Ordenações - Parte 1

Prof.: Leonardo Tórtoro Pereira

leonardop@usp.br

Ordenação

- ➔ Algoritmos de ordenação são essenciais no dia a dia da computação
- ➔ Ordenar produtos/filmes/jogos/músicas por maior avaliação, menor preço, maior preço, nome, etc.
- ➔ Ordenar resultados de busca para o mais similar ao procurado
- ➔ Ordenar inventários em jogos...

Ordenação

- ➔ Devido à importância, algoritmos mais eficientes para ordenar são pesquisados há décadas
- ➔ Hoje temos diversos algoritmos, com diversas complexidades diferentes
- ➔ Alguns são melhores que outros, e outros podem ser melhores em situações específicas!

Ordenação

- Por toda essa diversidade, são um caso de estudo MUITO interessante para comparar a eficiência de algoritmos!
- Normalmente dizemos que estamos ordenando as “chaves” de um registro
  - Ou seja, apenas um item específico de uma estrutura possivelmente mais complexa
  - Exemplo: ordenar sua pokédex pelo número da dex :)

Ordenação

- Qualquer tipo de dado pode ser usado como chave.
- Só é preciso que seja ordenável
  - E como ordenar pode ser à critério do programador
  - Exemplo: números, letras, palavras...
  - Ou: ordenar por peso e, daqueles com mesmo peso, ordenar por valor
  - Ou: ordenar pela soma do valor das letras de uma palavra em ASC II...

Ordenação

- Dois grupos de algoritmos de ordenação
  - Interna
    - Todo o arquivo cabe na memória principal
  - Externa
    - Arquivo precisa ser armazenado em disco/fita
    - Se for em disco do tipo HD, acesso é sequencial ou em grandes blocos
    - Atualmente esse problema foi mitigado com SSDs

Ordenação

- A ordenação mais conhecida é baseada em comparação
  - Compara duas chaves e ordena
- Porém, existe ordenação por distribuição
  - Resumidamente:
    - Dividir algo em grupos (que normalmente tem uma ordem entre si)
    - E ordenar os itens dentro de cada grupo, isoladamente

Ordenação Interna

Ordenação Interna

- Principal aspecto é o menor tempo de execução
- Porém, gasto de memória auxiliar pode ser importante para alguns algoritmos que necessitam disso.
  - Preferência por executar tudo em um vetor. Raramente, uso de tabela ou pilha
  - Listas encadeadas ou cópias extras para ordenação costumam não ter muita importância

