

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**

# Lógica de programação e algoritmos

## Busca e ordenação

Aula 2

Código da aula: [DADOS]ANO1C3B3S23A2

Lógica de  
programação e  
algoritmos

## Mapa da Unidade 1 Componente 4

Prática de busca e  
ordenação

semana

25

Algoritmos de  
contagem e  
acumulação

semana

27

semana

21

Recursividade

semana

23

**Você está aqui!**

Busca e ordenação

semana

28

Pilhas e filas

Lógica de  
programação e  
algoritmos

## Mapa da Unidade 1 Componente 4

**Você está aqui!**

Busca e ordenação

**Aula 2**

Código da aula: [DADOS]ANO1C3B3S23A2

**23**



## Objetivos da aula

- Apresentar o algoritmo de busca binária;
- Incentivar a pesquisa dos alunos por outros algoritmos de busca.



## Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou à internet.



## Duração da Aula

50 minutos.



## Competência Técnica

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados.



## Competência Socioemocional

- Demonstrar resiliência para lidar com pressões e enfrentar novos desafios, bem como com frustrações quando um projeto de Ciência de Dados falhar.

Construindo  
o **conceito**

## Revisão: busca linear

- ▶ Na aula passada, aprendemos que uma estratégia de **busca de elementos** em uma lista é a busca linear.
- ▶ A **busca linear** percorre cada elemento de uma lista **sequencialmente** até encontrar o elemento desejado.



Construindo  
o **conceito**

**Considere a lista a seguir**

**23, 47, 35, 19, 56, 42, 13, 8, 39**

Elaborado especialmente para este curso.

## Pior caso da busca linear

- ▶ Na lista do slide anterior, se quisermos buscar o número 39, qual será a complexidade?
- ▶ Para encontrar o número 39 em nossa lista, a busca linear faz 9 comparações – ele é o último número. Isso ilustra o “pior caso” para a busca linear em uma lista de 9 elementos, necessitando de 9 operações.



Construindo  
o **conceito**

**Será que há uma forma mais fácil de  
fazer isso?**

**23, 47, 35, 19, 56, 42, 13, 8, 39**

Elaborado especialmente para este curso.

Construindo  
o **conceito**

## Busca binária

- ▶ A **busca binária** começa ordenando a lista. Em seguida, compara-se o elemento do meio da lista ordenada com o valor-alvo.
- ▶ Se não for igual, a busca continua na metade da lista, em que o valor poderia estar, reduzindo o problema pela metade a cada passo. Isso continua até encontrar o valor ou esgotar a lista.

Construindo  
o **conceito**

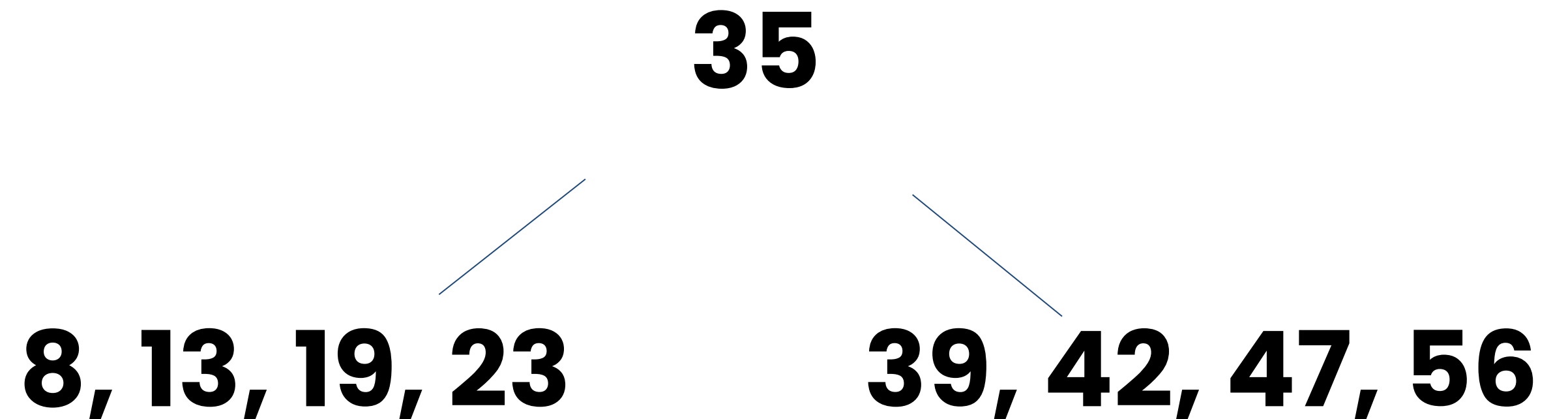
## **Passo 1: ordenar a lista**

**23, 47, 35, 19, 56, 42, 13, 8, 39**

Elaborado especialmente para este curso.

Construindo  
o **conceito**

**Passo 2: comparar o 39 com o meio  
da lista, separando a lista no meio**



Elaborado especialmente para este curso.

Construindo  
o **conceito**

**Passo 3: buscar na nova metade  
da lista**

**39, 42, 47, 56**

Elaborado especialmente para este curso.



Construindo  
o **conceito**

## Passo 3

**39**

**42**

**47, 56**

Elaborado especialmente para este curso.



Construindo  
o **conceito**

**Passo 4: encontramos o termo que  
buscávamos**

**39**

Elaborado especialmente para este curso.

Construindo  
o **conceito**

## Complexidade da busca binária

- ▶ Na busca binária, a cada comparação, cortamos o espaço que falta buscar pela metade.
- ▶ Dessa forma, a complexidade desse algoritmo é  $O(\log n)$ .

Construindo  
o **conceito**

## Conectando conceitos: indexação

- ▶ Assim como a busca binária usa uma “lista ordenada” para encontrar elementos rapidamente, os bancos de dados usam “indexação” para acelerar a busca de registros.
- ▶ Na verdade, a busca binária é uma das estratégias para indexar um banco de dados, aumentando o desempenho.



Colocando  
em **prática**

## Outros algoritmos de busca

Vamos praticar?

- ▶ Reúnam-se em grupos de 4 pessoas.
- ▶ Busquem por algum outro algoritmo que seja usado para indexar banco de dados e escrevam um texto explicando como ele funciona.
- ▶ Se sobrar tempo, apresentem para a sala o algoritmo escolhido pelo grupo.



**Fim da aula**



**Grupos de 4 pessoas**



**Enviar no AVA**



© Getty Images

O que nós  
**aprendemos**  
**hoje?**

## Então ficamos assim...

- 1** Relembramos que a busca linear percorre sequencialmente uma lista para encontrar um elemento.
- 2** Vimos como a busca binária, com sua abordagem de divisão e conquista em uma lista ordenada, oferece uma complexidade de tempo  $O(\log n)$ .
- 3** Conectamos nosso aprendizado com o módulo de banco de dados, entendendo que a indexação nada mais é do que uma estratégia de busca.



# Saiba mais

Está pronto para dominar a busca binária em Python? Acesse este artigo e aprenda como implementar essa técnica de maneira eficaz:

MATOS, B. R. S. Busca binária: aprenda a implementar em Python. *Alura*, 24 maio 2022. Disponível em: <https://www.alura.com.br/artigos/busca-binaria-aprenda-implementar-python/>. Acesso em: 25 jun. 2024.



# Referências da aula

Identidade visual: imagens © Getty Images

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. *Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python*. Porto Alegre: Bookman, 2022.

# Educação Profissional Paulista

Técnico em  
**Ciência de  
Dados**