

Estrutura de Dados II

Prof. Me. Pietro M. de Oliveira

Busca Binária

Checa o elemento do meio

- Se encontrar,
- Dependendo do valor do elemento
 - Metade superior, ou metade inferior
- Repete-se o processo

Otimização: evita a busca em posições onde já se sabe que o alvo não estará, sempre reduzindo o conjunto pela metade.

Necessita de arranjo ordenado

“Dividir para conquistar”

Melhor aplicado em vetores estáticos

- Versão dinâmica: inviável (solução: usar árvore)

Versão recursiva

- A concepção de busca binária sugere recursividade

Versão iterativa

- Exige algumas variáveis de controle
- Aparentemente, maior complexidade no código

BuscaBináriaRec(*A, chave, início, final*)

1. Se *início* \leq *final* então
2. *meio* $\leftarrow (final + início) / 2$
3. Se *A[meio]* = *chave* então
4. retorne *meio*
5. Se não, se *chave* < *A[meio]* então
6. **BuscaBináriaRec(*A, chave, início, meio-1*)**
7. Se não, se *chave* > *A[meio]*
8. **BuscaBináriaRec(*A, chave, meio+1, final*)**
9. Se não
10. retorne “*elemento ausente*”

Exemplo: buscar o item de valor = 67

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

- ***início*** $\leftarrow 0$
- ***final*** $\leftarrow 9$
- ***início*** \leq ***final***? Então calcule o ***meio***
 - ***meio*** $\leftarrow (9+0)/2 = 4$

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

- ***$A[\text{meio}] = \text{chave?}$***

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

- **$A[\textit{meio}] = \textit{chave}$?**
 - Não, **$\textit{chave} > A[\textit{meio}]$** !
 - **$\textit{final} \leftarrow \textit{final}$**
 - **$\textit{início} \leftarrow \textit{meio} + 1$**
 - Chamada recursiva ($\textit{início} = 5$; $\textit{final} = 9$)

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

- ***início*** $\leftarrow 5$
- ***final*** $\leftarrow 9$
- ***início*** \leq ***final***? Então calcule o ***meio***
 - ***meio*** $\leftarrow (9+5)/2 = 7$

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

- ***$A[\textit{meio}] = \textit{chave}?$***

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

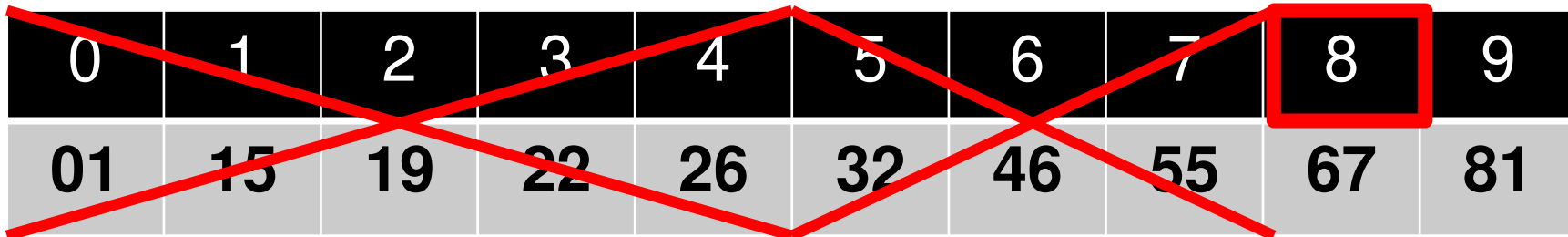
Exemplo: buscar o item de valor = 67

- **$A[\text{meio}] = \text{chave}$?**
 - Não, **$\text{chave} > A[\text{meio}]$** !
 - **$\text{final} \leftarrow \text{meio}$**
 - **$\text{início} \leftarrow \text{meio} + 1$**
 - Chamada recursiva ($\text{início} = 8$; $\text{final} = 9$)