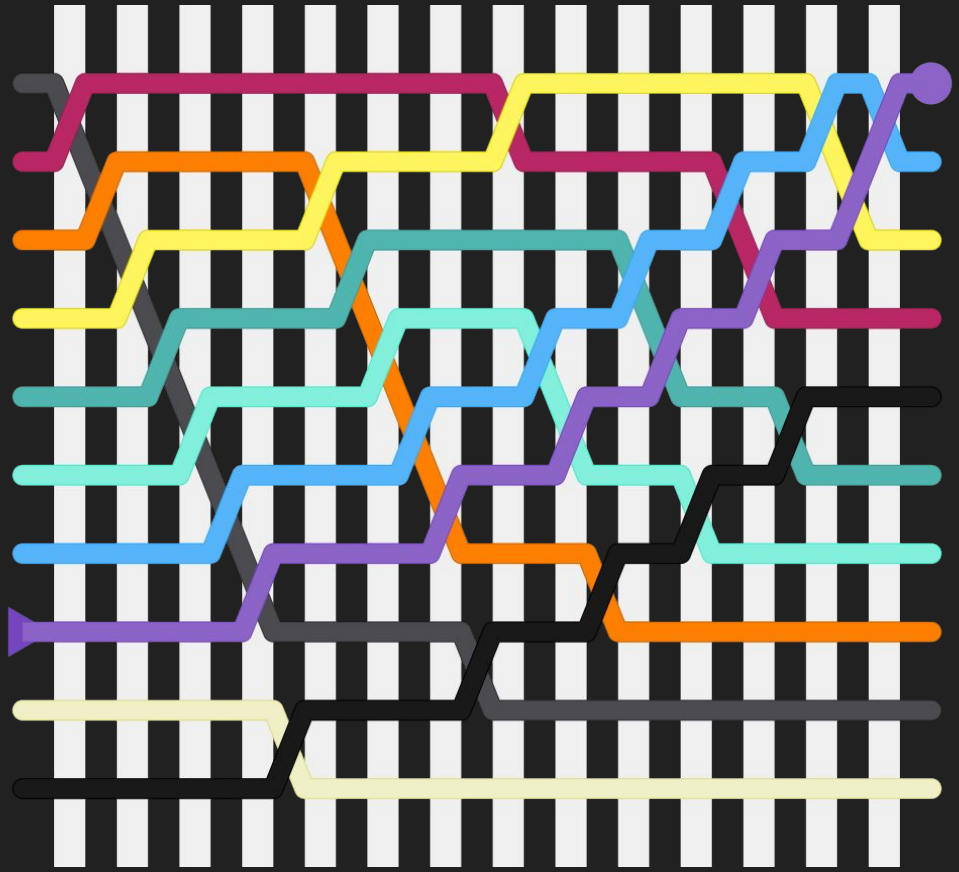
Ordenação Interna

- Podem ser classificados em dois tipos de métodos:
  - Simples
    - Bons para pequenos arquivos: $O(n^2)$
    - Fáceis de entender. Podem até mesmo ser mais eficientes em arquivos pequenos
  - Eficientes
    - Bons para arquivos maiores: O(nlogn)

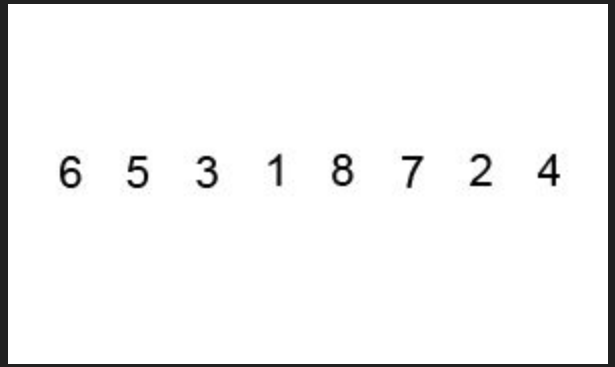
Bubble Sort

Bubble Sort

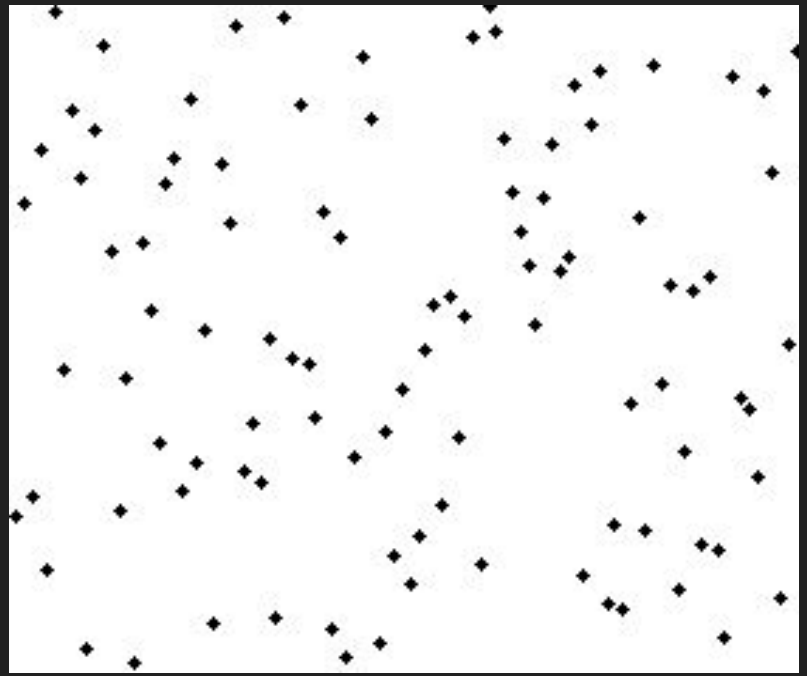
- Percorrer o vetor diversas vezes
  - A cada passagem, jogar para o topo o maior elemento da sequência



