Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital

AULA 05

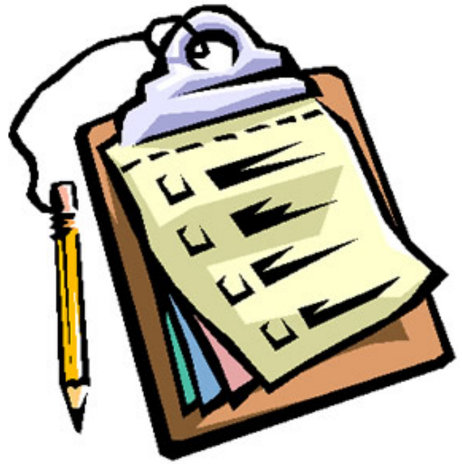
Estrutura de dados básico I (EDB1)

Prof. Msc. Janiheryson Felipe (Felipe)

Natal, RN
2023

OBJETIVO DA AULA

- Apresentar algoritmos de ordenação;
  - Conhecer o algoritmo Merge sort:
  - Conhecer o algoritmo Quick sort:



ALGORITMOS DE ORDENAÇÃO

MERGE SORT

O Merge Sort é um algoritmo de ordenação eficiente que utiliza a abordagem "divide and conquer" (dividir e conquistar) para ordenar uma lista de elementos. Ele foi desenvolvido por John von Neumann em 1945 e é considerado um dos algoritmos mais importantes e populares em ciência da computação.

MERGE SORT

O Merge Sort tem uma complexidade de tempo de O(n log n), o que significa que é muito eficiente para ordenar grandes conjuntos de dados. Ele também é um algoritmo estável, o que significa que ele preserva a ordem dos elementos iguais na lista de entrada na lista ordenada de saída.

MERGE SORT

No entanto, o Merge Sort tem uma complexidade de espaço de O(n), o que significa que ele pode exigir uma quantidade significativa de memória para ordenar grandes conjuntos de dados. Além disso, a implementação do Merge Sort pode ser complexa em comparação com outros algoritmos de ordenação, como o Bubble Sort ou o Insertion Sort.