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

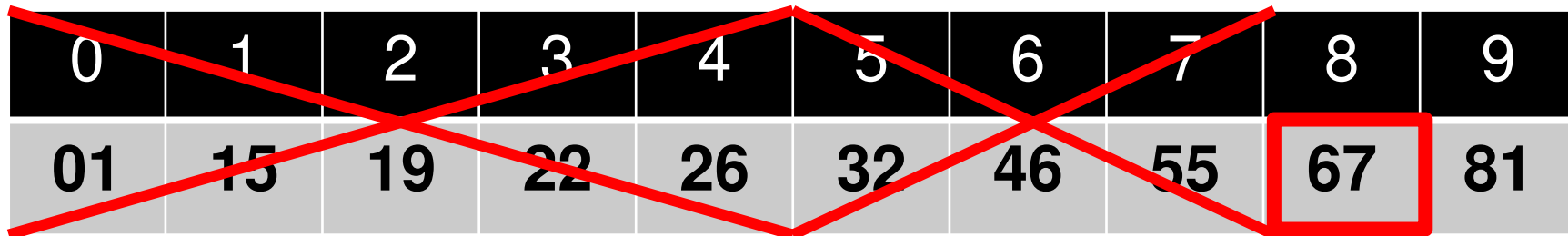
- ***início*** $\leftarrow 8$
- ***final*** $\leftarrow 9$
- ***início*** \leq ***final***? Então calcule o ***meio***
 - ***meio*** $\leftarrow (9+8)/2 = 8$



0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

- ***$A[\text{meio}] = \text{chave?}$***

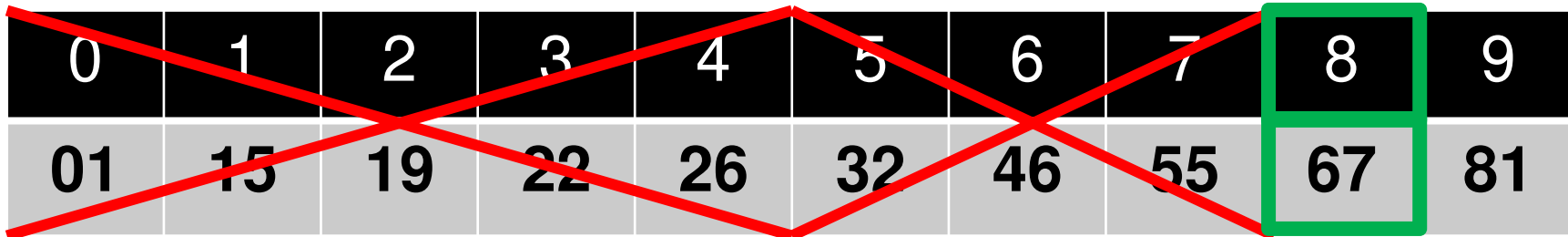


0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 67

- **$A[\textit{meio}] = \textit{chave}$?**
 - Sim!
 - retorne *meio*

Elemento encontrado com 3 iterações



0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

- ***início*** $\leftarrow 0$
- ***final*** $\leftarrow 9$
- ***início*** \leq ***final***? Então calcule o ***meio***
 - ***meio*** $\leftarrow (9+0)/2 = 4$

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

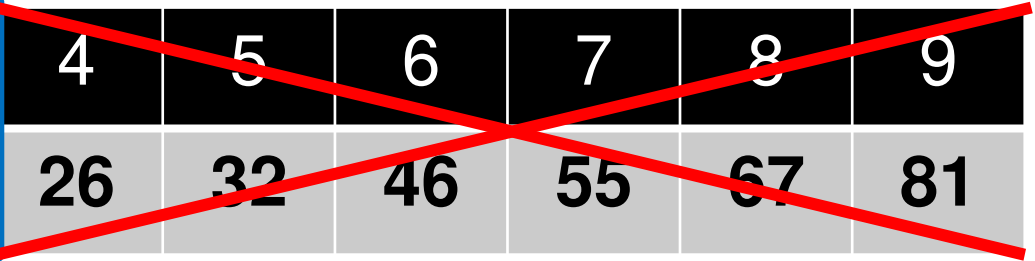
- ***$A[\text{meio}] = \text{chave?}$***

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

- **$A[\textit{meio}] = \textit{chave}$?**
 - Não, **$\textit{chave} < A[\textit{meio}]$!**
 - **$\textit{final} \leftarrow \textit{meio}-1$**
 - **$\textit{início} \leftarrow \textit{início}$**
 - Chamada recursiva ($\textit{início} = 3$; $\textit{final} = 3$)