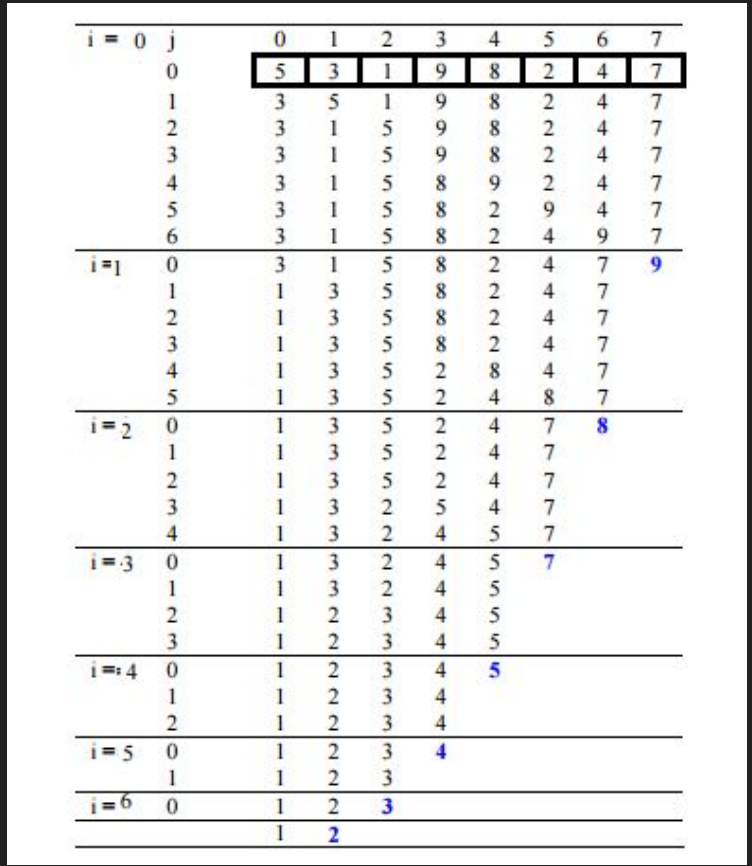
Fonte:
https://en.wikipedia.org/wiki/Bubble_sort#/media/File:Bubblesort-edited-color.svg



Fonte:
https://upload.wikimedia.org/wikipedia/commons/c/c8/Bubble-sort-example-300px.gif



Fonte:
https://upload.wikimedia.org/wikipedia/commons/3/37/Bubble_sort_animation.gif



		0	1	2	3	4	5	6	7
i = 0	j								
	0	5	3	1	9	8	2	4	7
	1	3	5	1	9	8	2	4	7
	2	3	1	5	9	8	2	4	7
	3	3	1	5	9	8	2	4	7
	4	3	1	5	8	9	2	4	7
	5	3	1	5	8	2	9	4	7
	6	3	1	5	8	2	4	9	7
i=1	0	3	1	5	8	2	4	7	**9**
	1	1	3	5	8	2	4	7	
	2	1	3	5	8	2	4	7	
	3	1	3	5	8	2	4	7	
	4	1	3	5	2	8	4	7	
	5	1	3	5	2	4	8	7	
i=2	0	1	3	5	2	4	7	**8**	
	1	1	3	5	2	4	7		
	2	1	3	5	2	4	7		
	3	1	3	2	5	4	7		
	4	1	3	2	4	5	7		
i=3	0	1	3	2	4	5	**7**		
	1	1	3	2	4	5			
	2	1	2	3	4	5			
	3	1	2	3	4	5			
i=4	0	1	2	3	4	**5**			
	1	1	2	3	4				
	2	1	2	3	4				
i=5	0	1	2	3	**4**				
	1	1	2	3					
i=6	0	1	2	**3**					
		1	**2**						

Fonte:
https://www.geeksforgeeks.org/bubble-sort/

Bubble sort

classe	Algoritmo de ordenação
estrutura de dados	Array, Listas ligadas
complexidade pior caso	$O(n^2)$
complexidade caso médio	$O(n^2)$
complexidade melhor caso	$O(n)$
complexidade de espaços pior caso	$O(1)$ auxiliar

