

Binary Search

Allana Paganini, Bianca Maccarini,
Jaqueline Camargo e João Danielewicz

The background is white and features several decorative elements: a solid tan rectangle in the top-left corner; a tan irregular polygon in the top-right; a tan irregular polygon in the bottom-right; a tan irregular polygon in the bottom-center; and sets of concentric blue circles in the top-right and bottom-left corners.

O que é o Binary Search?



Binary Search, também conhecido como busca binária, é um algoritmo de busca utilizado para encontrar um determinado elemento em uma lista ordenada de forma eficiente. Ele divide repetidamente a lista ao meio, comparando o elemento buscado com o elemento do meio da lista. Com base nessa comparação, o algoritmo descarta metade da lista, continuando a busca na metade restante. Esse processo é repetido até que o elemento seja encontrado ou até que não haja mais elementos para comparar.



Exemplificação

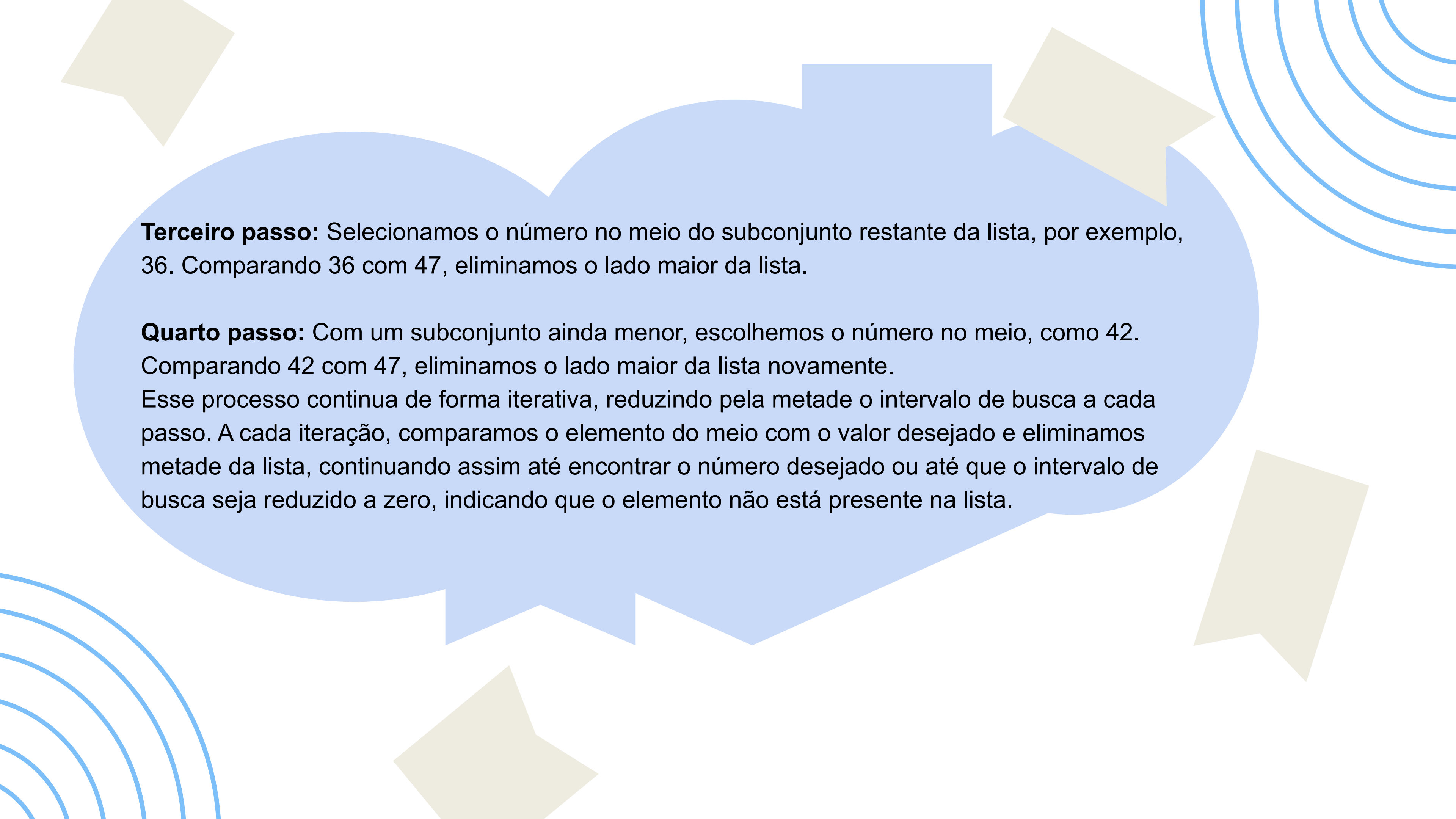
Primeiro passo: Selecionar o número que está no meio da lista. 14 será comparado com 47, como 47 é maior, o lado menor da lista será eliminado.

