

Busca Binária

Algoritmos e Estruturas de Dados III
(CI1057)

Prof. Eduardo Almeida

Objetivo

- Buscar por valores em vetor ordenado
- Continuamente dividir o espaço de busca pela metade

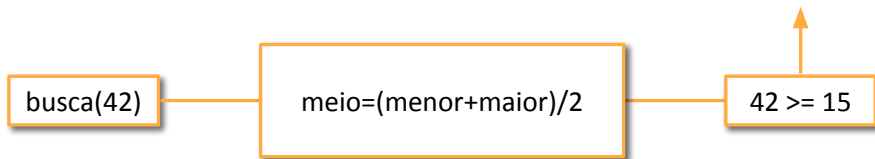
Suposição

- Vetor ordenado

Busca binária

- Considere o conjunto de dados abaixo (ordenado):

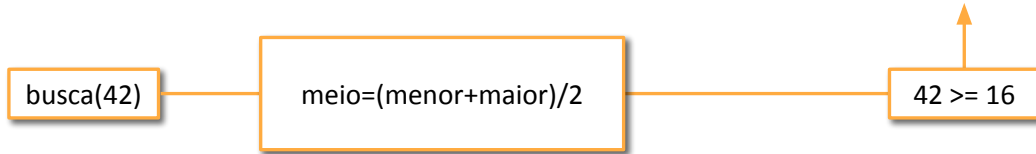
	menor			meio			maior
	↓			↓			↓
Posição	0	1	2	3	4	5	6
Valor	4	5	8	15	16	23	42



Busca binária

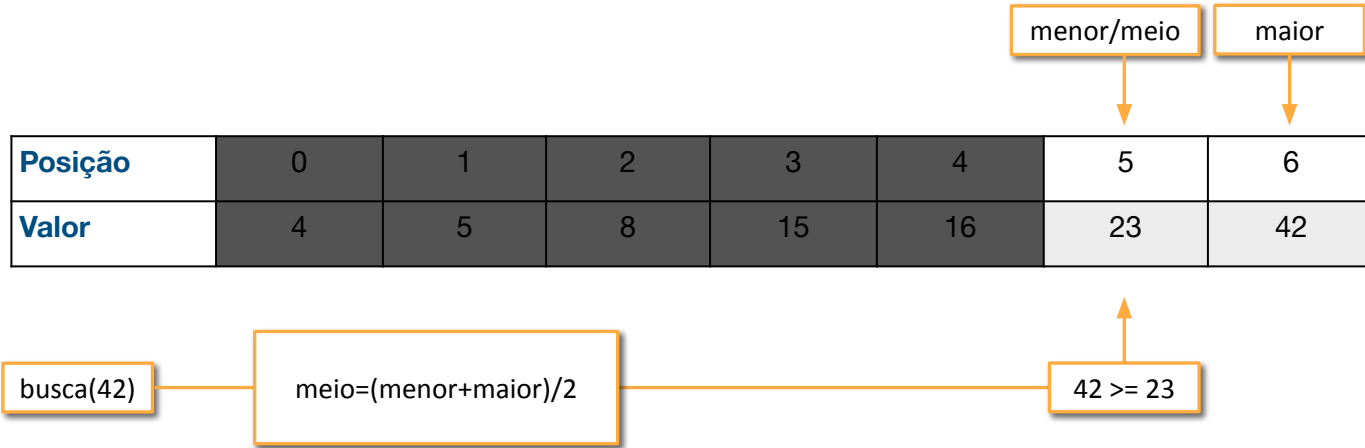
- Considere o conjunto de dados abaixo (ordenado):

				menor	meio		maior
				↓	↓		↓
Posição	0	1	2	3	4	5	6
Valor	4	5	8	15	16	23	42



Busca binária

- Considere o conjunto de dados abaixo (ordenado):