Fonte: https://pt.wikipedia.org/wiki/Bubble_sort

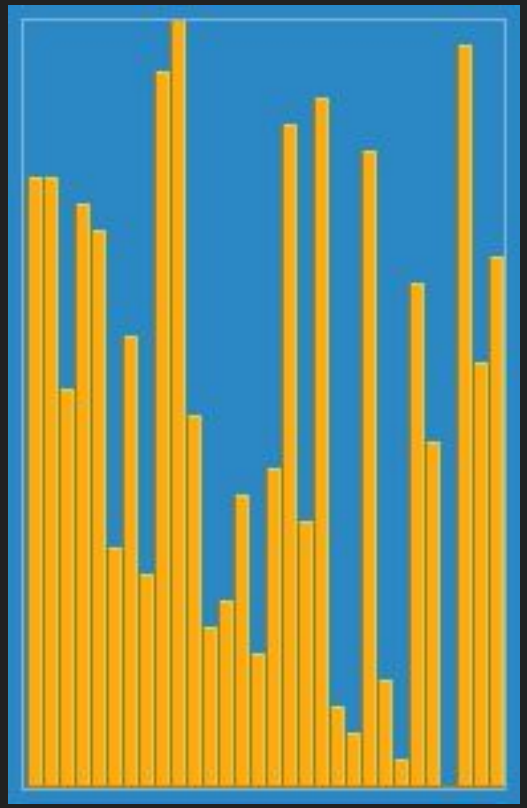
Insertion Sort

Insertion Sort

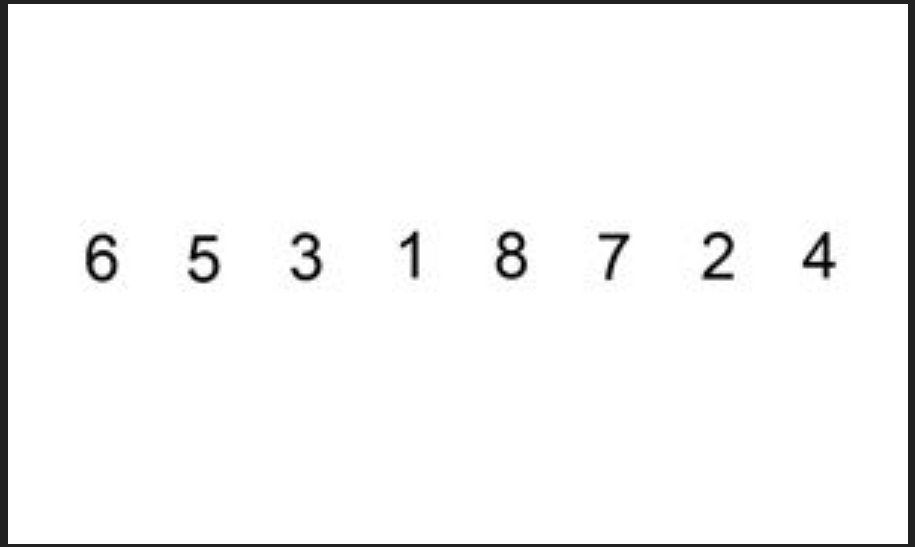
- ➔ Insere um item por vez numa estrutura
- ➔ É como se você ordenasse um baralho retirando uma carta por vez do monte e colocando ela no lugar certo de uma lista ordenada

Insertion Sort

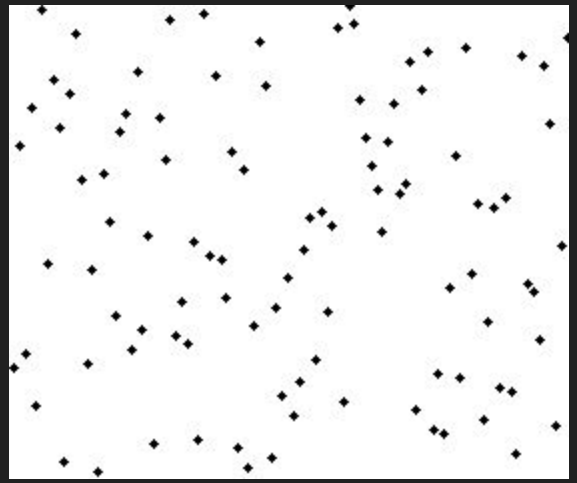
- Iterar do índice 1 ao n do array
- Comparar o elemento atual com o anterior
- Se for menor que o predecessor, comparar com os elementos anteriores
  - Mover os elementos maiores uma posição acima para dar espaço ao elemento que será trocado