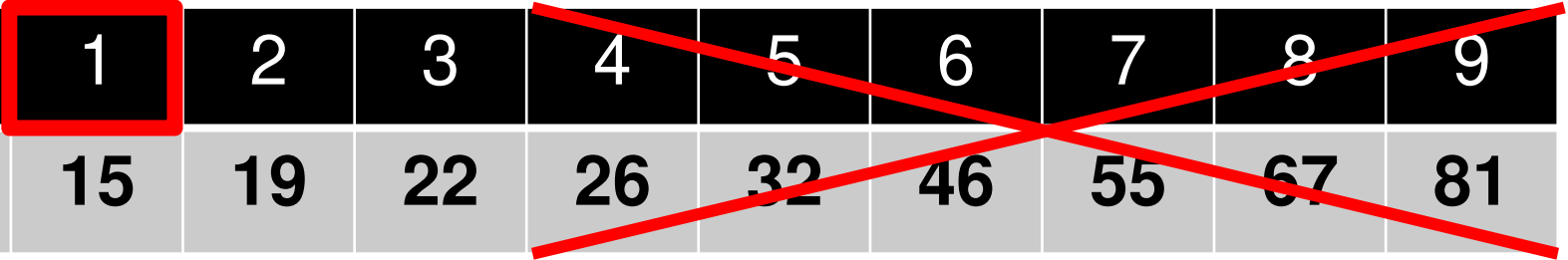
0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81



Exemplo: buscar o item de valor = 20

- ***início*** $\leftarrow 0$
- ***final*** $\leftarrow 3$
- ***início*** \leq ***final***? Então calcule o ***meio***
 - ***meio*** $\leftarrow (3+0)/2 = 1$

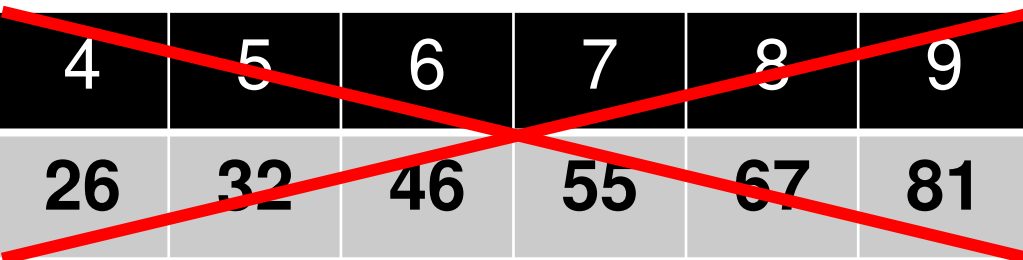
0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81



Exemplo: buscar o item de valor = 20

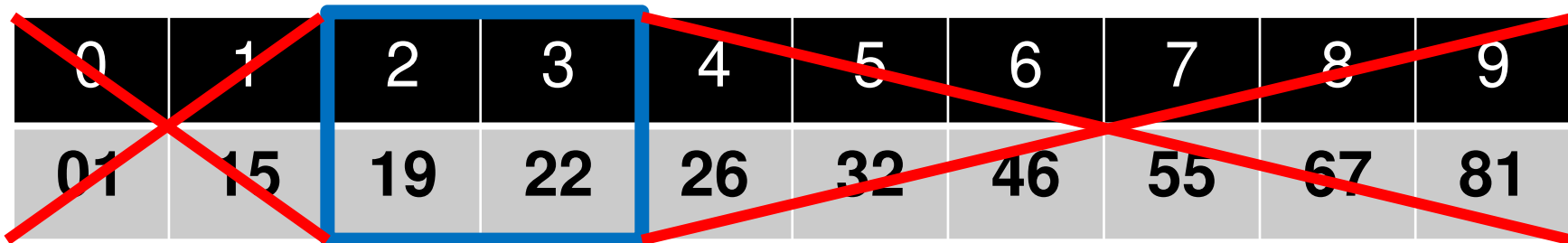
- ***$A[\text{meio}] = \text{chave?}$***