MERGE SORT

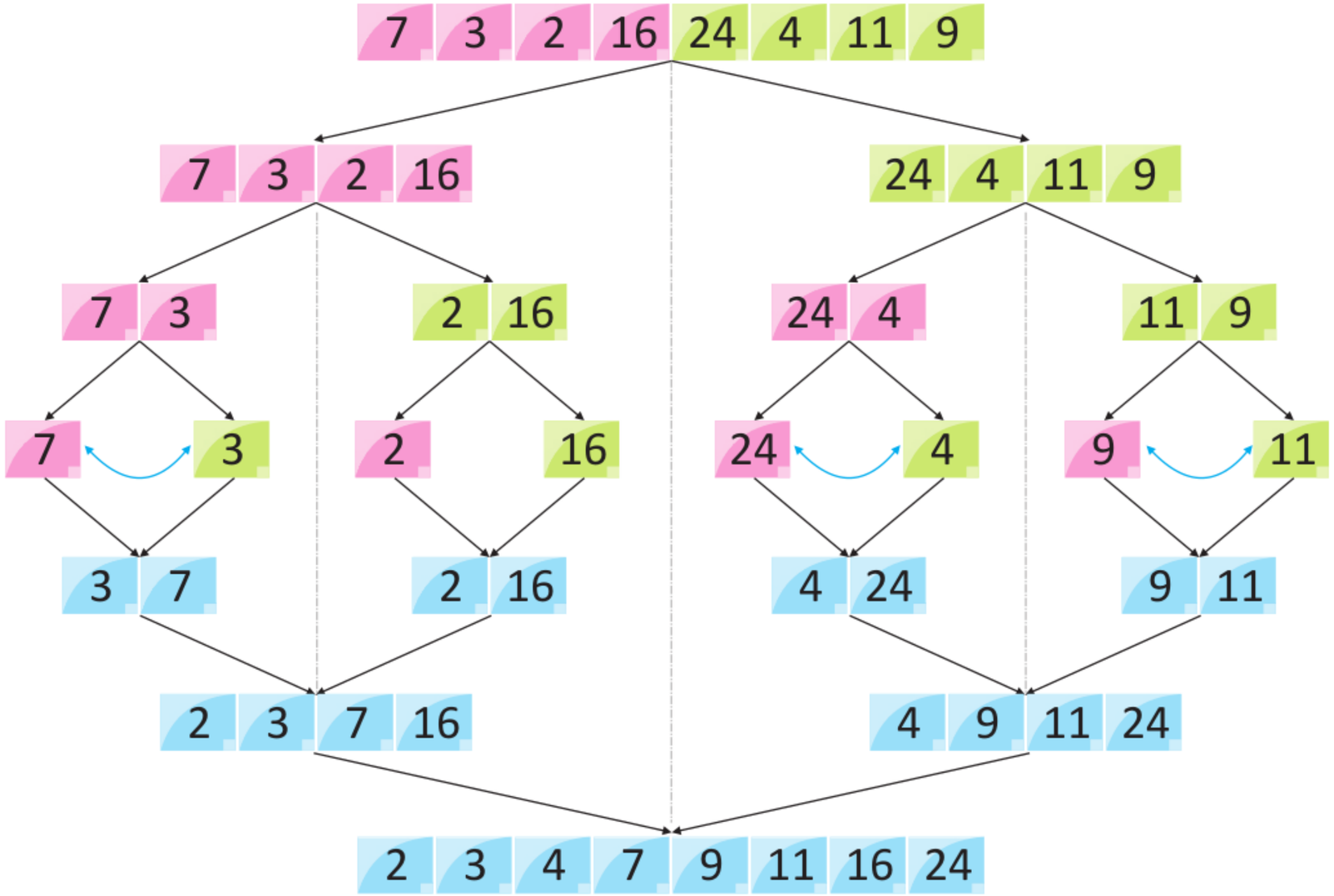
O algoritmo Merge Sort funciona da seguinte maneira:

1. Divida a lista de elementos em duas partes iguais.
2. Ordene recursivamente cada uma das partes, dividindo novamente cada parte pela metade até que cada parte tenha apenas um elemento (isso é a etapa de "dividir").