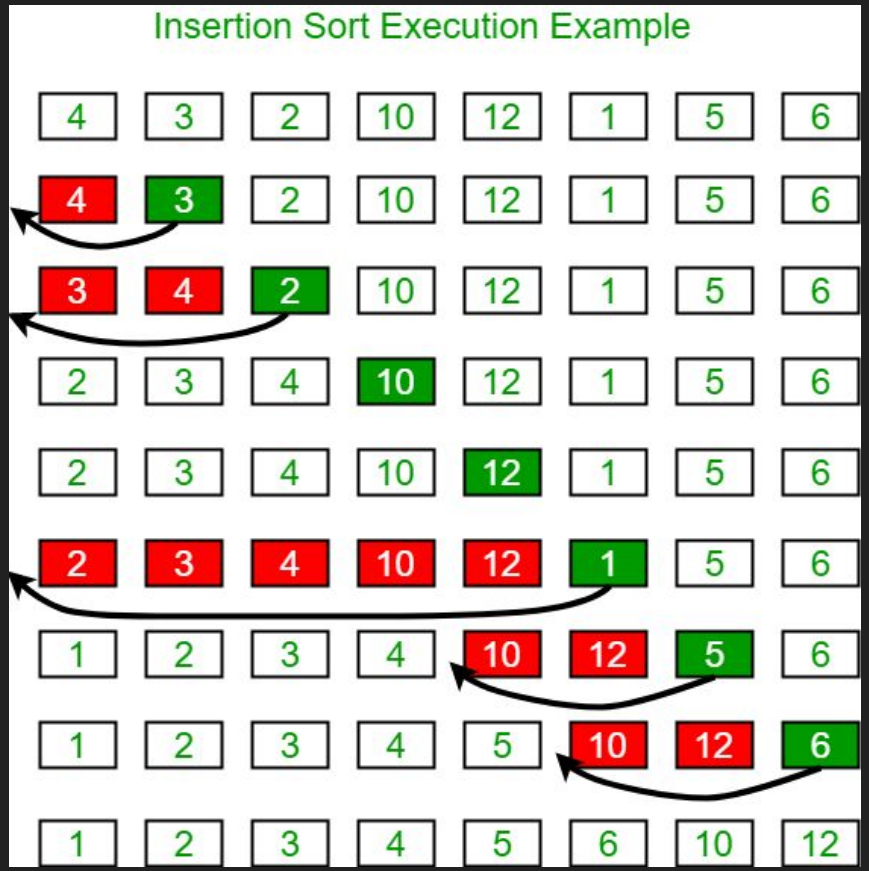
Fonte:
https://upload.wikimedia.org/wikipedia/commons/4/42/Insertion_sort.gif



Fonte:
https://upload.wikimedia.org/wikipedia/commons/0/0f/Insertion-sort-example-300px.gif



Fonte:
https://upload.wikimedia.org/wikipedia/commons/2/25/Insertion_sort_animation.gif