Busca binária

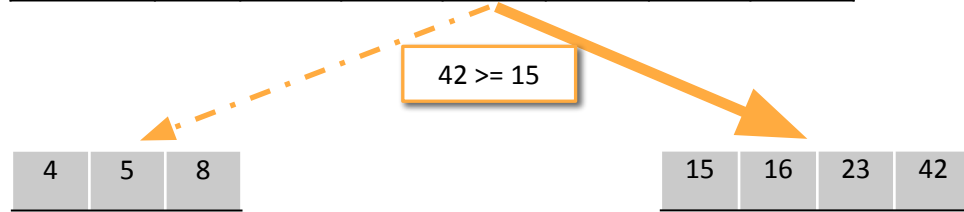
- Considere o conjunto de dados abaixo (ordenado):

Posição	0	1	2	3	4	5	6
Valor	4	5	8	15	16	23	42



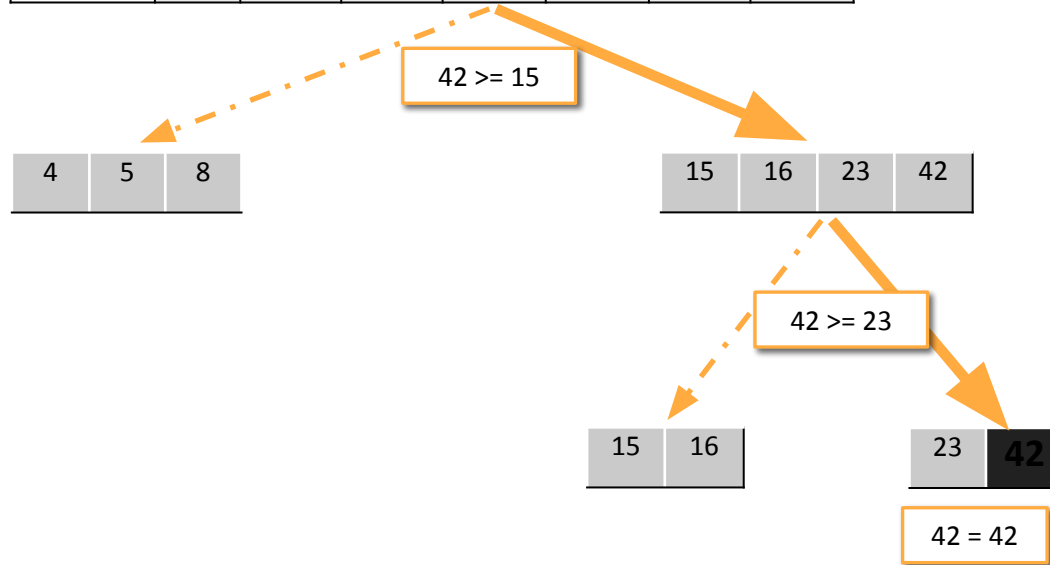
Resumindo

Posição	0	1	2	3	4	5	6
Valor	4	5	8	15	16	23	42



Resumindo

Posição	0	1	2	3	4	5	6
Valor	4	5	8	15	16	23	42



Problema

- Instância: (arr, x, l, r) , onde x é um valor e $v[l..r]$ é um vetor ordenado
- Resposta: $m \in [l-1..r]$ tal que $x \leq arr[i]$ para todo $i \in [m+1..r]$

```
int binarySearch(int *arr, int x, int l, int r) {  
    if(r < l) return -1;  
    int mid = l + ((r-l)/2);  
    if(arr[mid] == x)  
        return mid;  
    if(arr[mid] > x)  
        return binarySearch(arr, x, l, mid - 1);  
    return binarySearch(arr, x, mid + 1, r);  
}
```

Execução

i	1	2	3	4	5	6	7	8
v[i]	4	8	8	15	16	16	23	42

- Instância: (v,8,1,8)

```
int binarySearch(int *arr, int x, int l, int r) {  
    if(r < l) return -1;  
    int mid = l + ((r-1)/2);  
    if(arr[mid] == x)  
        return mid;  
    if(arr[mid] > x)  
        return binarySearch(arr, x, l, mid - 1);  
    return binarySearch(arr, x, mid + 1, r);  
}
```

Próxima aula

- TAD Dicionário