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81



Exemplo: buscar o item de valor = 20

- **$A[\text{meio}] = \text{chave}$?**
 - Não, **$\text{chave} > A[\text{meio}]$** !
 - **$\text{final} \leftarrow \text{meio}$**
 - **$\text{início} \leftarrow \text{meio} + 1$**
 - Chamada recursiva ($\text{início} = 2$; $\text{final} = 3$)



0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

- ***início*** $\leftarrow 2$
- ***final*** $\leftarrow 3$
- ***início*** \leq ***final***? Então calcule o ***meio***
 - ***meio*** $\leftarrow (3+2)/2 = 2$

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

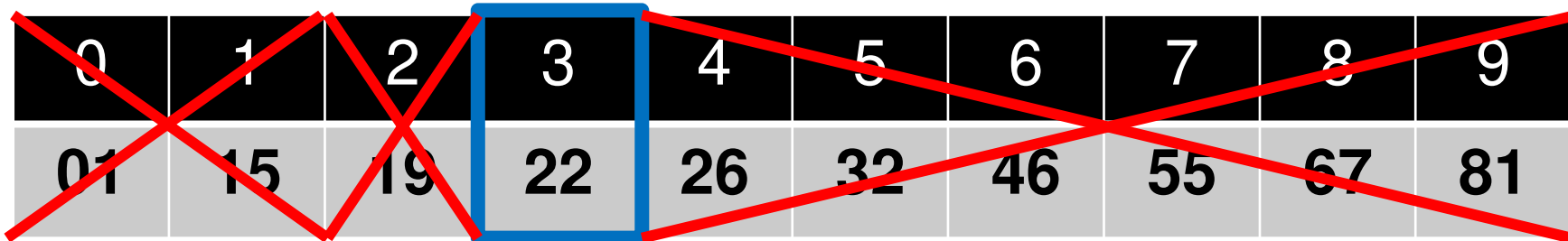
Exemplo: buscar o item de valor = 20

- ***$A[\text{meio}] = \text{chave?}$***

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

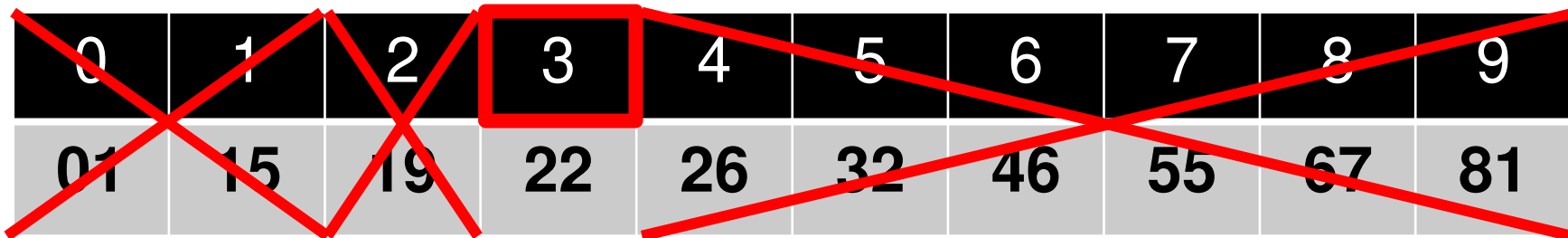
- **$A[\text{meio}] = \text{chave}$?**
 - Não, **$\text{chave} > A[\text{meio}]$** !
 - **$\text{final} \leftarrow \text{meio}$**
 - **$\text{início} \leftarrow \text{meio} + 1$**
 - Chamada recursiva ($\text{início} = 3$; $\text{final} = 3$)



0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

- ***início*** $\leftarrow 3$
- ***final*** $\leftarrow 3$
- ***início*** \leq ***final***? Então calcule o ***meio***
 - ***meio*** $\leftarrow (3+3)/2 = 3$