Fonte: https://www.geeksforgeeks.org/insertion-sort/

Insertion sort

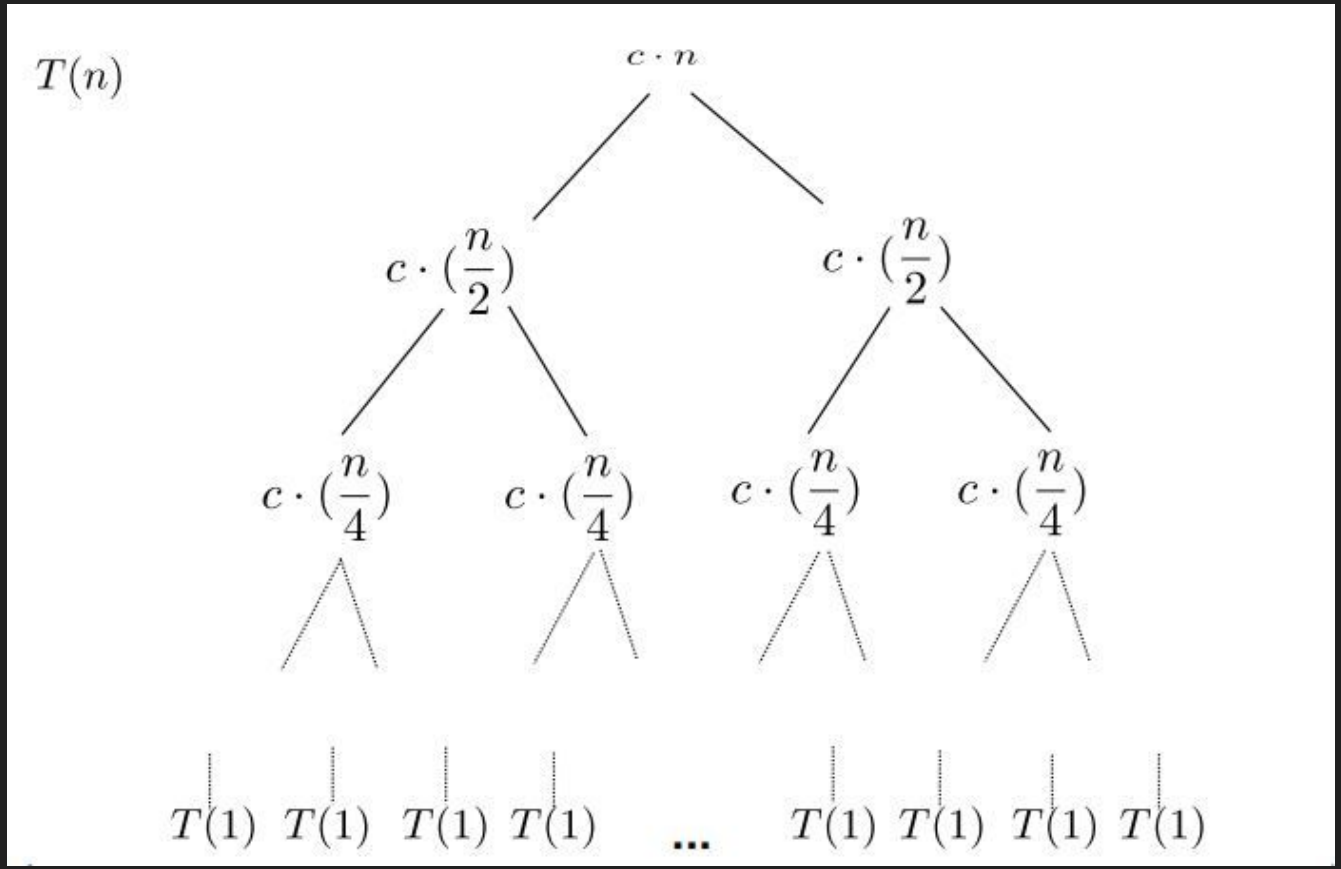
classe	Algoritmo de ordenação
estrutura de dados	Array, Listas ligadas
complexidade pior caso	$O(n^2)$
complexidade caso médio	$O(n^2)$
complexidade melhor caso	$O(n)$
complexidade de espaços pior caso	$O(n)$ total, $O(1)$ auxiliar