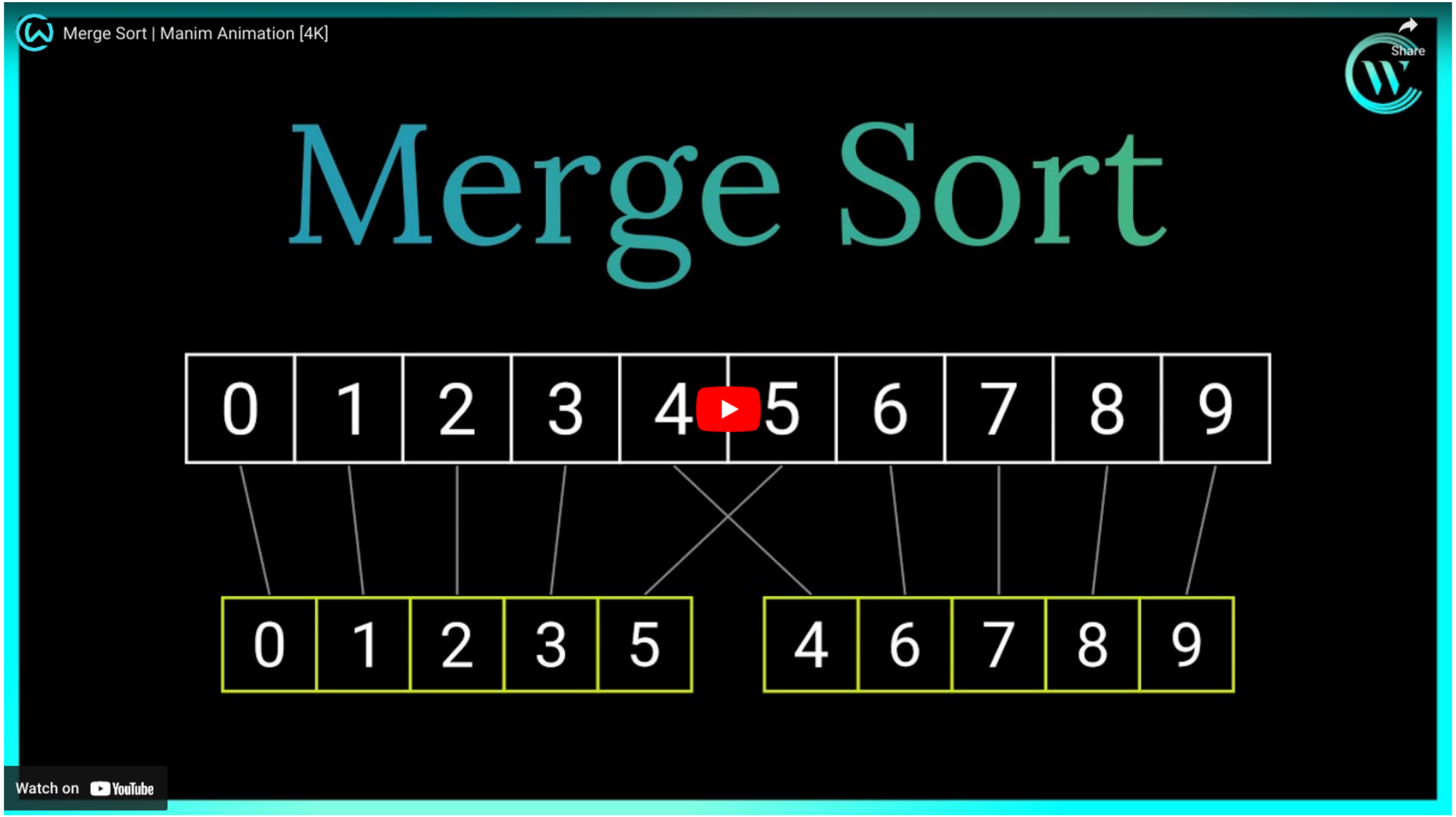
MERGE SORT

3. Junte as partes ordenadas usando a operação de "merge". O merge funciona comparando o menor elemento das duas partes e colocando-o na posição correta em uma nova lista. O processo é repetido até que todas as partes tenham sido mescladas em uma lista ordenada (isso é a etapa de "conquistar").

MERGE SORT


7 3 2 16 24 4 11 9
7 3 2 16
24 4 11 9
7 3
2 16
24 4
11 9
7
3
2
16
24
4
9
11
3 7
2 16
4 24
9 11
2 3 7 16
4 9 11 24
2 3 4 7 9 11 16 24

MERGE SORT


Merge Sort | Manim Animation [4K]
Share
Merge Sort
0 1 2 3 4 5 6 7 8 9
0 1 2 3 5
4 6 7 8 9
Watch on YouTube

QUICK SORT

O QuickSort é um algoritmo de ordenação muito utilizado na prática, por ser rápido e eficiente na maioria dos casos. Ele segue o paradigma "Dividir para Conquistar", que consiste em dividir o problema em subproblemas menores, resolvê-los recursivamente e, por fim, combinar as soluções para obter a solução do problema original.

QUICK SORT

O algoritmo funciona da seguinte maneira: seleciona-se um elemento do vetor (chamado de pivô), divide-se o vetor em duas partições: uma com elementos menores que o pivô e outra com elementos maiores que o pivô. Em seguida, o algoritmo é aplicado recursivamente em cada uma dessas partições até que a lista esteja ordenada.