0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

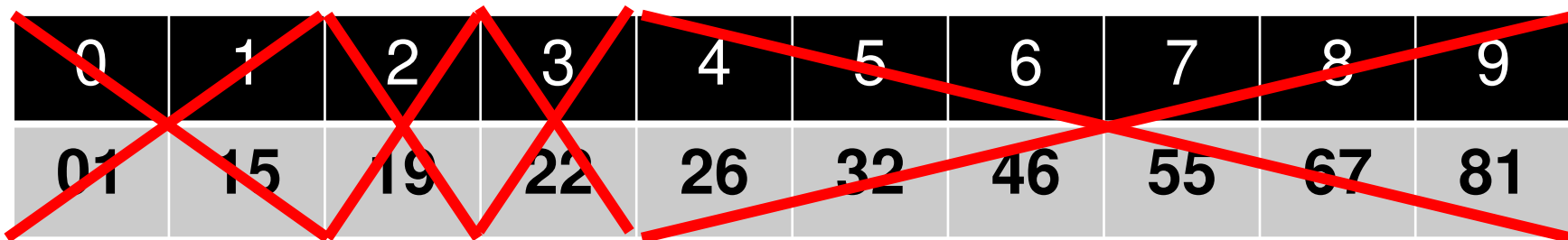
Exemplo: buscar o item de valor = 20

- ***$A[\text{meio}] = \text{chave?}$***

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

- **$A[\text{meio}] = \text{chave}$?**
 - Não, **$\text{chave} < A[\text{meio}]$** !
 - **$\text{final} \leftarrow \text{meio}-1$**
 - **$\text{início} \leftarrow \text{início}$**
 - Chamada recursiva ($\text{início} = 3$; $\text{final} = 2$)

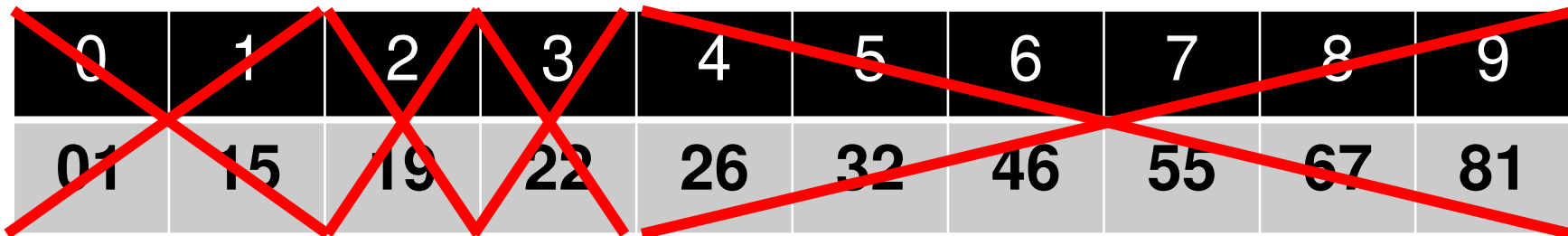


0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Exemplo: buscar o item de valor = 20

- ***início*** $\leftarrow 3$
- ***final*** $\leftarrow 2$
- ***início*** \leq ***final***?
 - Não!

Elemento não encontrado, com 4 iterações



0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Busca por Interpolação

Tentativa de calcular o **elemento intermediário** de maneira diferenciada

- Utiliza uma fórmula para isso
- A fórmula leva em conta os índices, a chave e os elementos das “pontas”

Supõe que os **elementos** estarão **distribuídos uniformemente** no arranjo

Busca muito parecida com a busca binária

$$meio = inicio + (fim - inicio) * \frac{(chave - vec[inicio])}{(vec[fim] - vec[inicio])}$$

Exemplo: buscar o elemento alvo = 22

- $meio \leftarrow 0 + (9-0) * (22-01) / (81-01) = 2,36 \approx 2$
- $vec[meio] = chave$? Não, $chave > vec[meio]$
- Chamada recursiva ($início = 3$, $fim = 9$)

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

$$meio = inicio + (fim - inicio) * \frac{(chave - vec[inicio])}{(vec[fim] - vec[inicio])}$$

Exemplo: buscar o elemento alvo = 22

- $meio \leftarrow 3 + (9-3) * (22-22) / (81-22) = 3$
- $vec[meio] = chave?$ Sim!

Encontrou o alvo com duas iterações

0	1	2	3	4	5	6	7	8	9
01	15	19	22	26	32	46	55	67	81

Estrutura de Dados II

Prof. Me. Pietro M. de Oliveira