Segundo passo: Selecionar o número que está no meio da lista não eliminada. $45 < 47$, o lado menor será novamente eliminado.

Isso irá se repetir até que o número desejado seja encontrado.

Search for 47

0	4	7	10	14	23	45	47	53
---	---	---	----	----	----	----	----	----



Terceiro passo: Seleccionamos o número no meio do subconjunto restante da lista, por exemplo, 36. Comparando 36 com 47, eliminamos o lado maior da lista.

Quarto passo: Com um subconjunto ainda menor, escolhemos o número no meio, como 42. Comparando 42 com 47, eliminamos o lado maior da lista novamente.

Esse processo continua de forma iterativa, reduzindo pela metade o intervalo de busca a cada passo. A cada iteração, comparamos o elemento do meio com o valor desejado e eliminamos metade da lista, continuando assim até encontrar o número desejado ou até que o intervalo de busca seja reduzido a zero, indicando que o elemento não está presente na lista.

De uma forma mais leiga; A busca binária é como encontrar um número em um livro, começando no meio e indo para a metade certa a cada vez, até encontrar o número desejado, economizando tempo ao pular páginas.



The background features a light cream color with abstract geometric elements. In the top-left, there is a tan triangle pointing right. In the top-right, a tan rectangle is partially visible. On the far right, there are several concentric blue circles. In the bottom-left, more concentric blue circles are visible. In the bottom-right, a tan triangle points left. A central, light blue, irregular polygonal shape contains the text.

Aplicação em C#

Melhor caso

O número a ser buscado está no meio do Array.

PASSO 1

Lista a ser usada: | 3 | 7 | 9 | 15 | 17 | 24 | 36 | 38 | 45 |

O número desejado está na posição 5 da lista.

“Segundo” melhor caso

O número a ser buscado está no meio de alguma das subdivisões da lista.

PASSO 1

Lista a ser usada: | 3 | 7 | 9 | 15 | 17 | 24 | 36 | 38 | 45 |

PASSO 2

Lista a ser usada: | 24 | 36 | 38 | 45 |

O número desejado está na posição 7 da lista.

Pior caso

O número a ser buscado está em uma das extremidades do Array.

```
PASSO 1
Lista a ser usada: | 3 | 7 | 9 | 15 | 17 | 24 | 36 | 38 | 45 |

PASSO 2
Lista a ser usada: | 3 | 7 | 9 | 15 |

PASSO 3
Lista a ser usada: | 3 |

O número desejado está na posição 1 da lista.
```

```
PASSO 1
Lista a ser usada: | 3 | 7 | 9 | 15 | 17 | 24 | 36 | 38 | 45 |

PASSO 2
Lista a ser usada: | 24 | 36 | 38 | 45 |

PASSO 3
Lista a ser usada: | 38 | 45 |

PASSO 4
Lista a ser usada: | 45 |

O número desejado está na posição 9 da lista.
```

Fim!