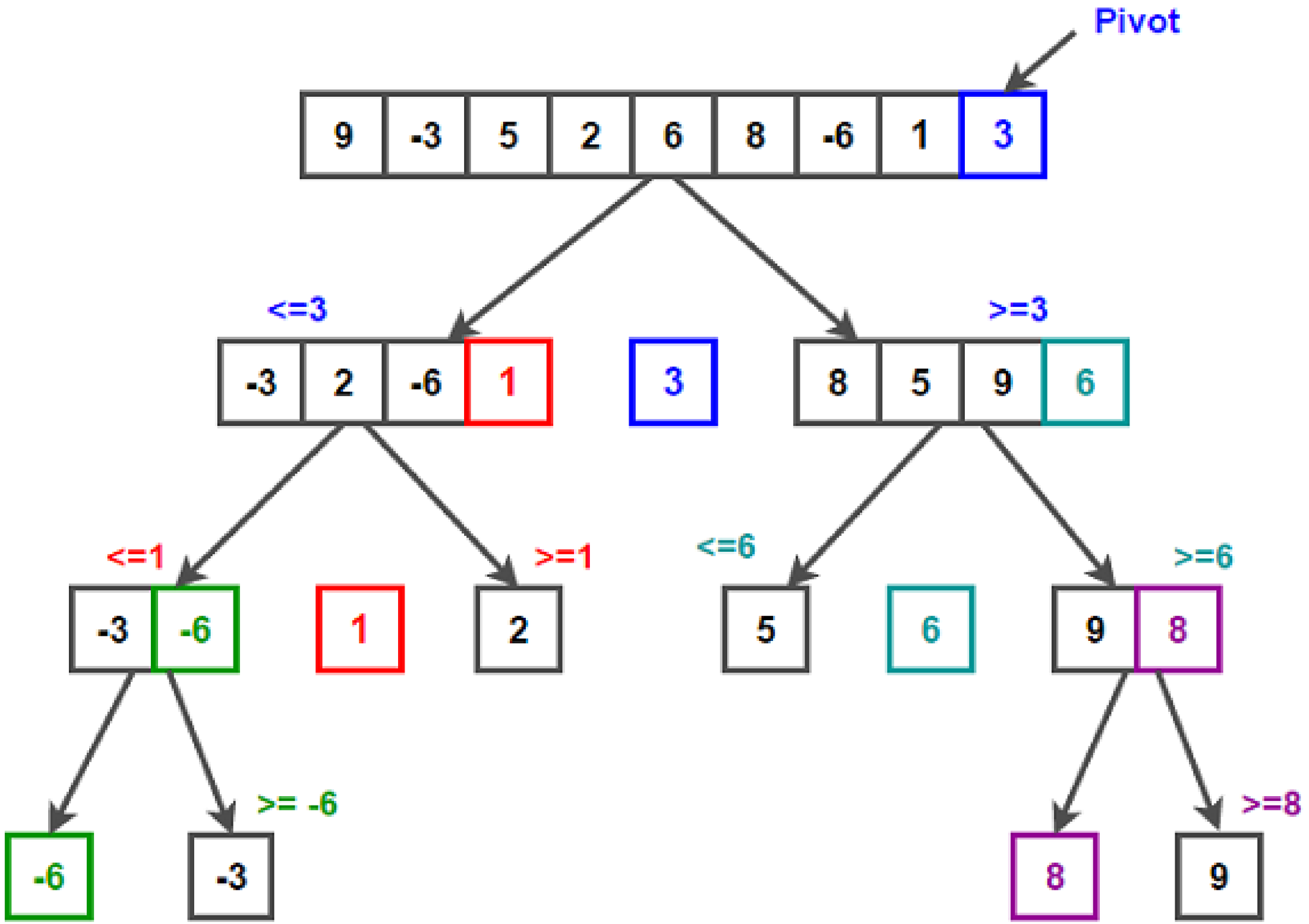
QUICK SORT

O pivô pode ser escolhido de diversas maneiras, mas a escolha mais comum é selecionar o primeiro elemento da lista. O algoritmo é in-place, ou seja, não utiliza memória adicional para a ordenação, mas pode ser instável, ou seja, elementos iguais podem não manter sua ordem relativa após a ordenação.

