

### *Busca Binária*

### *Iterativa e Recursiva*

# Estruturas de Dados e Análise de Algoritmos

## *Busca Binária Iterativa*

- Algoritmo que faz a busca binária iterativa em um vetor;
- É um algoritmo que segue o paradigma da divisão e conquista;
- Exige que os elementos do vetor estejam ordenados;
- Baseia-se na ideia de localizar o centro do vetor a ser trabalhado;
- A cada iteração, verifica-se se o campo chave é menor, igual ou maior que o elemento do centro do vetor a ser trabalhado;
- Na próxima iteração, o vetor a ser trabalhado será uma das metades (de cima ou de baixo) do vetor trabalhado anteriormente, dependendo da comparação do campo chave com o centro do vetor encontrado.

# Estruturas de Dados e Análise de Algoritmos

## *Busca Binária Iterativa*

- No melhor caso, esse algoritmo executa 1 (uma) operação relevante, quando a chave de busca é localizada logo no primeiro índice do meio, pesquisado no vetor;
- No pior caso, esse algoritmo executa  $\log_2 N$  operações relevantes, quando a chave de busca não é localizada no vetor, onde  $N$  é o tamanho do vetor;

Conclusão: Quanto maior for a quantidade de dados a serem pesquisados, melhor será seu desempenho, comparado ao método de busca linear.

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

Vetor:

0	1	2	3	4	5
1	2	3	4	5	6

Vetor qualquer, obrigatoriamente ordenado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

Vetor:

0	1	2	3	4	5
1	2	3	4	5	6

A chave de busca será 6 (**k = 6**)

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

Vetor analisado

Vetor:

	0	1	2	3	4	5
	1	2	3	4	5	6

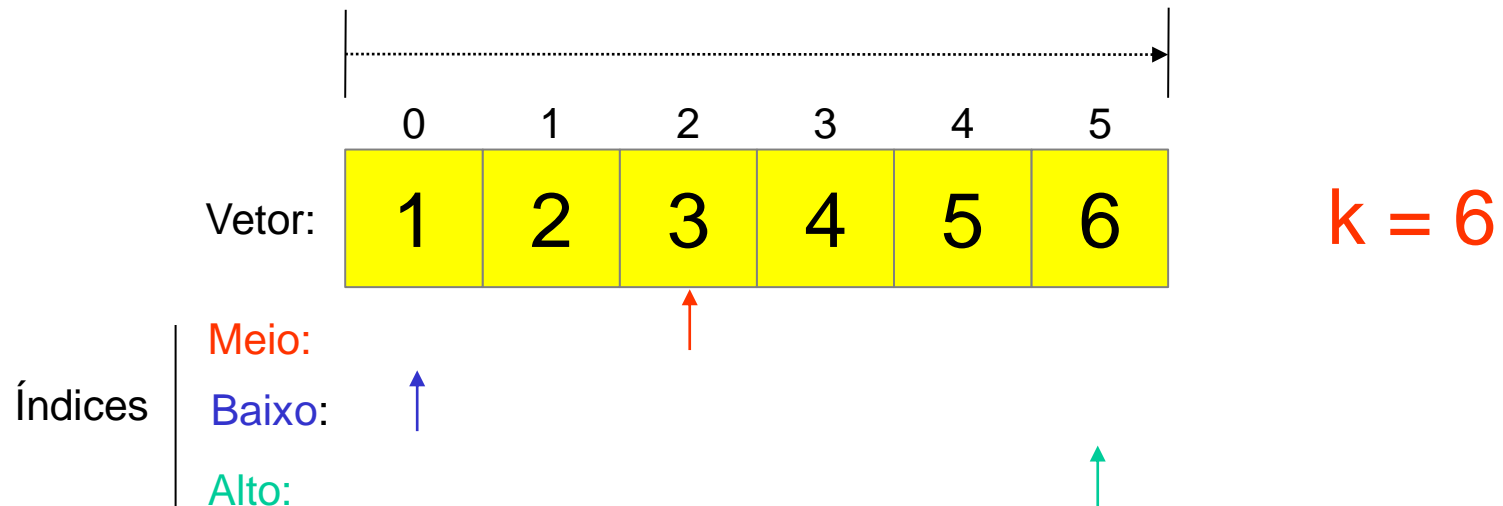
O algoritmo fará a busca binária no vetor analisado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

Vetor analisado



$$\text{Meio} = (\text{Baixo} + \text{Alto}) / 2;$$

$$\text{Meio} = (0 + 5) / 2 = 5 / 2 = 2.$$

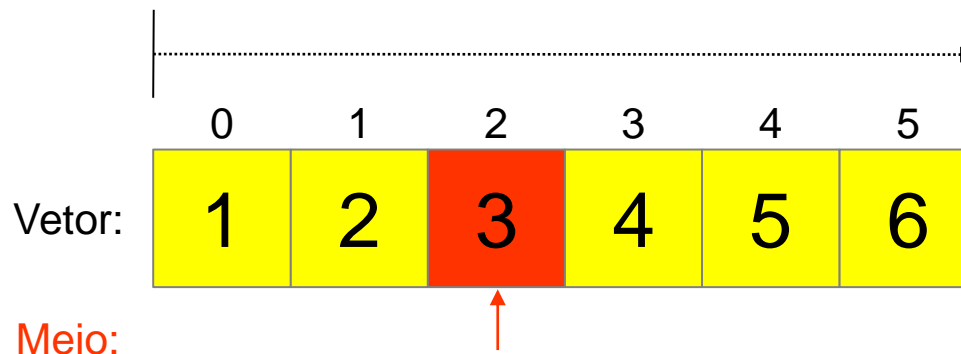
Iteração 1: Índice do meio igual à 2

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

Vetor analisado



$k = 6$

Comparação 1: Chave de busca com o meio

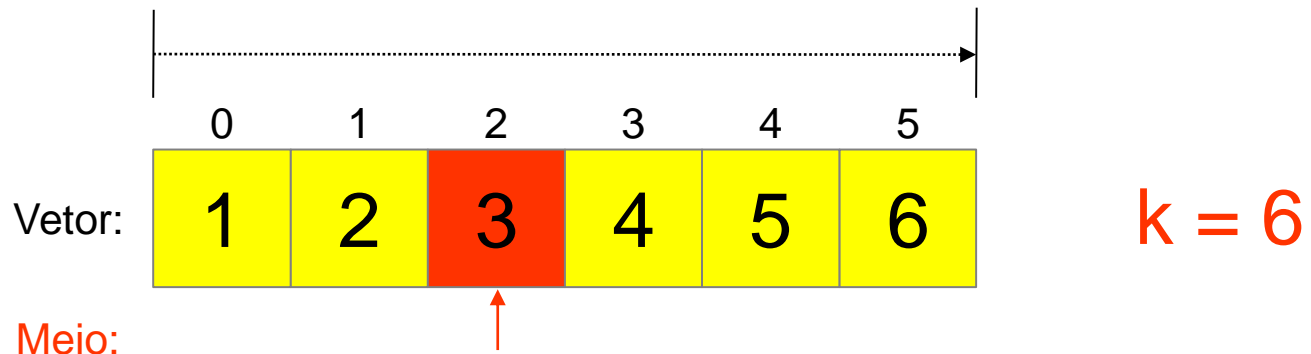


# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

Vetor analisado



Se  $6 < 3$ , metade de baixo;  
Senão, Se  $6 > 3$ , metade de cima;  
Senão, retorna o índice do meio.

