SÃO
PAULO
TECH
SCHOOL

SÃO PAULO TECH SCHOOL

ED

Estrutura de Dados e Armazenamento

Pesquisa Binária ou

Busca Binária

Profa. Célia Taniwaki

Pesquisa Sequencial vs Pesquisa Binária

- Quando é preciso verificar se um valor está em um vetor, normalmente executamos uma Pesquisa Sequencial.
- Na **Pesquisa Sequencial**, percorre-se o vetor, desde o início, seguindo na sequência do vetor, do índice zero até o último elemento do vetor, até encontrar o valor procurado. Se o valor não for encontrado, percorrerá o vetor inteiro.
- A Pesquisa Sequencial funciona tanto se os dados do vetor estiverem desordenados como se estiverem ordenados.

- Existe outro método de busca ou pesquisa, chamada de **Busca ou Pesquisa Binária**
- Esse método funciona apenas quando os dados estão ordenados.

Pesquisa Binária

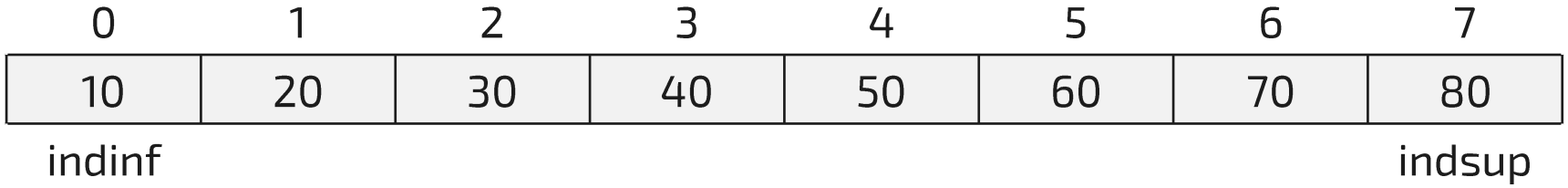
- Na Pesquisa Binária, ao invés de começarmos a procurar no início do vetor, verificamos se o elemento que está sendo procurado está no meio do vetor.
- Se estiver, então já encontramos.
- Se não estiver: verificamos: Valor procurado > elemento do meio ?
  - Sim, então procuramos na metade à direita do elemento do meio
  - Não, então procuramos na metade à esquerda do elemento do meio
- A pesquisa continua até encontrar o elemento procurado, ou até o algoritmo detectar que o elemento não está no vetor.

• Pesquisa Binária - Algoritmo

```
inteiro pesqBin (inteiro[] vetor, inteiro x)   // x é o valor procurado
início
   inteiro indinf, indsup, meio;
   indinf = 0;                  /* índice inferior */
   indsup = vetor.length - 1; /* índice superior */
   enquanto (indinf <= indsup)
   início
      meio = (indinf + indsup) / 2;       /* calcula o índice do meio */
      se (vetor[meio] == x)
         retorna meio;              // retorna o índice do elemento encontrado
      senão se (x < vetor[meio])  // x < elemento do meio ?
                indsup = meio – 1; // sim, então indsup passa a ser meio-1
            senão
                indinf = meio + 1; // não, então indinf passa a ser meio+1
    fim
    retorna -1; /* quando indinf > indsup, o elemento não está no vetor */
fim
```

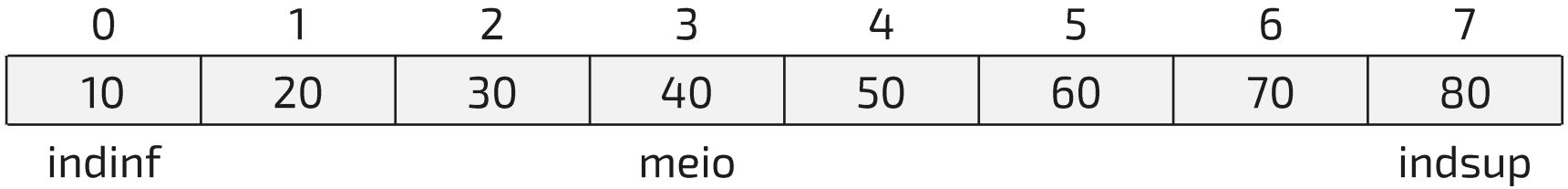
Pesquisa Binária – Exemplo (simulação)

Sejam os dados (simulação do algoritmo anterior):