Fonte: https://pt.wikipedia.org/wiki/Insertion_sort

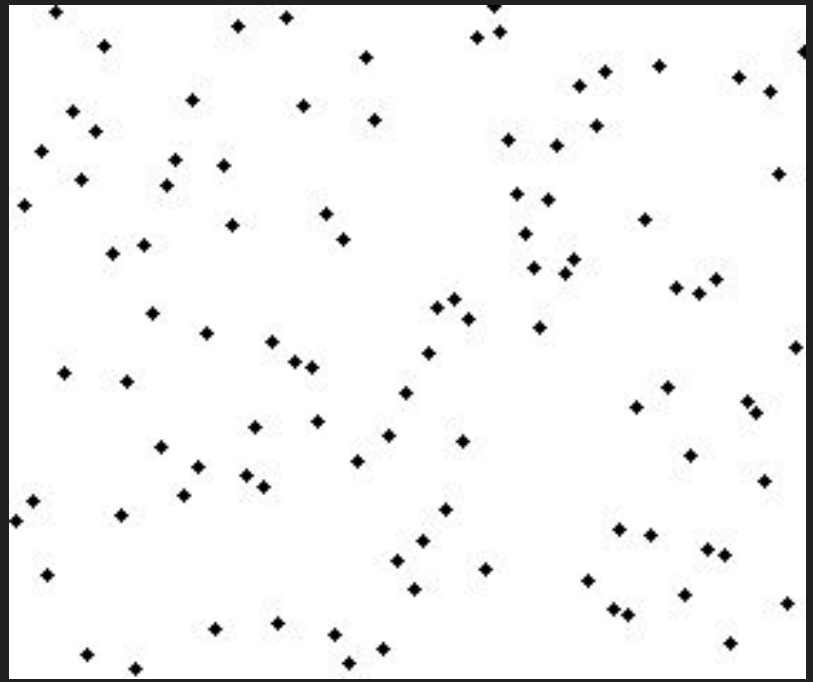
Merge Sort

Merge Sort

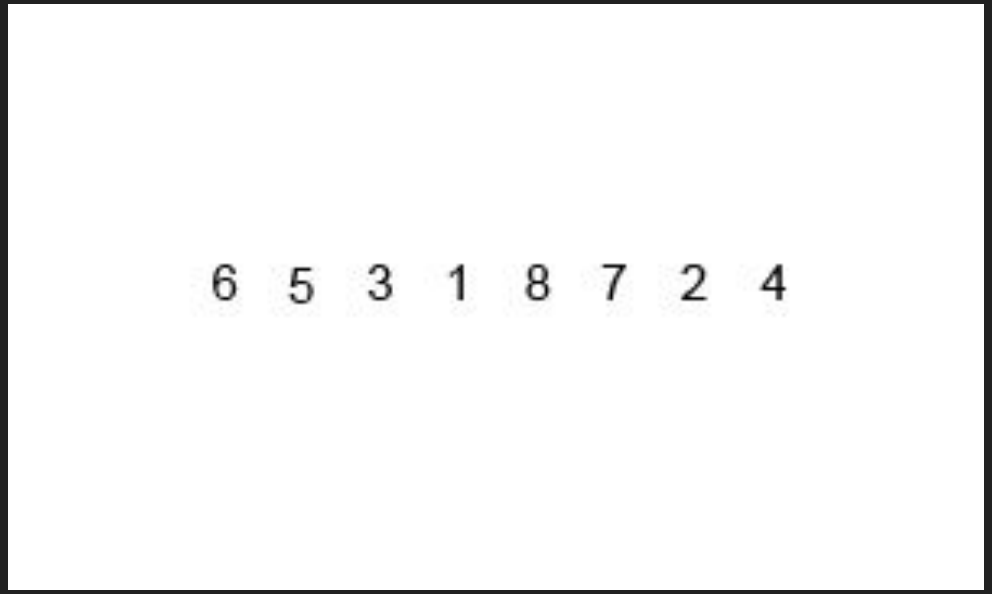
- Dividir o problema em subproblemas
  - Resolver eles com recursividade
- Após todos serem resolvidos, unir as soluções
- Consumo de memória e tempo de execução um pouco alta