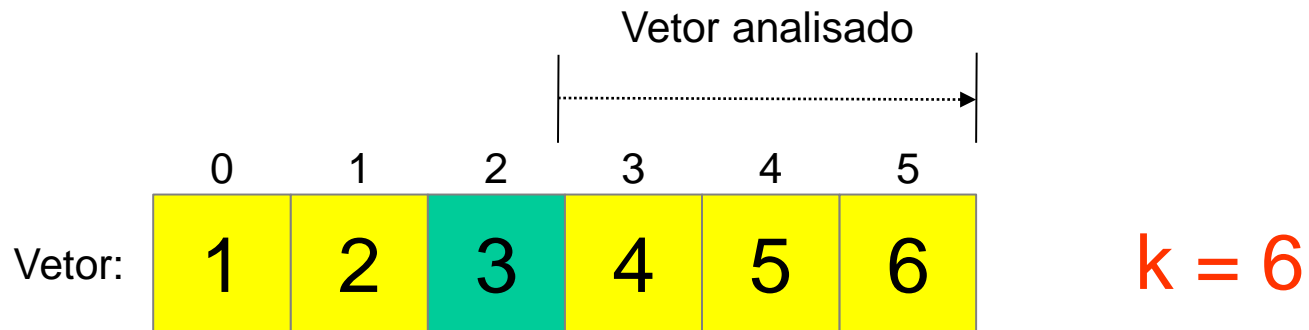
**Portanto, pode estar acima!**

A chave de busca não está no meio: outra iteração

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

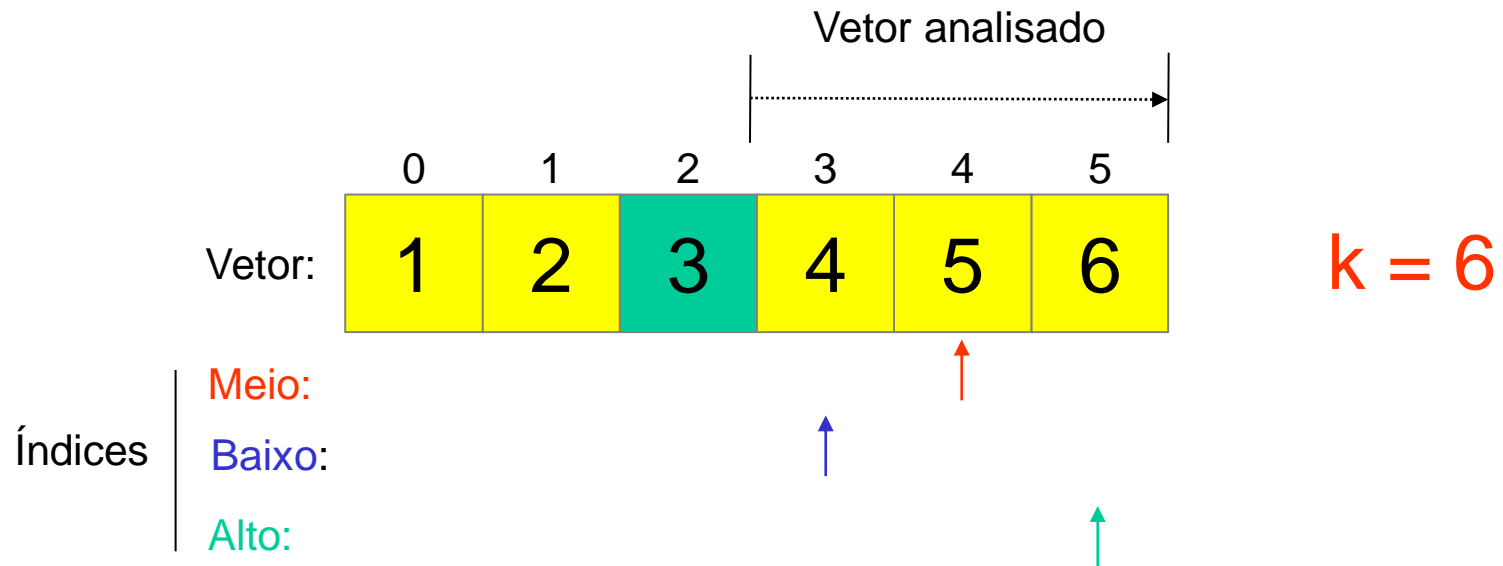


O algoritmo fará a busca binária no vetor analisado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1



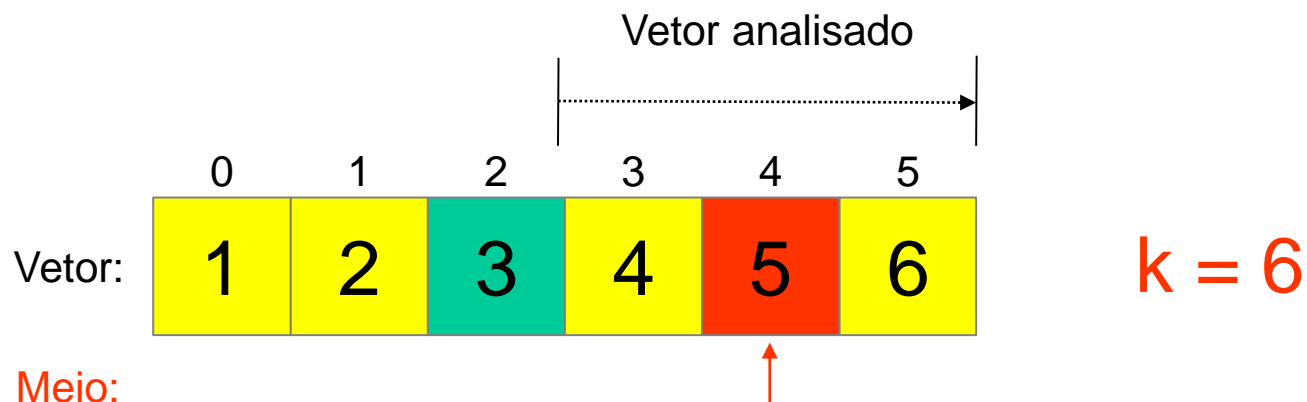
$$\text{Meio} = (\text{Baixo} + \text{Alto}) / 2;$$
$$\text{Meio} = (3 + 5) / 2 = 8 / 2 = 4.$$

Iteração 2: Índice do meio igual à 4

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

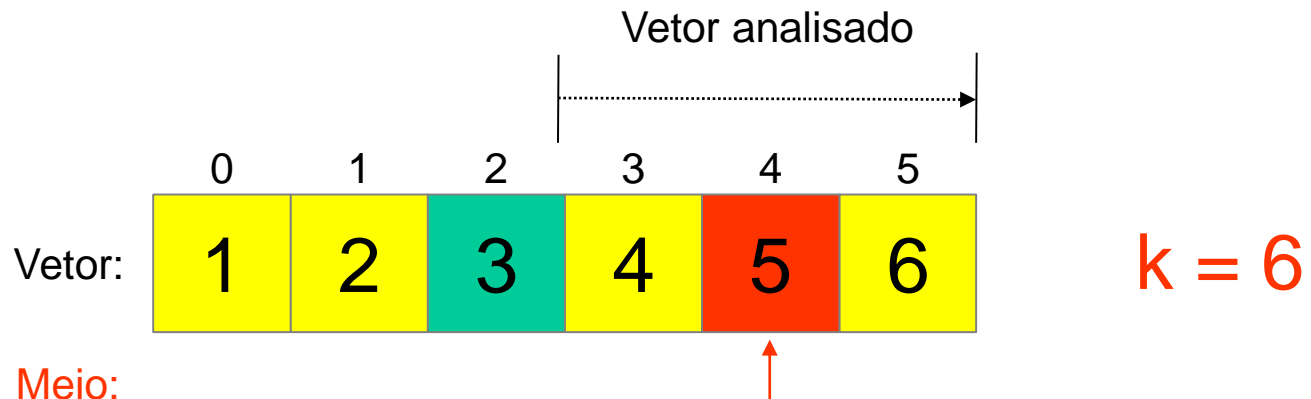


Comparação 2: Chave de busca com o meio

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1



Se  $6 < 5$ , metade de baixo;  
Senão, Se  $6 > 5$ , metade de cima;  
Senão, retorna o índice do meio.

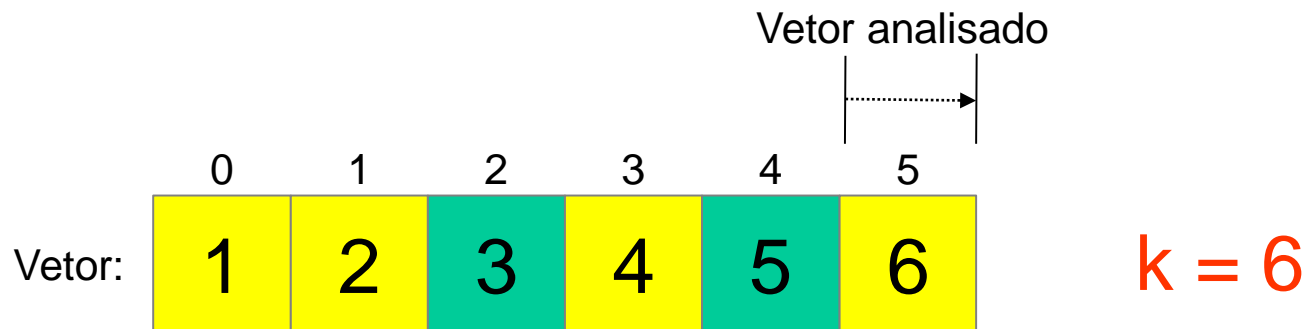
**Portanto, pode estar acima!**

A chave de busca não está no meio: outra iteração

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1

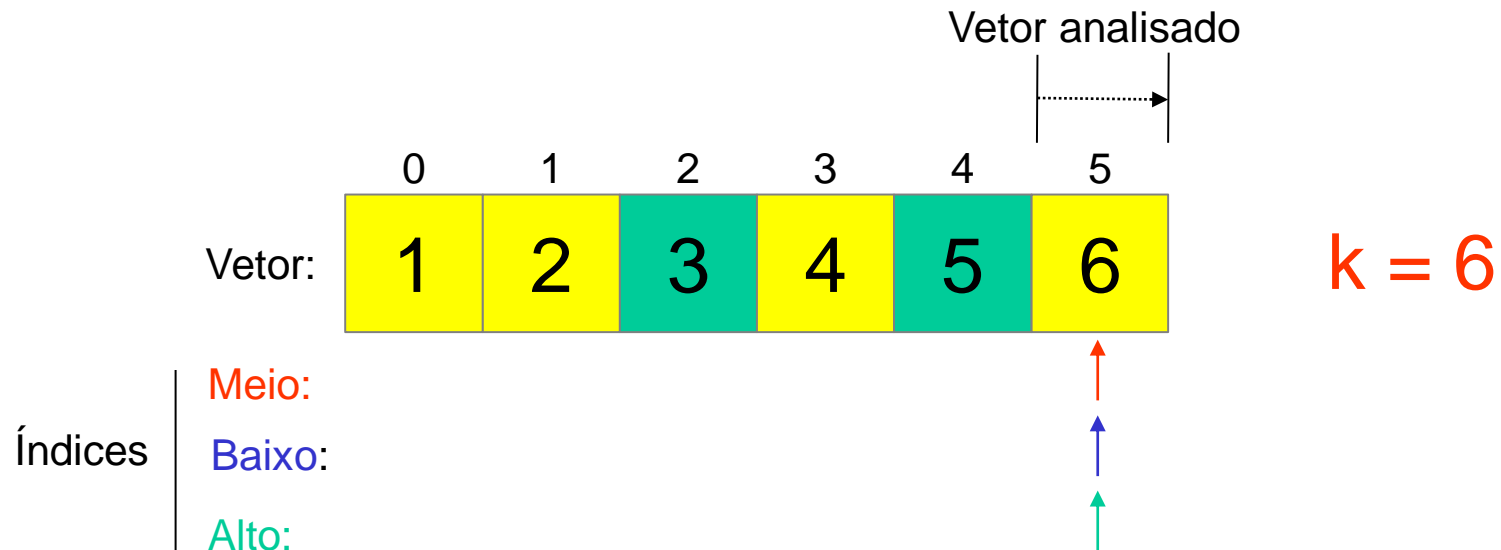


O algoritmo fará a busca binária no vetor analisado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1