⇒ vamos supor que o elemento procurado x = 50

⇒ meio = (indinf + indsup)/2 = 3



⇒ x = vetor[meio]? Não

⇒ x < vetor[meio]? Não

⇒ Então x > vetor[meio] ⇒ indinf = meio +1 = 4

Pesquisa Binária – Exemplo (simulação)

0	1	2	3	4	5	6	7
10	20	30	40	50	60	70	80
				indinf			indsup

⇒ A busca agora terá foco na parte do vetor, dos índices 4 a 7

⇒ meio = (indinf + indsup)/2 = 5

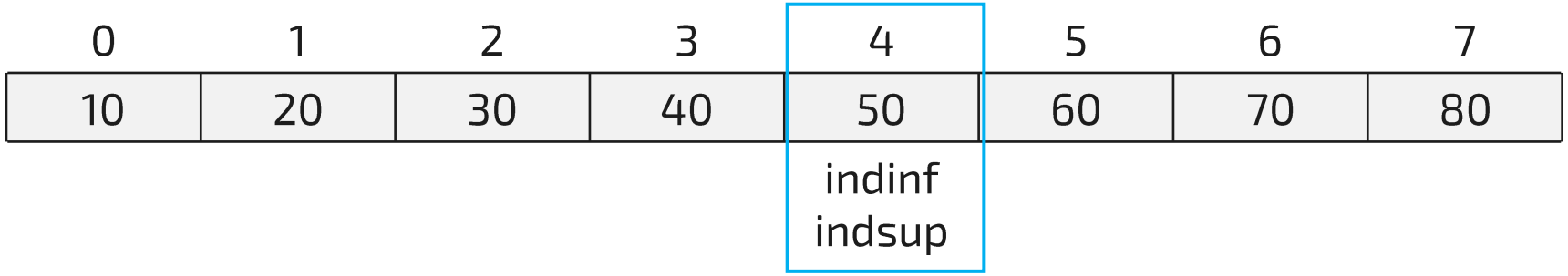
0	1	2	3	4	5	6	7
10	20	30	40	50	60	70	80
				indinf	meio		indsup

⇒ x = vetor[meio]? Não

⇒ x < vetor[meio]? Sim ⇒ indsup = meio-1 = 4

0	1	2	3	4	5	6	7
10	20	30	40	50	60	70	80
				indinf indsup			

Pesquisa Binária – Exemplo (simulação)



⇒ A busca agora terá foco na parte do vetor, dos índices 4 a 4

⇒ meio = (indinf + indsup)/2 = 4

⇒ x = vetor[meio]? Sim ⇒ Encontramos!!!

Dessa forma, em 3 iterações, encontramos o elemento procurado.

Para um vetor de 8 posições, em 3 iterações podemos encontrar o valor ou concluir que o valor não está no vetor.

Se fosse a busca sequencial, precisaríamos de 8 iterações para concluir que o valor não está no vetor.

Notação O-grande ou Big-O

- Em computação, utiliza-se a notação O-grande ou Big-O para representar a complexidade de tempo de execução de um algoritmo
- Leva-se em conta a quantidade de operações que o algoritmo executa em função da quantidade de dados que ele manipula

- A pesquisa sequencial tem complexidade O(n), sendo n o tamanho do vetor.
- Num vetor de 8 posições, são necessários 8 iterações para concluir que o elemento não está no vetor

- A pesquisa binária tem complexidade O($\log_2 n$)
- Quando n=8, $\log_2 n$ = 3. Pois $2^3 = 8$. Vimos que em 3 iterações, encontramos o elemento ou concluímos que ele não está no vetor.

Pesquisa binária vs Pesquisa sequencial

- Se o vetor tiver tamanho 1024, sabemos que $2^{10} = 1024$.
- Isso significa que na pesquisa sequencial, é preciso 1024 iterações para descobrir que o valor procurado não está no vetor.
- Na pesquisa binária, é preciso 10 iterações para descobrir que o valor procurado não está no vetor.

- Se o vetor tiver tamanho 4096, sabemos que $2^{12} = 4096$.
- Isso significa que na pesquisa sequencial, é preciso 4096 iterações para descobrir que o valor procurado não está no vetor.
- Na pesquisa binária, é preciso 12 iterações para descobrir que o valor procurado não está no vetor.

Agradeço
a sua atenção!

Célia Taniwaki

celia.taniwaki@sptech.school

SÃO PAULO TECH SCHOOL