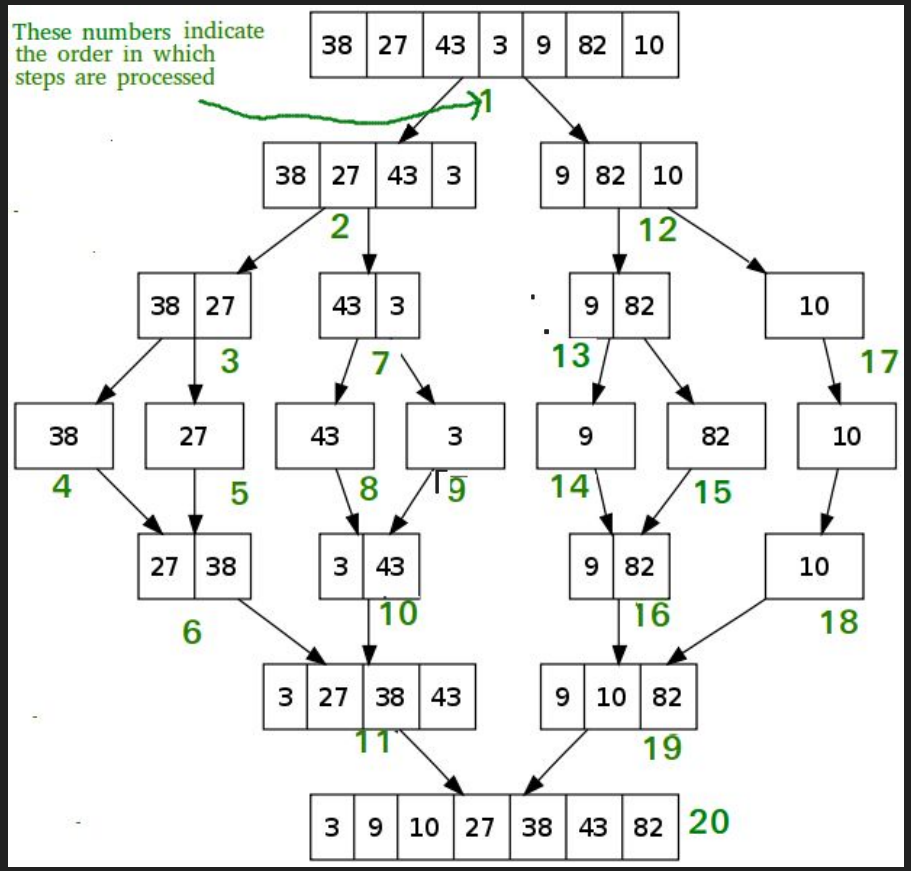
Fonte: https://pt.wikipedia.org/wiki/Merge_sort



Fonte:
https://upload.wikimedia.org/wikipedia/commons/c/c5/Merge_sort_animation2.gif

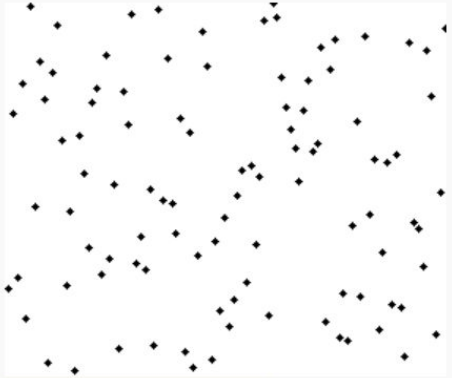


Fonte:
https://upload.wikimedia.org/wikipedia/commons/c/cc/Merge-sort-example-300px.gif



Fonte: https://www.geeksforgeeks.org/merge-sort/

Merge sort



classe	Algoritmo de ordenação
estrutura de dados	Array, Listas ligadas
complexidade pior caso	$\Theta(n \log n)$
complexidade caso médio	$\Theta(n \log n)$
complexidade melhor caso	$\Theta(n \log n)$ típico, $\Theta(n)$ variante natural
complexidade de espaços pior caso	$\Theta(n \log n)$

Fonte: https://pt.wikipedia.org/wiki/Merge_sort

Referências

- ZIVIANI, N. Projeto de Algoritmos. 2º edição, Thomson, 2004.
- https://pt.wikipedia.org/wiki/Bubble_sort
- https://en.wikipedia.org/wiki/Bubble_sort
- https://www.geeksforgeeks.org/bubble-sort/
- https://pt.wikipedia.org/wiki/Insertion_sort
- https://en.wikipedia.org/wiki/Insertion_sort
- https://www.geeksforgeeks.org/insertion-sort/
- https://pt.wikipedia.org/wiki/Merge_sort
- https://en.wikipedia.org/wiki/Merge_sort
- https://www.geeksforgeeks.org/merge-sort/