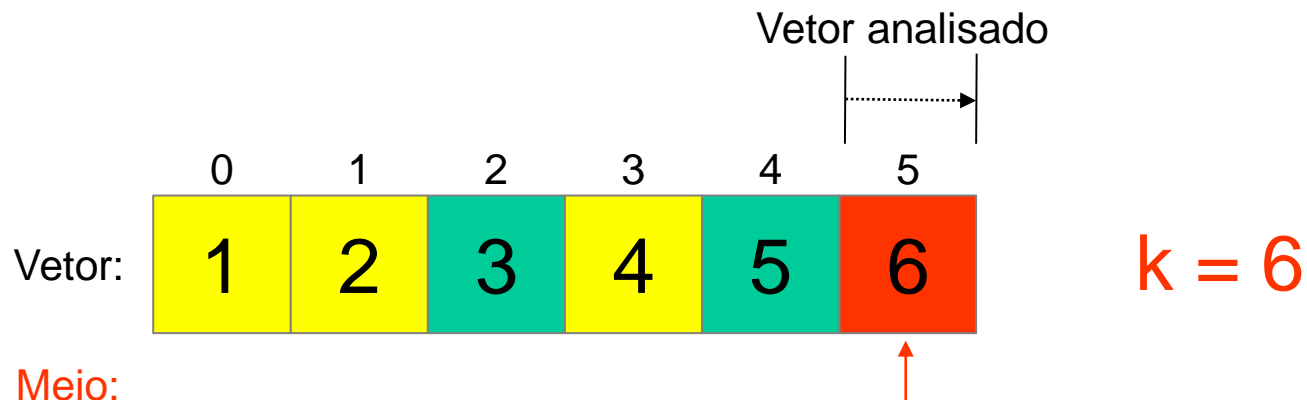
$$\text{Meio} = (\text{Baixo} + \text{Alto}) / 2;$$
$$\text{Meio} = (5 + 5) / 2 = 10 / 2 = 5.$$

Iteração 3: Índice do meio igual à 5

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1



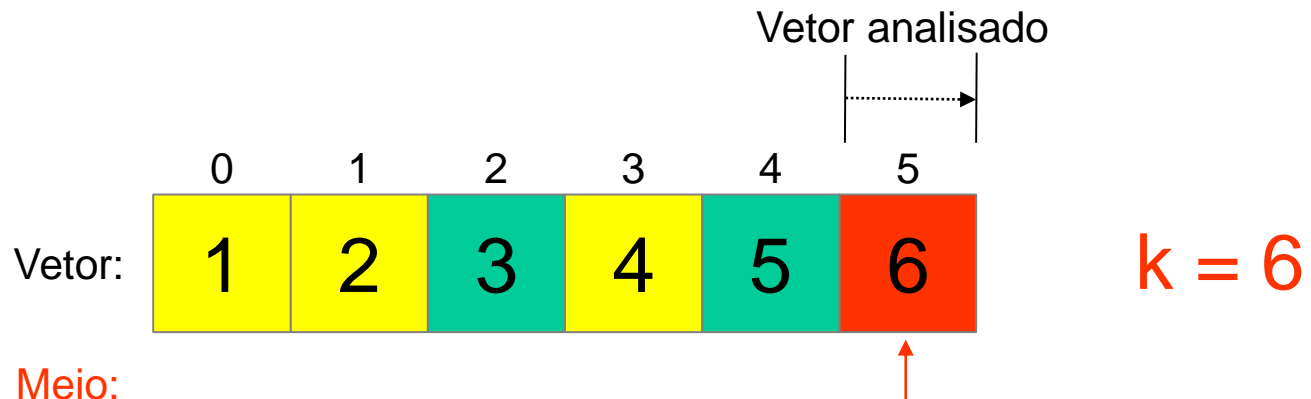
Comparação 3: Chave de busca com o meio



# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1



Se  $6 < 6$ , metade de baixo;  
Senão, Se  $6 > 6$ , metade de cima;  
Senão, retorna o índice do meio.

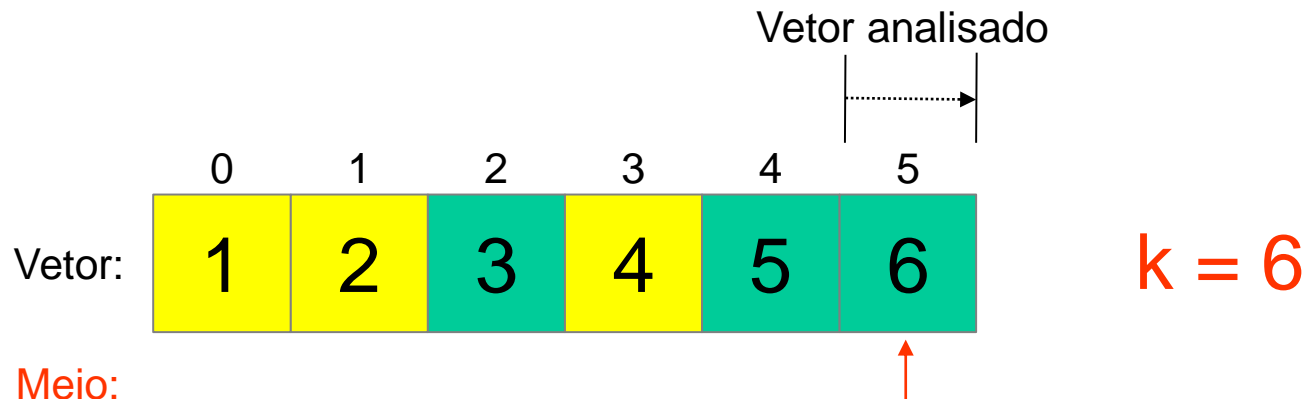
**Portanto, retorna o índice do meio: 5**

A chave de busca está no meio: final das iterações

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 1



Retorno: índice = 5;

**Final do algoritmo!**

A chave de busca foi encontrada no índice 5 do vetor!

## *Busca Binária Iterativa*

Outro exemplo...

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

Vetor:

0	1	2	3	4	5
1	2	3	4	5	6

Vetor qualquer, obrigatoriamente ordenado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

Vetor:

0	1	2	3	4	5
1	2	3	4	5	6

A chave de busca será 8 (**k = 8**)

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

Vetor analisado

Vetor:

	0	1	2	3	4	5
	1	2	3	4	5	6

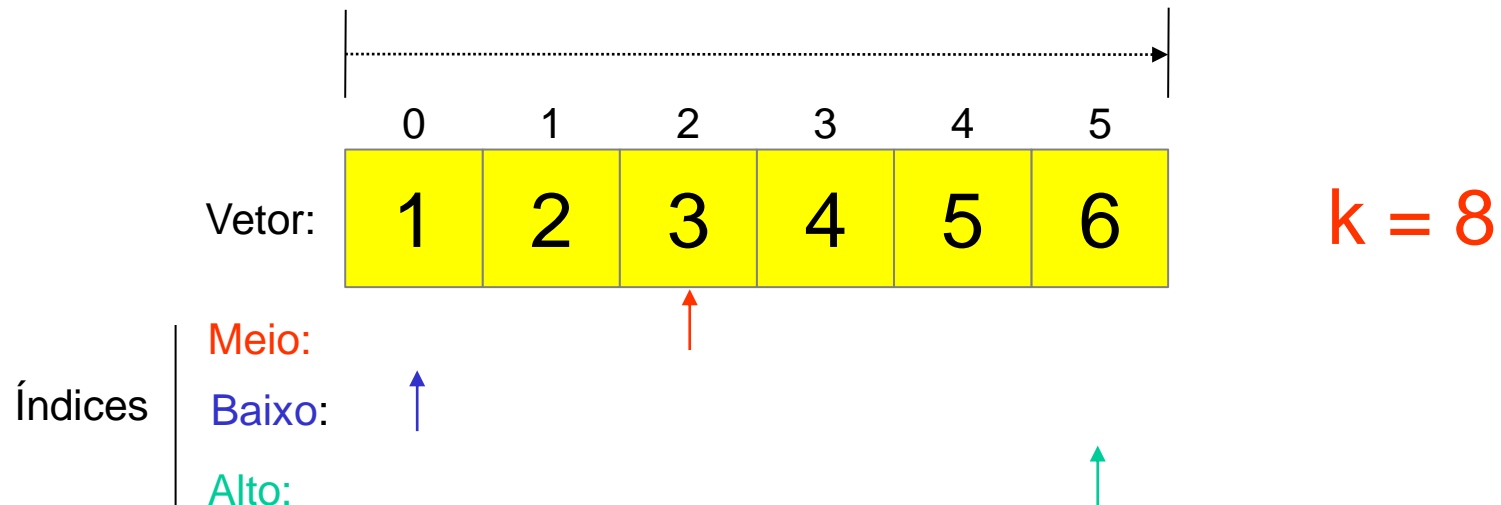
O algoritmo fará a busca binária no vetor analisado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

Vetor analisado



$$\text{Meio} = (\text{Baixo} + \text{Alto}) / 2;$$

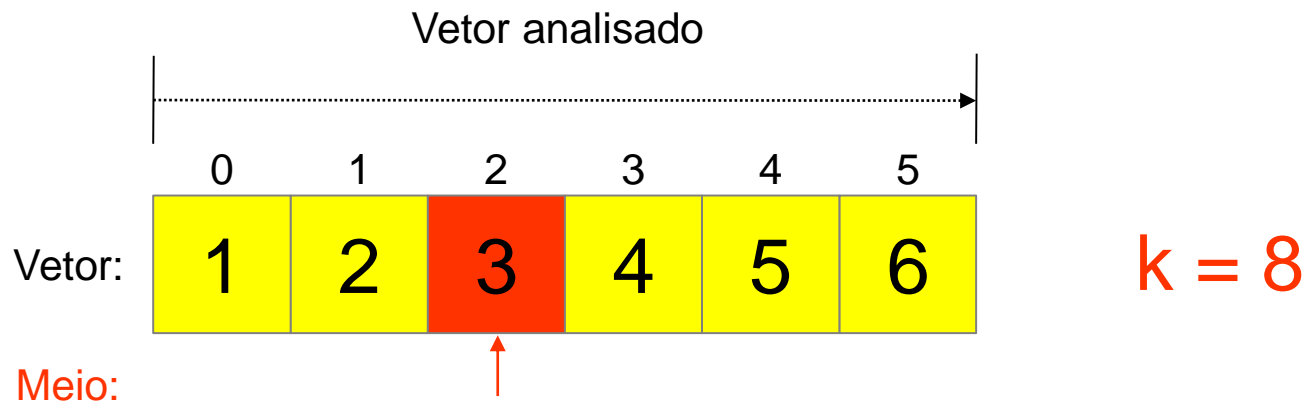
$$\text{Meio} = (0 + 5) / 2 = 5 / 2 = 2.$$

