguinte exemplo de um array não classificado, que iremos classificar com a ajuda do algoritmo Merge Sort.

A = (36,25,40,2,7,80,15)

**Etapa 1:** O algoritmo de classificação por mesclagem divide iterativamente uma matriz em metades iguais até atingir um valor atômico. No caso de haver um número ímpar de elementos em uma matriz, então uma das metades terá mais elementos do que a outra metade.

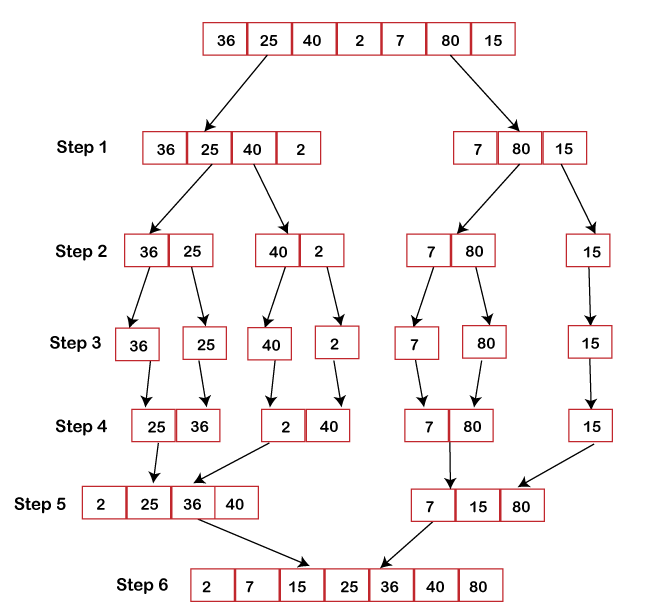
**Passo 2:** Depois de dividir uma matriz em duas submatrizes, notaremos que isso não prejudicou a ordem dos elementos como estavam na matriz original. Depois de agora, vamos dividir ainda mais essas duas matrizes em outras metades.

**Passo 3:** Novamente, vamos dividir essas matrizes até atingir um valor atômico, ou seja, um valor que não pode ser dividido posteriormente.

**Passo 4:** Em seguida, iremos mesclá-los de volta da mesma forma como foram divididos.

**Passo 5:** Para cada lista, iremos primeiro comparar o elemento e, em seguida, combiná-los para formar uma nova lista classificada.

**Passo 6:** Na próxima iteração, compararemos as listas de dois valores de dados e mesclá-los de volta em uma lista de valores de dados encontrados, todos colocados de maneira ordenada.



Portanto, a matriz é classificada.

## Análise de Merge Sort:

Seja T (n) o tempo total gasto pelo algoritmo Merge Sort.

* A classificação de duas metades levará no máximo 2T Classificação de mesclagem DAA.
* Quando mesclamos as listas ordenadas, obtemos uma comparação total de n-1 porque o último elemento restante precisará ser copiado para baixo na lista combinada e não haverá comparação.

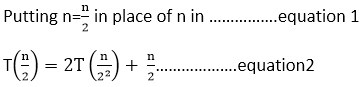
Assim, a fórmula relacional será

Classificação de mesclagem DAA

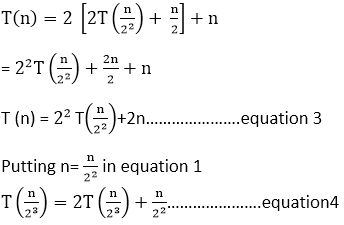
Mas ignoramos '-1' porque o elemento levará algum tempo para ser copiado nas listas de mesclagem.

Então, T (n) = 2T Classificação de mesclagem DAA+ n ... equação 1

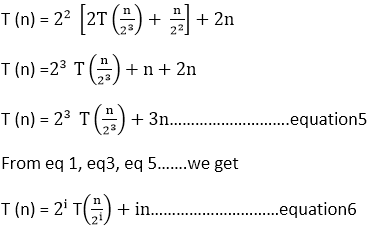
#### Nota: Condição de parada T (1) = 0 porque, finalmente, haverá apenas 1 elemento restante que precisa ser copiado e não haverá comparação.



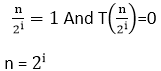
Coloque 2 equações em 1 equação



Colocando 4 equações em 3 equações



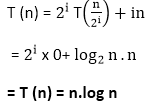
Da condição de parada:



Aplicar registro em ambos os lados:

log n = log 2 i  
logn = i log2  
Classificação de mesclagem DAA= i  
  
log 2 n = i

Da 6 equação



**Complexidade de melhor caso:** O algoritmo de classificação por mesclagem tem uma complexidade de tempo de melhor caso de **O (n \* log n)** para a matriz já classificada.

**Complexidade de caso médio:** A complexidade de tempo de caso médio para o algoritmo de classificação por mesclagem é **O (n \* log n)** , que acontece quando 2 ou mais elementos são misturados, ou seja, nem na ordem crescente nem na ordem decrescente.

**Complexidade de pior caso:** A complexidade de tempo de pior caso também é **O (n \* log n)** , que ocorre quando classificamos a ordem decrescente de uma matriz na ordem crescente.

**Complexidade do espaço:** A complexidade do espaço do merge sort é **O (n)** .

## Complexidade

|  |  |  |  |
| --- | --- | --- | --- |
| **Complexidade** | **Melhor caso** | **Caso Médio** | **Pior caso** |
| Complexidade de tempo | O (n log n) | O (n log n) | O (n log n) |
| Complexidade do Espaço |  |  | O(n) |

## Mesclar aplicativos de classificação

O conceito de classificação por mesclagem é aplicável nas seguintes áreas:

* Problema de contagem de inversão
* Classificação externa
* Aplicativos de comércio eletrônico