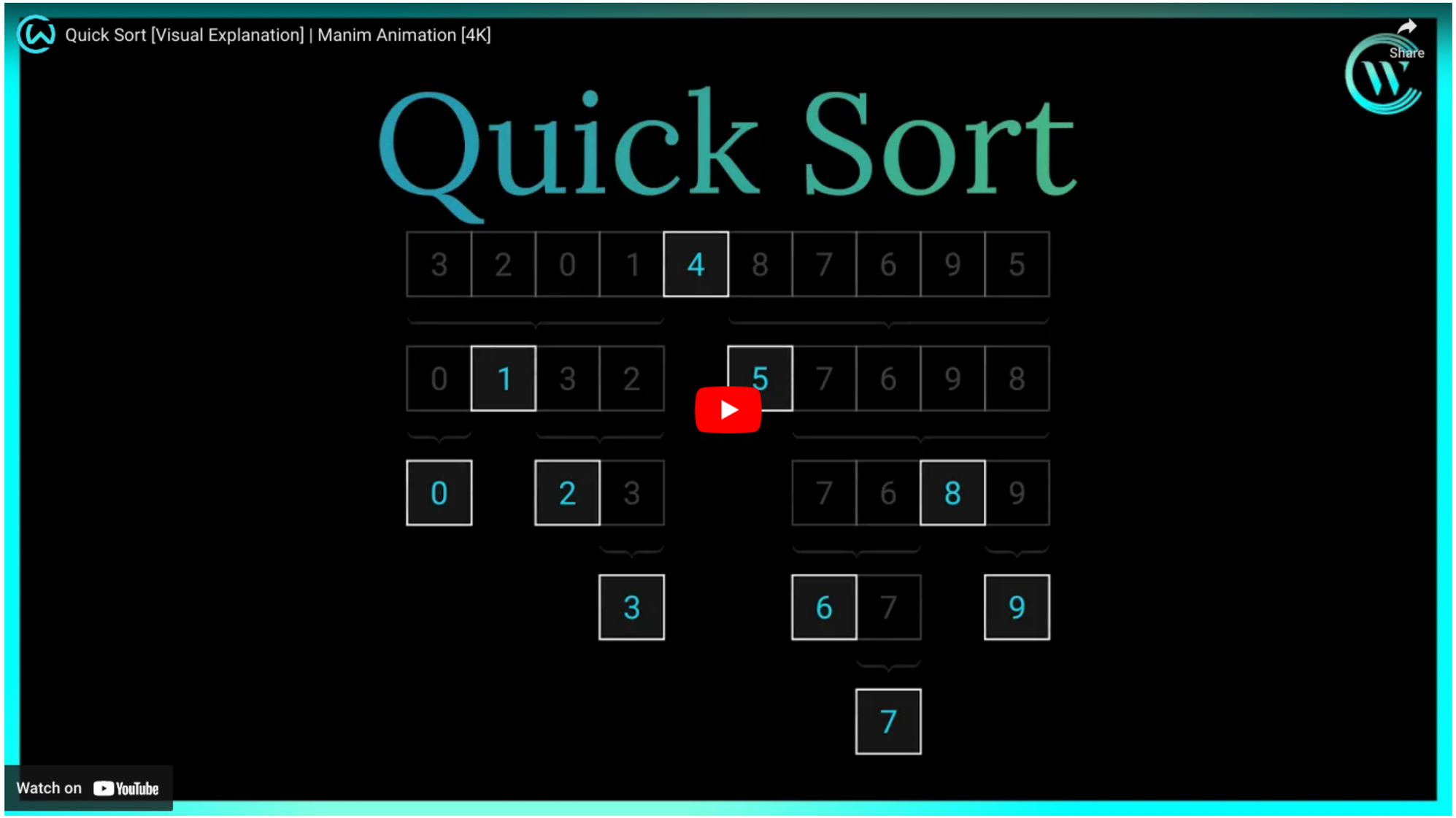
QUICK SORT

Apesar de possuir uma complexidade média de O(n log n), sua performance pode ser afetada se o pivô for mal escolhido, fazendo com que o algoritmo execute em $O(n^2)$ no pior caso. Porém, é possível adotar estratégias para minimizar essa possibilidade, como a escolha do pivô aleatoriamente ou através de algoritmos de seleção do pivô mais eficientes.

QUICK SORT



QUICK SORT


Quick Sort [Visual Explanation] | Manim Animation [4K]
Share
Quick Sort
3 2 0 1 4 8 7 6 9 5
0 1 3 2 5 7 6 9 8
0 2 3 7 6 8 9
3 6 7 9
7
Watch on YouTube

COMPARANDO OS ALGORITMOS

Algoritmo	Comparações			Movimentações			Espaço
	Melhor	Médio	Pior	Melhor	Médio	Pior	
Bubble	$O(n^2)$			$O(n^2)$			$O(1)$
Selection	$O(n^2)$			$O(n)$			$O(1)$
Insertion	$O(n)$	$O(n^2)$		$O(n)$	$O(n^2)$		$O(1)$
Merge	$O(n \log n)$			–			$O(n)$
Quick	$O(n \log n)$		$O(n^2)$	–			$O(n)$

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	Θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	Θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	Θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	Θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	Θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	Θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	Θ(n log(n))	O(n log(n))	O(n)

EXERCICIO 1 - BANCA IDECAN

Na computação existem algoritmos que utilizam diferentes técnicas de ordenação para organizar um conjunto de dados. Selecione o algoritmo de ordenação que usa um método eficiente com complexidade C(n) = O(n^2) no pior caso, e C(n) = O(n log n) no melhor e médio caso, com o seguinte funcionamento:
➢ Escolhe um elemento da lista chamado pivô. ➢ Reorganiza a lista de forma que os elementos menores que o pivô fiquem de um lado, e os maiores fiquem de outro. ➢ Recursivamente ordena a sub-lista abaixo e acima do pivô.
Assinale a alternativa correta.

A Selection Sort

B Insertion Sort

C Quick Sort

D Merge Sort

ANO: 2023 BANCA: IDECAN ÓRGÃO: SEFAZ-RR PROVA: IDECAN - 2023 - SEFAZ-RR - IMPLEMENTADOR DE SOFTWARE

EXERCICIO 2 - BANCA CPCOM

Não é um algoritmo clássico de ordenação:

A Quick Sort

B Prediction Sort.

C Merge Sort.

D Insertion Sort.

E Bubble Sort.

ANO: 2019 BANCA: CPCON ÓRGÃO: CÂMARA DE PAU DOS FERROS - RN PROVA: CPCON - 2019 - CÂMARA DE PAU DOS FERROS - RN - TÉCNICO EM INFORMÁTICA

EXERCICIO 3 - BANCA CESPE

julgue o item seguinte a respeito dos conceitos de algoritmo de ordenação.

O algoritmo merge sort ordena os elementos de um vetor percorrendo este diversas vezes e, a cada passagem, deslocando até o topo o maior elemento da sequência.

- Certo
- Errado

ANO: 2023 BANCA: CESPE / CEBRASPE ÓRGÃO: SEPLAN-RR PROVA: CESPE / CEBRASPE - 2023 - SEPLAN-RR - ANALISTA DE PLANEJAMENTO E ORÇAMENTO - ESPECIALIDADE: TECNOLOGIA DA INFORMAÇÃO

EXERCICIO 4 - BANCA CONSULPLAN

Algoritmos de ordenação são responsáveis por ordenar elementos de uma estrutura de dados de forma completa ou parcial. Sobre a complexidade dos algoritmos de ordenação, assinale, a seguir, o algoritmo de ordenação que, no pior caso, tem complexidade igual a O(n log n).