Iteração 1: Índice do meio igual à 2

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



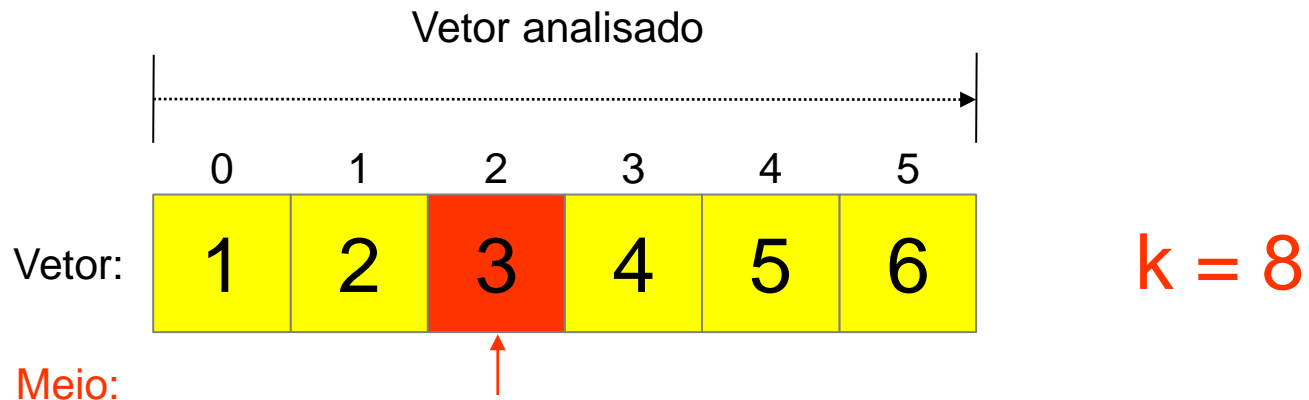
Comparação 1: Chave de busca com o meio



# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



Se  $8 < 3$ , metade de baixo;  
Senão, Se  $8 > 3$ , metade de cima;  
Senão, retorna o índice do meio.

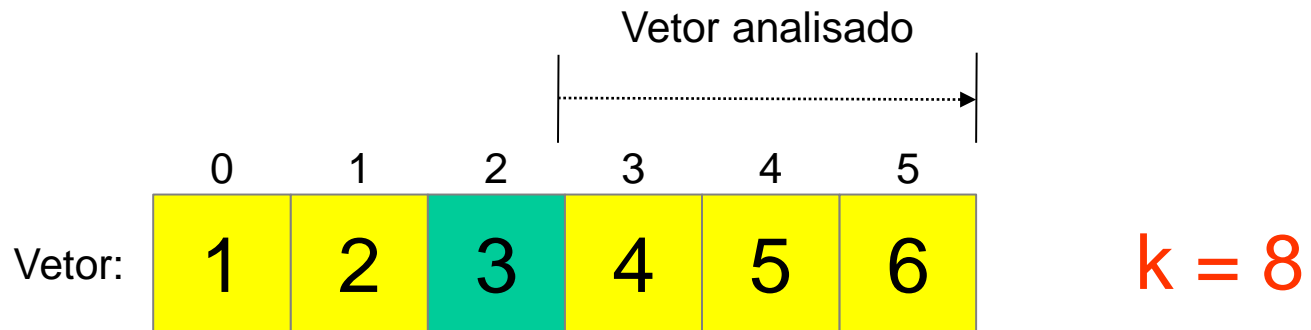
**Portanto, pode estar acima!**

A chave de busca não está no meio: outra iteração

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

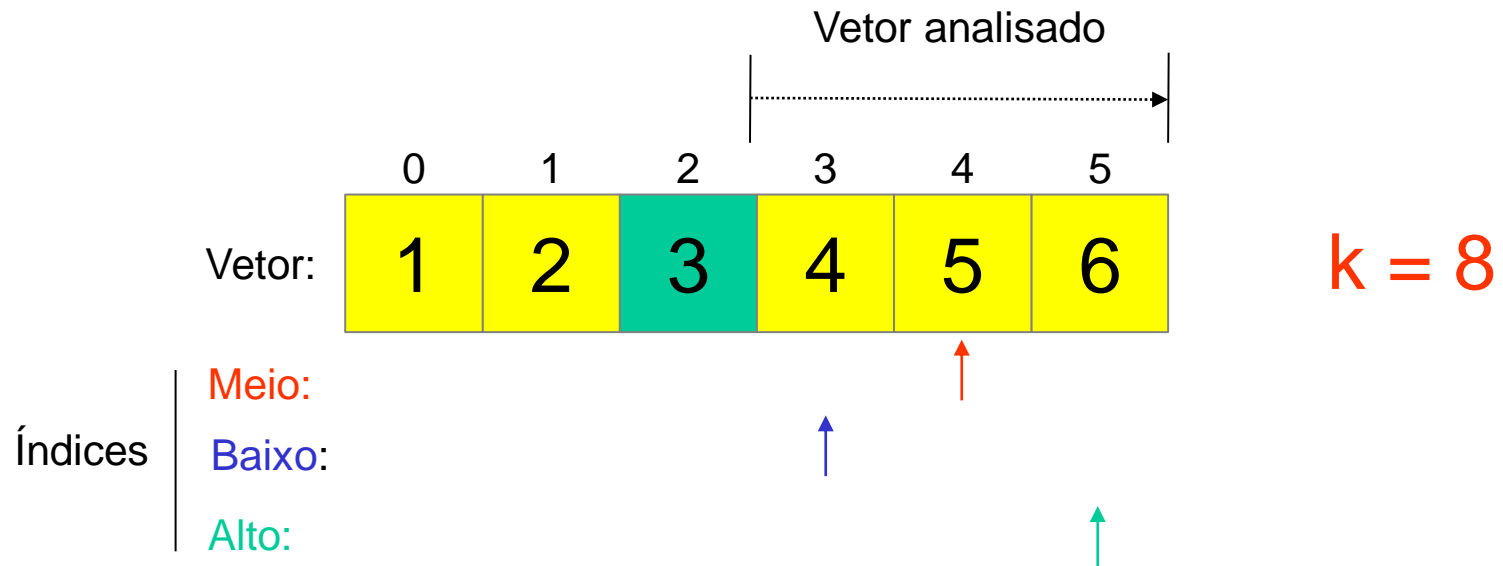


O algoritmo fará a busca binária no vetor analisado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



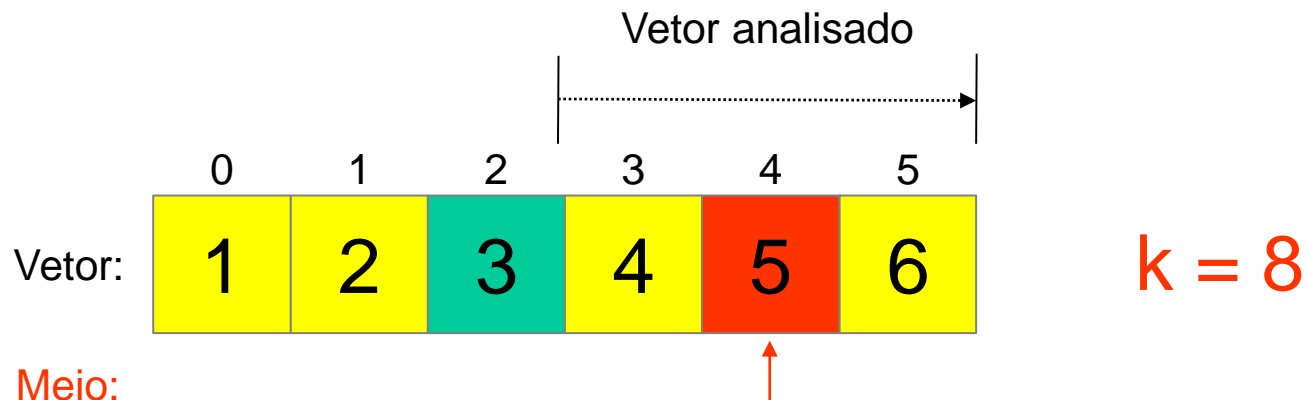
$$\text{Meio} = (\text{Baixo} + \text{Alto}) / 2;$$
$$\text{Meio} = (3 + 5) / 2 = 8 / 2 = 4.$$

Iteração 2: Índice do meio igual à 4

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

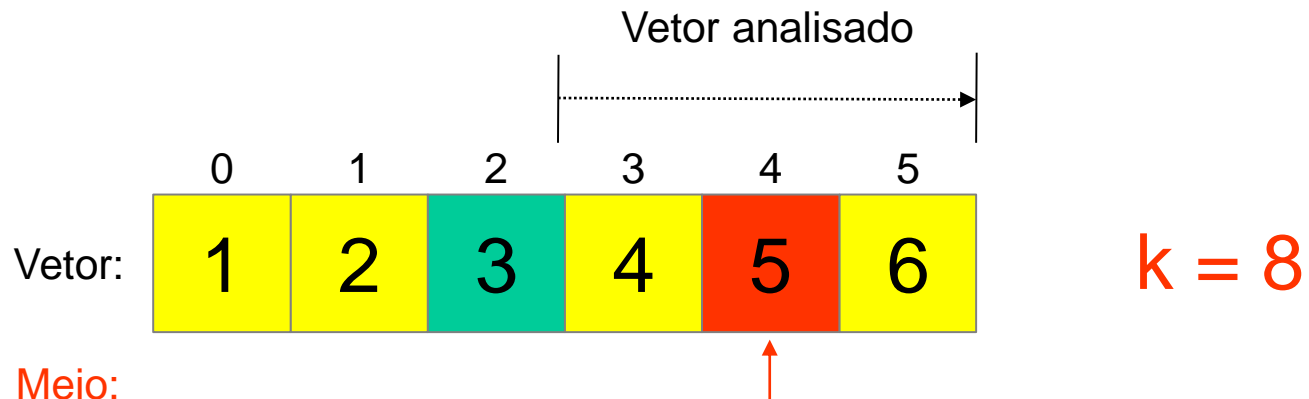


Comparação 2: Chave de busca com o meio

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



Se  $8 < 5$ , metade de baixo;  
Senão, Se  $8 > 5$ , metade de cima;  
Senão, retorna o índice do meio.

