

Programação Estruturada

- Encontro 14 -

Sistemas de Informação
Prof.º Philippe Leal
philippeleal@yahoo.com.br

Agenda

- Algoritmos de Busca
 - ✓ Busca Binária

Busca Binária

- Fazer a busca em um vetor ordenado representa um ganho de tempo:
 - ✓ Podemos terminar a busca mais cedo se o elemento procurado for menor que o valor da posição atual do vetor;
- No entanto, a Busca Sequencial Ordenada é uma estratégia de busca extremamente simples, pois percorre todo o vetor no seu pior caso;
- A **Busca Binária** é muito mais **eficiente** do que a Busca Sequencial.

Busca Binária

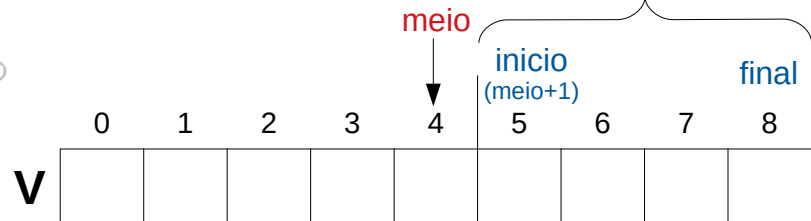
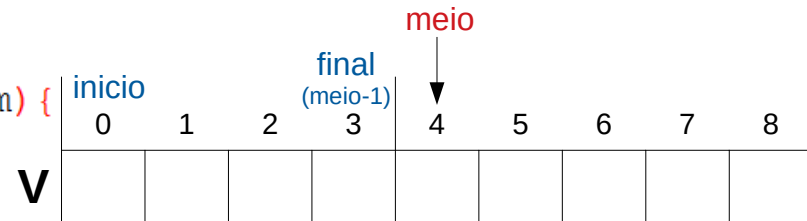
- Funcionamento:
 - ✓ É uma estratégia baseada na ideia de *dividir para conquistar*;
 - ✓ A cada passo, o algoritmo analisa o valor do meio do vetor:
 - Caso este valor seja igual ao elemento procurado, a busca termina;
 - Caso contrário, a busca continua na metade do vetor que condiz com o valor procurado.

Busca Binária

- Algoritmo:

- Exemplo da 1ª execução com vetor V de 9 elementos:

```
27
28 int buscaBinaria(int *V, int N, int elem){
29     int i, inicio, meio, final;
30     inicio = 0;
31     final = N-1;
32     while(inicio <= final){
33         meio = (inicio + final)/2;
34         if(elem < V[meio])
35             final = meio-1; //busca na metade da esquerda: do inicio até o final
36         else                (que agora é meio-1)
37             if(elem > V[meio])
38                 inicio = meio+1; //busca na metade da direita: do inicio (que
39                                 agora é meio+1) até o final
40             return meio;
41     }
42     return -1; //elemento não encontrado
43 }
```



- Exemplo:

	0	1	2	3	4	5	6	7	8	9
V	-8	-5	1	4	14	21	23	54	67	90

elem	4	Elemento procurado
------	---	--------------------

	0	1	2	3	4	5	6	7	8	9	
meio=4	-8	-5	1	4	14	21	23	54	67	90	Valor é menor: buscar no início

	0	1	2	3	4	5	6	7	8	9	
meio=1	-8	-5	1	4	14	21	23	54	67	90	Valor é maior: buscar no final

	0	1	2	3	4	5	6	7	8	9	
meio=2	-8	-5	1	4	14	21	23	54	67	90	Valor é maior: buscar no final

	0	1	2	3	4	5	6	7	8	9	
meio=3	-8	-5	1	4	14	21	23	54	67	90	Valor é igual: terminar a busca

Busca Binária

- Complexidade:
 - ✓ Considerando um vetor com **n** elementos, o tempo de execução é:
 - **$O(1)$** (melhor caso): o elemento procurado está no meio do vetor;
 - **$O(\log_2 n)$** (pior caso): o elemento não existe.

Busca Binária

- Complexidade:
 - ✓ Para se ter uma ideia da vantagem da Busca Binária, em um vetor contendo **$n = 1000$** elementos, no pior caso:
 - A Busca Sequencial irá executar **1000 comparações**;
 - A Busca Binária irá executar apenas **10 comparações**.

Referências

- BACKES, A. R. **Estrutura de Dados Descomplicada: em Linguagem C.** 1a Ed. Rio de Janeiro: Elsevier, 2016.
- Material do Professor André Backes (FACOM - UFU)