A *Quick sort.*

B *Merge sort.*

C *Bubble sort.*

D *Insertion sort.*

E *Selection sort.*

ANO: 2023 BANCA: INSTITUTO CONSULPLAN ÓRGÃO: MPE-BA PROVA: INSTITUTO CONSULPLAN - 2023 - MPE-BA - ANALISTA TÉCNICO – ANÁLISE DE SISTEMAS

EXERCICIO 5 - BANCA COMVEST UFAM

O método de ordenação caracterizado por ser o mais simples, cuja ideia é percorrer o vetor (ou array) diversas vezes, e a cada passagem fazer flutuar para o topo o maior elemento da sequência, é o método:

A) Bubble Sort.

B) Merge Sort.

C) Heap Sort.

D) Quick Sort.

E) Selection Sort.

ANO: 2019 BANCA: COMVEST UFAM ÓRGÃO: UFAM PROVA: COMVEST UFAM - 2019 - UFAM - TÉCNICO DE LABORATÓRIO - TECNOLOGIA DA INFORMAÇÃO

EXERCICIO 6 - BANCA CESPE

No que se refere aos conceitos de ordenamento, julgue o seguinte item.

A complexidade de tempo do algoritmo bubble sort é do tipo O(n × logn), logo, no caso desse algoritmo, o tempo de execução aumenta exponencialmente com o acréscimo do valor de n.

- Certo
- Errado

ANO: 2022 BANCA: CESPE / CEBRASPE ÓRGÃO: BNB PROVA: CESPE / CEBRASPE - 2022 - BNB - ANALISTA DE SISTEMAS - DESENVOLVIMENTO DE SISTEMAS

EXERCICIO 7 - BANCA FCC

Usando o algoritmo Bubble Sort um técnico deseja ordenar o conteúdo de um array utilizando o código JavaScript abaixo, presente no corpo de uma página HTML.

```
var arr = [220, 63, 25, 8, 5, 7, 295, 589];
for (var i = 0; i < arr.length; i++) {
    for (var j = 0; j < (arr.length - i - 1); j++)
        if (arr[j] > arr[j + 1]) {
            var temp = arr[j]
            I
            .....
            arr[j + 1] = temp
        }
    }
}
```

EXERCICIO 7 - BANCA FCC

```
var arr = [220, 63, 25, 8, 5, 7, 295, 589];
for (var i = 0; i < arr.length; i++) {
    for (var j = 0; j < (arr.length - i - 1); j++)
        if (arr[j] > arr[j + 1]) {
            var temp = arr[j]

             I
            .....

            arr[j + 1] = temp
        }
    }
}
```

A) arr[j] = arr[j + i]

B) arr[i] = arr[j + 1]

C) arr[j] = arr[j - 1]

D) arr[j + 1] = arr[j + 1]

E) arr[j] = arr[j + 1]

ANO: 2022 BANCA: FCC ÓRGÃO: TRT - 5ª REGIÃO (BA) PROVA: FCC - 2022 - TRT - 5ª REGIÃO (BA) - TÉCNICO JUDICIÁRIO - TECNOLOGIA DA INFORMAÇÃO

EXERCICIO 8 - BANCA IBFC

Um tipo de algoritmo muito usado na resolução de problemas computacionais são os algoritmos de ordenação. Assinale a única alternativa que esteja tecnicamente incorreta quanto a especificar exatamente algoritmos de ordenação simples

- A Quick Sort
- B Smart Sort
- C Insertion So
- D Bubble Sort

ANO: 2022 BANCA: CESPE / CEBRASPE ÓRGÃO: BNB PROVA: CESPE / CEBRASPE - 2022 - BNB - ANALISTA DE SISTEMAS - DESENVOLVIMENTO DE SISTEMAS

DÚVIDAS???