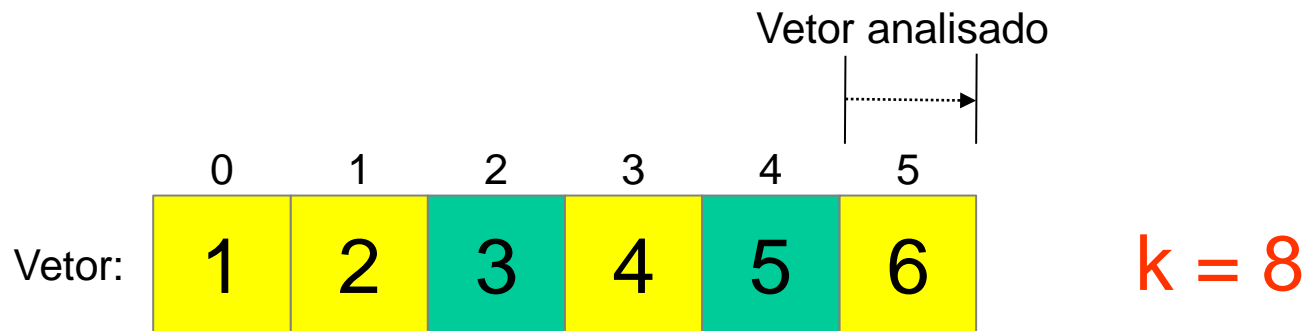
**Portanto, pode estar acima!**

A chave de busca não está no meio: outra iteração

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

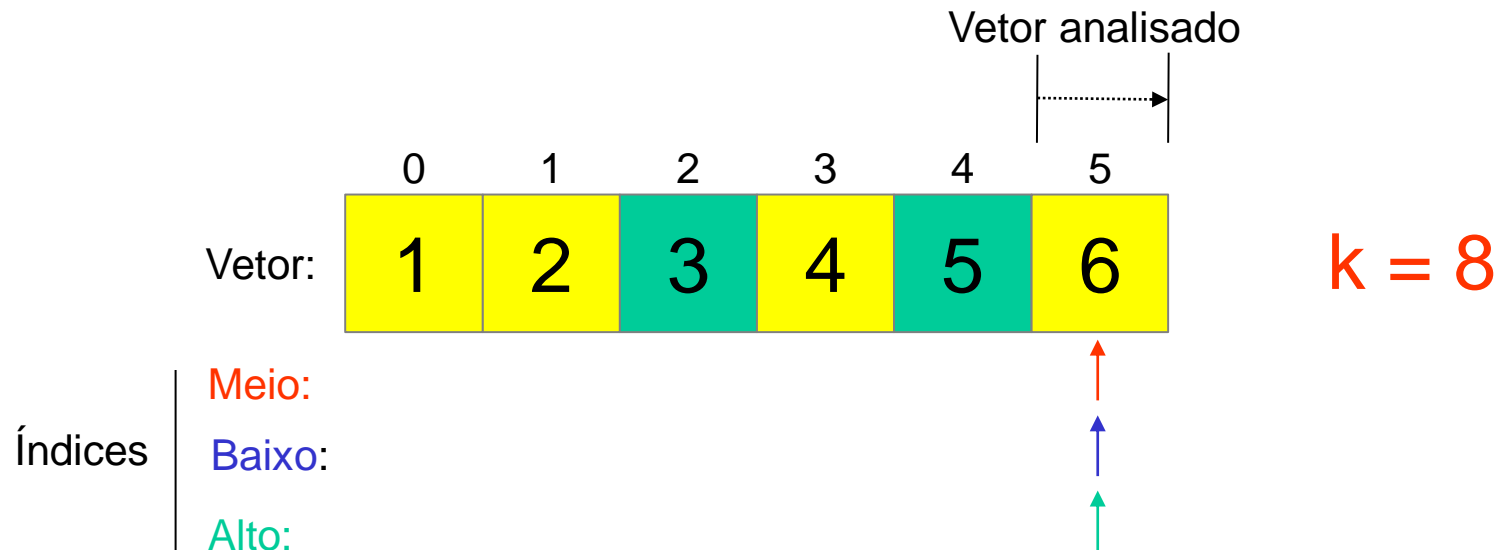


O algoritmo fará a busca binária no vetor analisado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



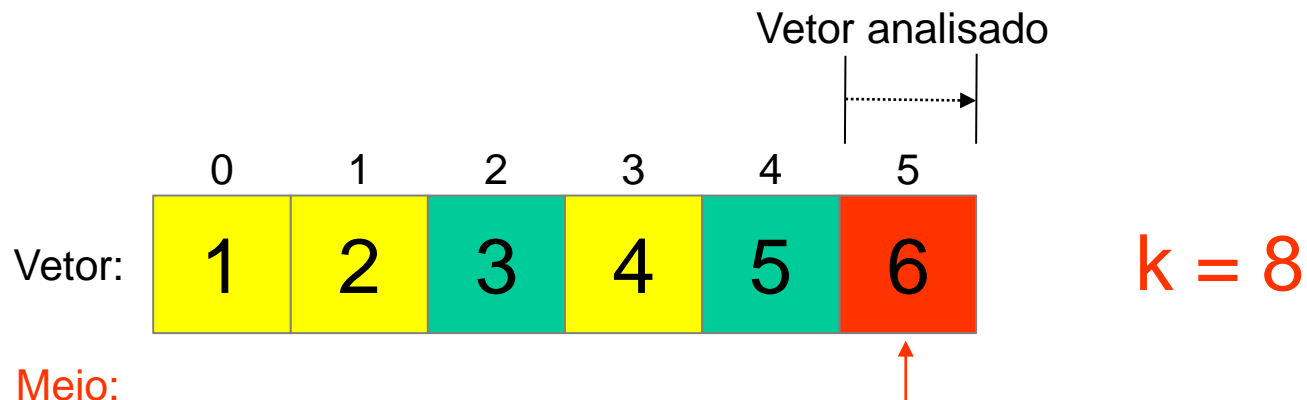
$$\text{Meio} = (\text{Baixo} + \text{Alto}) / 2;$$
$$\text{Meio} = (5 + 5) / 2 = 10 / 2 = 5.$$

Iteração 3: Índice do meio igual à 5

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



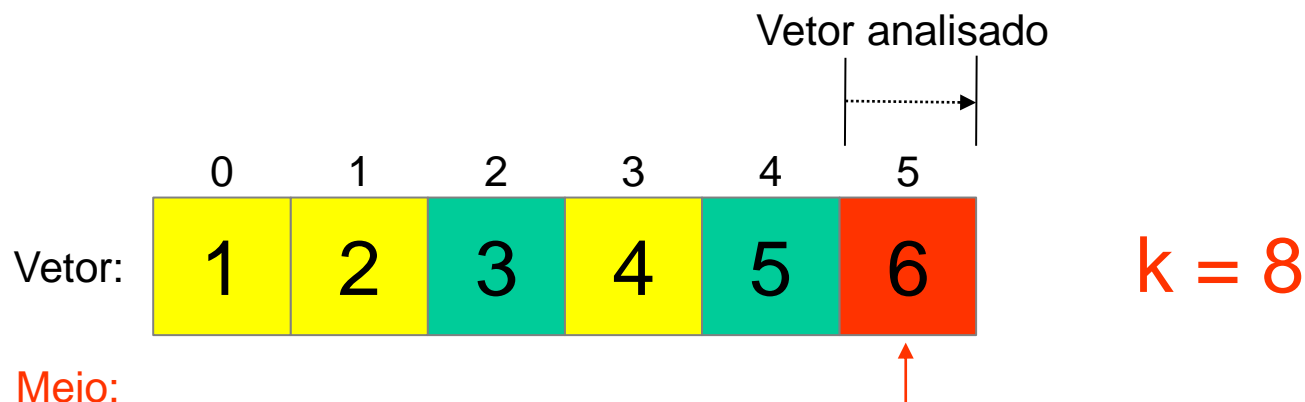
Comparação 3: Chave de busca com o meio



# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



Se  $8 < 6$ , metade de baixo;  
Senão, Se  $8 > 6$ , metade de cima;  
Senão, retorna o índice do meio.

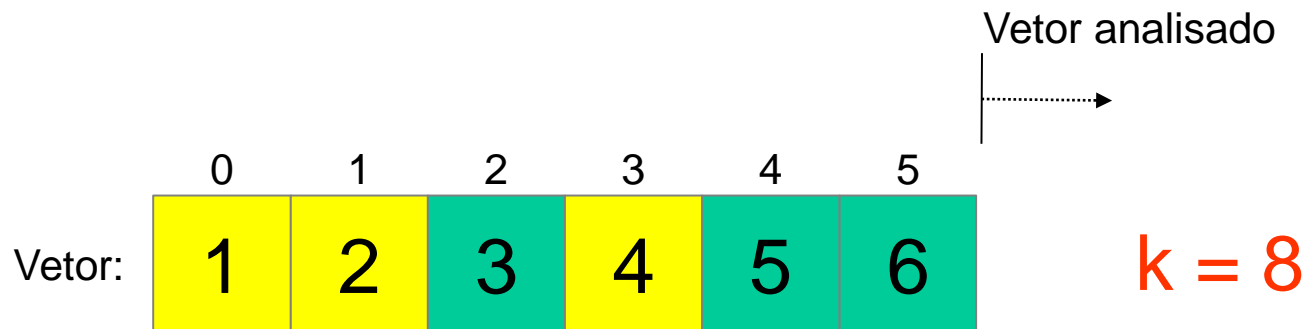
**Portanto, pode estar acima!**

A chave de busca não está no meio: outra iteração

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2

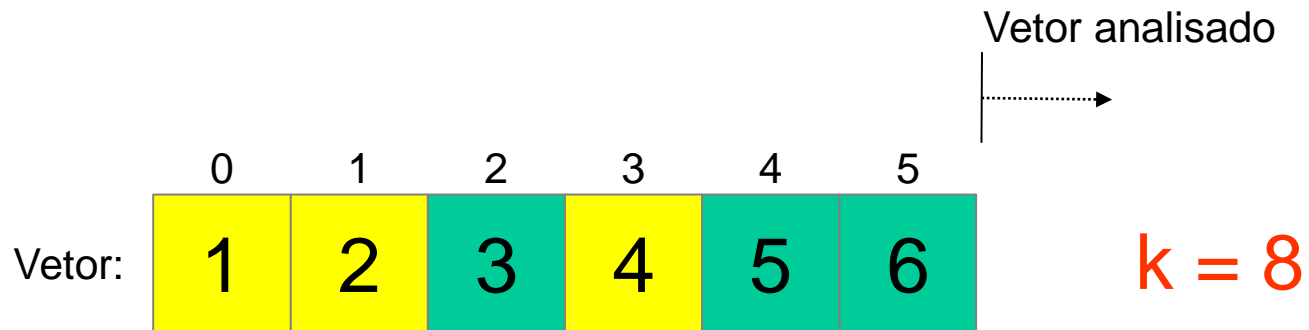


O algoritmo fará a busca binária no vetor analisado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Iterativa

### Exemplo 2



Retorno: índice = 6 ou -1;

**Final do algoritmo!**

A chave de busca não foi encontrada no vetor!

## Busca Binária Iterativa

### Exemplo de Código

```
public static int busca_binaria(int iVet[], int iK)
{
    int iBaixo, iAlto, iMeio;

    iBaixo=0;
    iAlto=iVet.length-1;
    while(iBaixo <= iAlto)
    {
        iMeio=(iBaixo+iAlto)/2;
        if(iK < iVet[iMeio]) iAlto=iMeio-1;
        else if(iK > iVet[iMeio]) iBaixo=iMeio+1;
        else return iMeio;
    }
    return -1;
}
```

Método iterativo *busca\_binaria()*

## *Buscas Recursivas*

- As buscas, como todos os outros algoritmos iterativos, podem ser implementadas utilizando-se os algoritmos recursivos equivalentes;
- As buscas recursivas são algoritmos simples e eficientes;
- A cada recursão, verifica-se se a chave de busca é igual ao elemento armazenado no vetor;
- Caso a chave de busca seja encontrada, encerra-se a busca, retornando-se o índice do vetor onde o elemento foi encontrado;
- Caso a chave de busca não seja encontrada, encerra-se a busca, retornando-se o índice máximo do vetor acrescido de um (tamanho do vetor).

# Estruturas de Dados e Análise de Algoritmos

## *Busca Binária Recursiva*

### Exemplo 3

Vetor:

0	1	2	3	4	5
10	20	30	40	50	60

Vetor qualquer, supostamente ordenado!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3

Vetor:

0	1	2	3	4	5
10	20	30	40	50	60



O algoritmo fará a busca no vetor, crescentemente

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3

Vetor:

0	1	2	3	4	5
10	20	30	40	50	60

```
public static int busca(int iV[], int iB, int iA, int iK)
{
    int iM = (iB + iA)/2;
    if(iB <= iA)
        if(iK < iV[iM])    return busca(iV, iB, iM - 1, iK);
        else if(iK > iV[iM])    return busca(iV, iM + 1, iA, iK);
        else return iM;
    return -1;
}
```

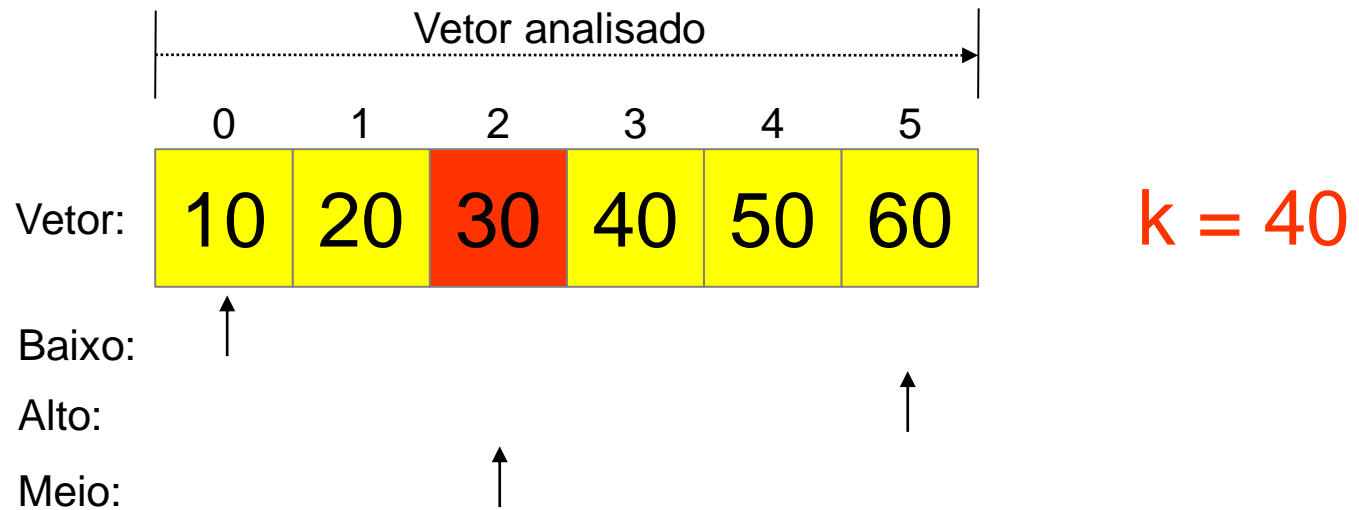
A chave de busca será 40 (k = 40)



# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3



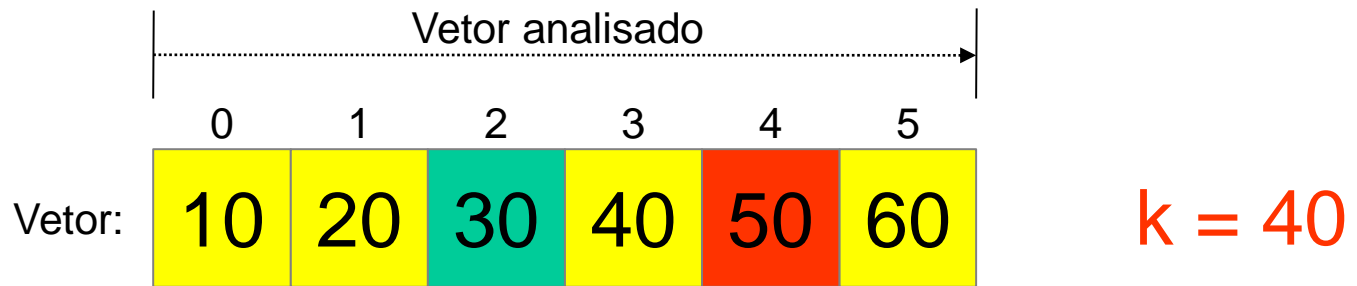
```
public static int busca(iV, 0, 5, 40)
{
    int 2 = (0 + 5) / 2;
    if (0 <= 5)
        if (40 < 30) return busca(iV, 0, 1, 40);
        else if (40 > 30) return busca(iV, 3, 5, 40);
        else return 2;
    return -1;
}
```

Recursão 1: Busca  $k=40$  no Meio  $= (5+0) / 2 = 2$

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3



Baixo:

Alto:

Meio:

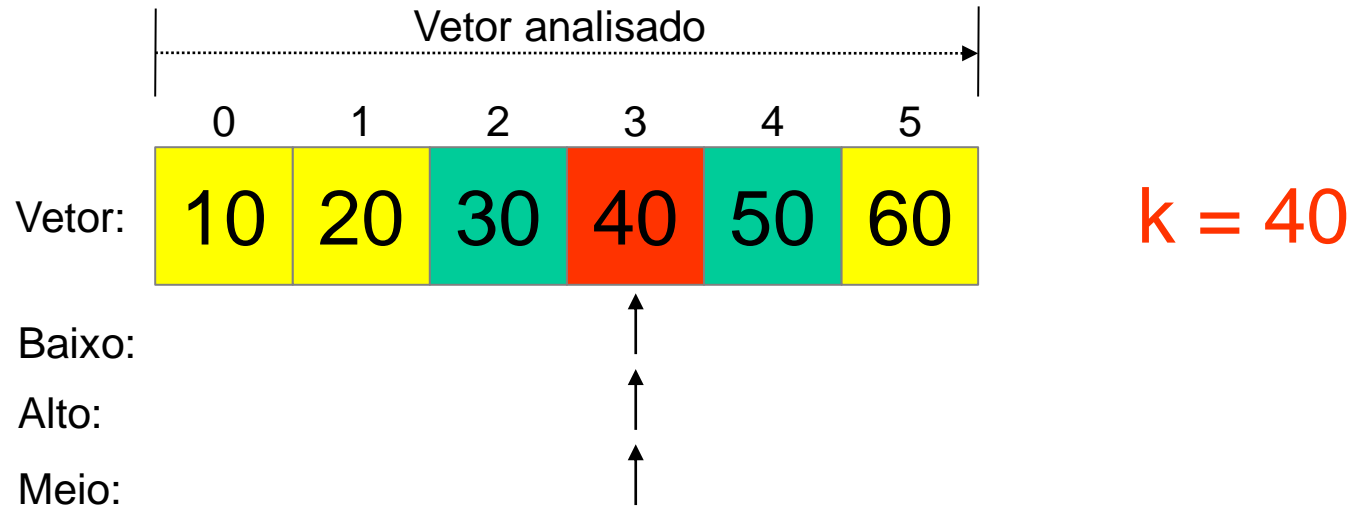
```
public static int busca(iV, 3, 5, 40)
{
    int 4 = (3 + 5) / 2;
    if(3 <= 5)
        if(40 < 50) return busca(iV, 3, 3, 40);
        else if(40 > 50) return busca(iV, 5, 5, 40);
        else return 4;
    return -1;
}
```

Recursão 2: Busca  $k=40$  no Meio  $= (5+3) / 2 = 4$

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3



```
public static int busca(iV, 3, 3, 40)
{
    int 3 = (3 + 3)/2;
    if(3 <= 3)
        if(40 < 40) return busca(iV, 3, 2, 40);
        else if(40 > 40) return busca(iV, 4, 3, 40);
        else return 3;
    return -1;
}
```

Recursão 3: Busca k=40 no Meio=3; Retorna Meio=3

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3



Baixo:

Alto:

Meio:

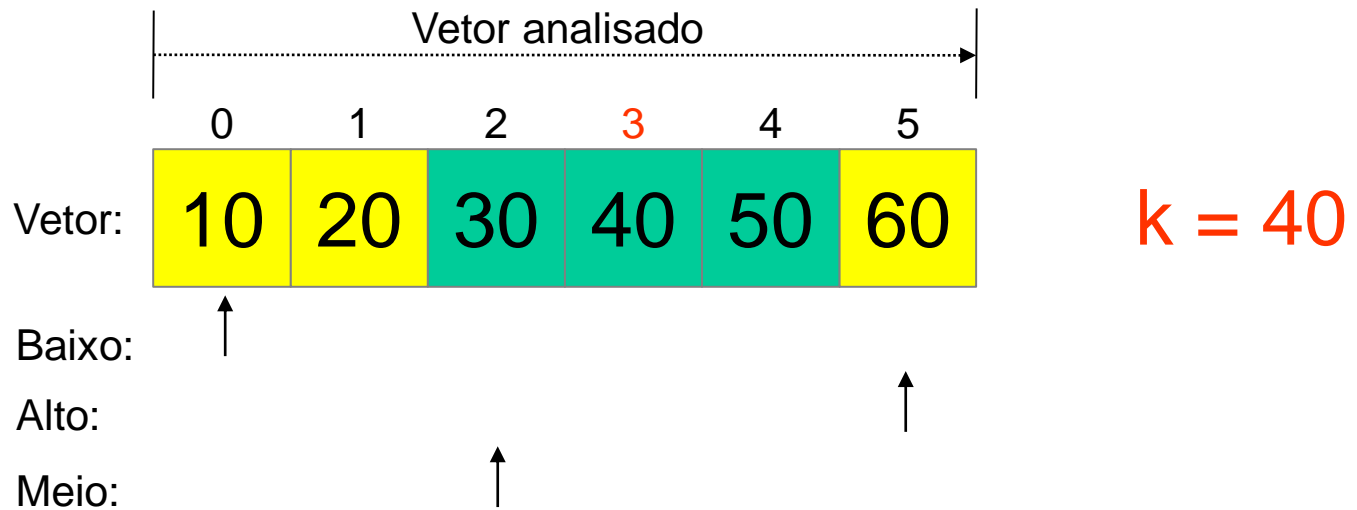
```
public static int busca(iV, 3, 5, 40)
{
    int 4 = (3 + 5) / 2;
    if(3 <= 5)
        if(40 < 50) return busca(iV, 3, 3, 40);
        else if(40 > 50) return busca(iV, 5, 5, 40);
        else return 4;
    return -1;
}
```

Retorna para a Recursão 2 com índice=3

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3



```
public static int busca(iV, 0, 5, 40)
{
    int 2 = (0 + 5)/2;
    if(0 <= 5)
        if(40 < 30) return busca(iV, 0, 1, 40);
        else if(40 > 30) return busca(iV, 3, 5, 40);
        else return 2;
    return -1;
}
```

Retorna para a Recursão 1 com índice=3

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 3



```
public static int busca(int iV[], int iB, int iA, int iK)
{
    int iM = (iB + iA)/2;
    if(iB <= iA)
        if(iK < iV[iM])    return busca(iV, iB, iM - 1, iK);
        else if(iK > iV[iM])    return busca(iV, iM + 1, iA, iK);
        else return iM;
    return -1;
}
```

A chave de busca foi encontrada no índice = 3!

# Estruturas de Dados e Análise de Algoritmos

## *Busca Binária Recursiva*

Outro exemplo...

# Estruturas de Dados e Análise de Algoritmos

## *Busca Binária Recursiva*

### Exemplo 4

Vetor:

0	1	2	3	4	5
10	20	30	40	50	60

Vetor qualquer, supostamente ordenado!



# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4

Vetor:

0	1	2	3	4	5
10	20	30	40	50	60



O algoritmo fará a busca no vetor, crescentemente

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4

Vetor:

0	1	2	3	4	5
10	20	30	40	50	60

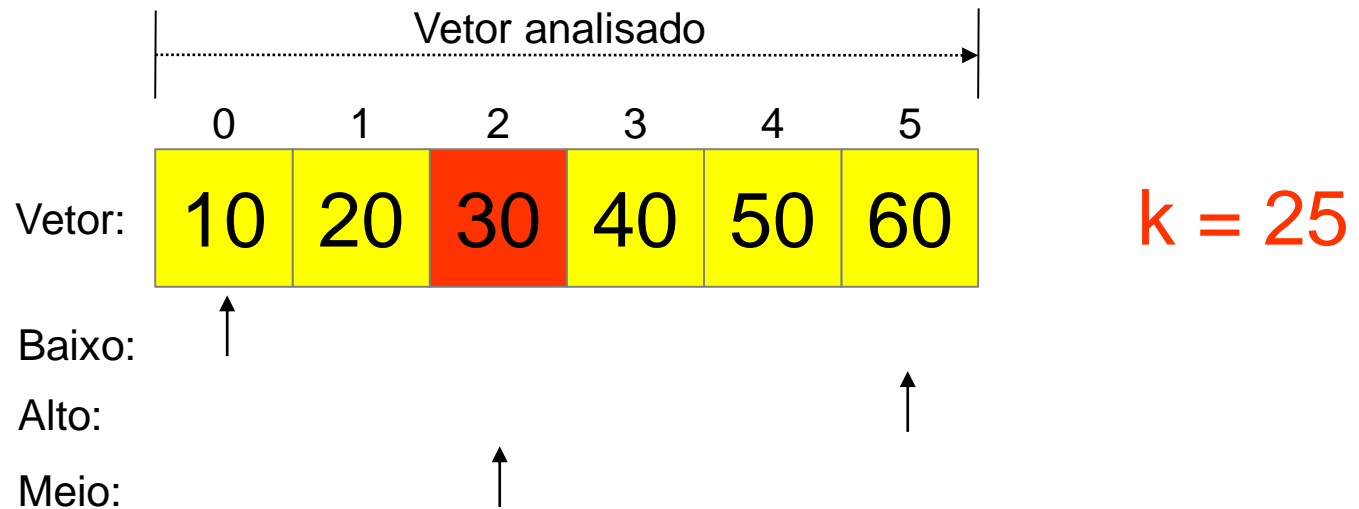
```
public static int busca(int iV[], int iB, int iA, int iK)
{
    int iM = (iB + iA)/2;
    if(iB <= iA)
        if(iK < iV[iM])    return busca(iV, iB, iM - 1, iK);
        else if(iK > iV[iM]) return busca(iV, iM + 1, iA, iK);
        else return iM;
    return -1;
}
```

A chave de busca será 25 ( $k = 25$ )

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4



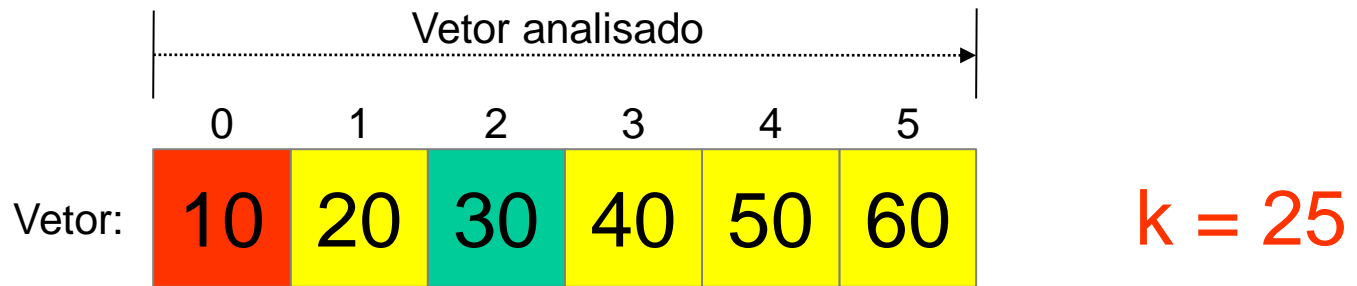
```
public static int busca(iV, 0, 5, 25)
{
    int 2 = (0 + 5)/2;
    if(0 <= 5)
        if(25 < 30) return busca(iV, 0, 1, 25);
        else if(25 > 30) return busca(iV, 3, 5, 25);
        else return 2;
    return -1;
}
```

Recursão 1: Busca  $k = 25$  no Meio  $= (5 + 0) / 2 = 2$

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4



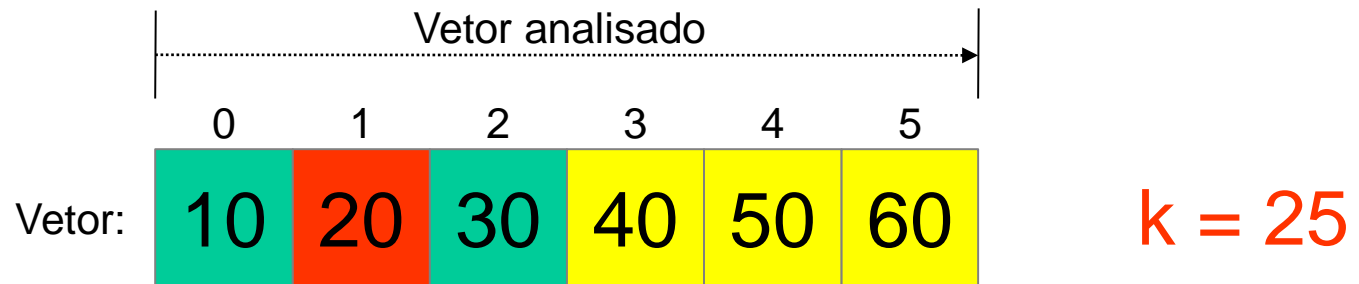
```
public static int busca(iV, 0, 1, 25)
{
    int 0 = (0 + 1)/2;
    if(0 <= 1)
        if(25 < 10) return busca(iV, 0, -1, 25);
        else if(25 > 10) return busca(iV, 1, 1, 25);
        else return 0;
    return -1;
}
```

Recursão 2: Busca  $k = 25$  no Meio  $= (0 + 1) / 2 = 0$

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4



Baixo:

Alto:

Meio:

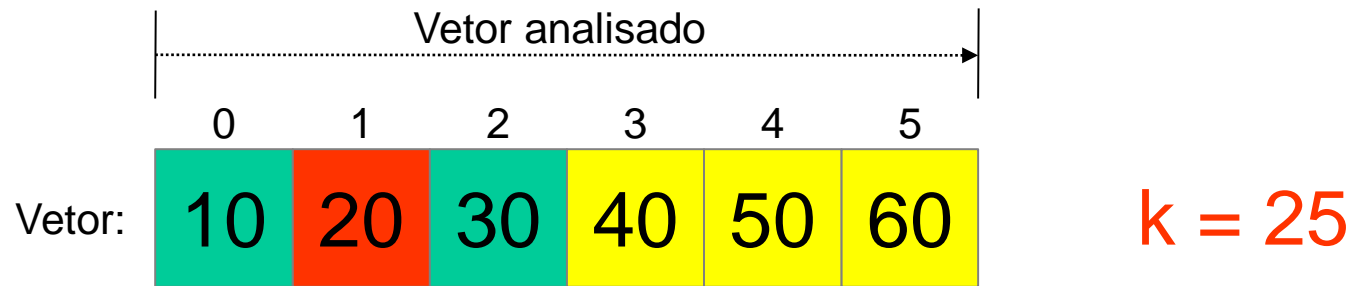
```
public static int busca(iV, 1, 1, 25)
{
    int l = (1 + 1) / 2;
    if (l <= 1)
        if (25 < 20) return busca(iV, 1, 0, 25);
        else if (25 > 20) return busca(iV, 2, 1, 25);
        else return 1;
    return -1;
}
```

Recursão 3: Busca  $k = 25$  no Meio  $= (1 + 1) / 2 = 1$

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4



Baixo:

Alto:

Meio:

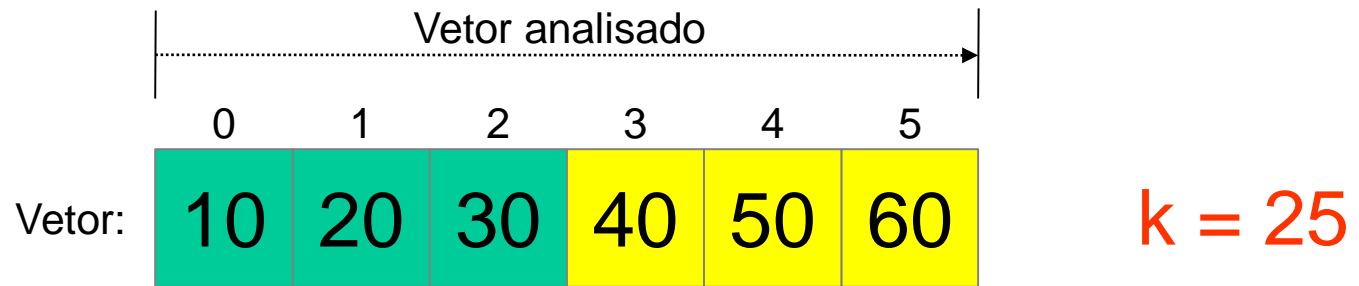
```
public static int busca(iV, 2, 1, 25)
{
    int 1 = (2 + 1)/2;
    if(2 <= 1)
        if(25 < 20) return busca(iV, 2, 0, 25);
        else if(25 > 20) return busca(iV, 2, 1, 25);
        else return 1;
    return -1;
}
```

Recursão 4: Não achou! retorna índice = -1

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4



Baixo:

Alto:

Meio:

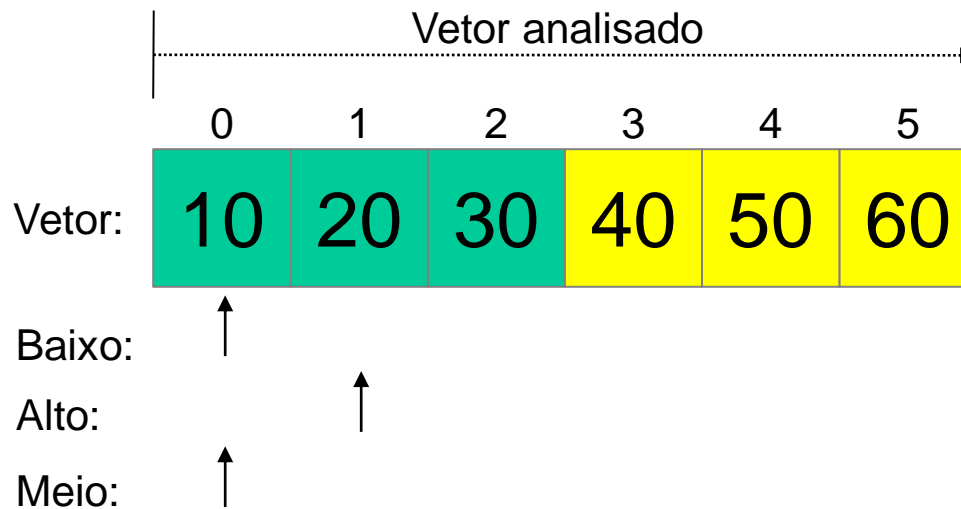
```
public static int busca(iV, 1, 1, 25)
{
    int l = (1 + 1) / 2;
    if (l <= 1)
        if (25 < 20) return busca(iV, 1, 0, 25);
        else if (25 > 20) return busca(iV, 2, 1, 25);
        else return 1;
    return -1;
}
```

Retorna para a Recursão 3 com índice = -1

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4



$k = 25$

```
public static int busca(iV, 0, 1, 25)
{
    int 0 = (0 + 1)/2;
    if(0 <= 1)
        if(25 < 10) return busca(iV, 0, -1, 25);
        else if(25 > 10) return busca(iV, 1, 1, 25);
        else return 0;
    return -1;
}
```

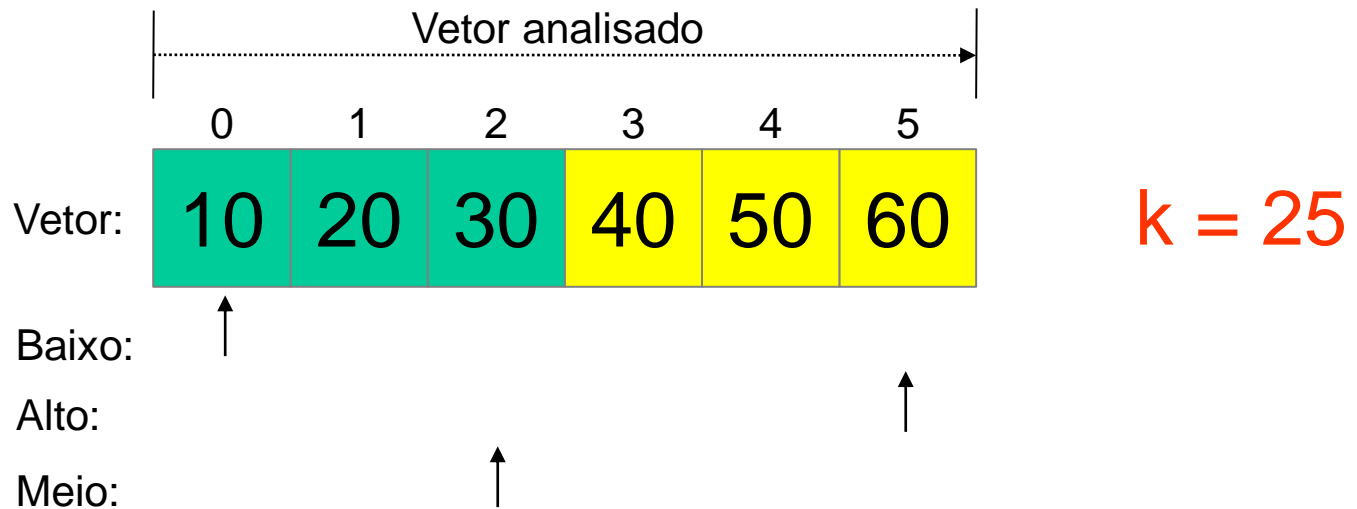
Retorna para a Recursão 2 com índice = -1



# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4



```
public static int busca(iV, 0, 5, 25)
{
    int 2 = (0 + 5)/2;
    if(0 <= 5)
        if(25 < 30) return busca(iV, 0, 1, 25);
        else if(25 > 30) return busca(iV, 3, 5, 25);
        else return 2;
    return -1;
}
```

Retorna para a Recursão 1 com índice = -1

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Exemplo 4

Vetor:

0	1	2	3	4	5
10	20	30	40	50	60

**k = 25**

**int indice = -1;**

A chave não foi encontrada no vetor!

# Estruturas de Dados e Análise de Algoritmos

## Busca Binária Recursiva

### Código dos Exemplos 3 e 4

```
import javax.swing.*;

public class BuscaLinearRecursiva
{
    public static void main(String Args[])
    {
        int iVetor[] = new int[] {10, 20, 30, 40, 50, 60};
        int iIndice, iK;

        iK = Integer.parseInt(JOptionPane.showInputDialog("Digite a chave de busca:"));

        iIndice = busca(iVetor, 0, iVetor.length-1, iK);

        String sTitle="Busca Linear Recursiva - Saída";
        String sIO="O número " + iK;

        if(iIndice >= 0)
            sIO += " está localizado no índice " + iIndice + " do vetor.";
        else sIO += " não está localizado no vetor.";
        JOptionPane.showMessageDialog(null,sIO,sTitle,JOptionPane.INFORMATION_MESSAGE);

        System.exit(0);
    }

    public static int busca(int iV[], int iB, int iA, int iK)
    {
        int iM = (iB + iA)/2;
        if(iB <= iA)
            if(iK < iV[iM]) return busca(iV, iB, iM - 1, iK);
            else if(iK > iV[iM]) return busca(iV, iM + 1, iA, iK);
            else return iM;
        return -1;
    }
}
```

# Estruturas de Dados e Análise de Algoritmos

## *Busca Binária Iterativa e Recursiva*

FIM