



UFC - UNIVERSIDADE FEDERAL DO CEARÁ

Centro de Ciências - Física e Matemática

Departamento de Computação

FP - Fundamentos de Programação - 2024.1

Pesquisa Binária

Professor: José Caminha Alencar Araripe Júnior
Fortaleza, 25 de agosto de 2024



Pesquisa Binária

A “**pesquisa binária**” é um procedimento utilizado para **localizar** um dado “valor” dentre os valores de um vetor. Os valores deste vetor devem, necessariamente, estarem ordenados. O algoritmo de “pesquisa binária” segue o paradigma de divisão e conquista.

Como o pressuposto inicial é de que **o vetor está ordenado**, então o procedimento é realizar sucessivas **divisões do vetor** (ou espaço de busca), sempre comparando o valor do elemento buscado (a chave) com o valor do **elemento no meio** do vetor (ou espaço de busca).

Se o valor do elemento do meio do vetor (ou espaço de busca) for igual ao valor do elemento buscado (chave), a busca termina com sucesso.

Caso contrário, se o valor do **elemento do meio** do vetor (ou espaço de busca) for **menor** que o valor do elemento buscado, então a busca continua na **metade posterior** do vetor. Ou, se o **elemento do meio** do vetor (ou espaço de busca) for **maior** que o valor do elemento buscado, então a busca continua na **metade anterior** do vetor.

Esse procedimento deve continuar até localizar o valor buscado ou concluir que ele não faz parte do vetor.



1. Pesquisa Binária: Exemplo.

Valor de Busca: **vb**.

Vetor										
	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
índices	1	2	3	4	5	6	7	8	9	10

Os elementos do **Vetor** já estão ordenados conforme os valores de cada índice.

Definir o valor do índice da metade do **Vetor**, o índice do elemento do meio do **Vetor**:

Índice Inicial = 1

Índice Final = N, no caso do exemplo N = 10;

Índice Médio = (Índice Inicial + Índice Final) / 2

Índice Médio = (1 + 10) / 2

Índice Médio = 5

Ver figura a seguir:

Vetor										
	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
Índices	1	2	3	4	5	6	7	8	9	10

Compara o valor de **vb** com o valor de **v5** que é o elemento da posição média do **Vetor**.

Se o valor procurado **vp** for maior que o valor do elemento **v5**, então a pesquisa continua na metade posterior do **Vetor**. A metade anterior não será mais observada nessa pesquisa. Ver figura a seguir:

	v6	v7	v8	v9	v10
índices	6	7	8	9	10

Definir o elemento do meio da metade posterior do **Vetor** (espaço de busca), figura acima, como o local da próxima busca:

Fazer:

Índice Inicial = Índice Médio + 1

Índice Inicial = 5 + 1

Índice Inicial = 6

Índice Médio = (Índice Inicial + Índice Final) / 2



UFC - Fundamentos de Programação - Pesquisa Binária - 2024.1

$$\text{Índice Médio} = (6 + 10) / 2$$

$$\text{Índice Média} = 8$$

Compara o valor de **vp** com o valor do elemento **v8** que é o elemento da posição média da parte posterior do **Vetor**.

Se forem iguais, **vp** e **v8**, então está localizado o **vp** e está na posição **8** do **Vetor**.

Se não, continua.

Se o valor de **vp** for menor que o valor de **v8**, então a pesquisa continua na metade anterior da parcela restante do **Vetor**. O restante o **Vetor** é desprezado para fins dessa pesquisa. Ver figura a seguir:

	v6	v7	v8
índices	6	7	8

Definir o elemento do meio da metade restante do **Vetor**, figura acima, como o local da próxima busca:

Fazer:

$$\text{Índice Final} = \text{Índice Médio}$$

$$\text{Índice Final} = 8$$

$$\text{Índice Médio} = (\text{Índice Inicial} + \text{Índice Final}) / 2$$

$$\text{Índice Médio} = (6 + 8) / 2$$

$$\text{Índice Média} = 7$$

Compara o valor de **vp** com o valor do elemento **v7** que é o elemento da posição média da parte posterior do **Vetor**.

Se forem iguais, **vp** e **v7**, então está localizado o **vp** e está na posição **7** do **Vetor**.

Se não, continua.....



2. Exemplo de Pesquisa Binária: Algoritmo

Elaborar um algoritmo para localizar em um VETOR de 1.000 elementos, um determinado valor VP (valor procurado) e informar a posição que esse valor se encontra.

A base de dados deve ter 1.000 **valores** numéricos gerados aleatoriamente no intervalo de 1 até 2000. A função GERA VR(limite) é que vai gerar os valores aleatórios.

Essa base de dados tem que ser ordenada (classificada) de forma crescente, do menor para o maior.

O algoritmo deve, também, determinar:

- (i) Quantas iterações do comando de repetição foram realizadas;
- (ii) Quantas comparações foram realizadas.

Ordenação é o processo de arranjar um conjunto de informações semelhantes numa ordem crescente ou decrescente. Para este trabalho a ordenação dos elementos do VETOR será realizada pela subrotina CLASSIFICA (VETOR[1000], N).



Início

```
Variável N, VP, I, LI, LF, IM, VETOR[1000], CONT: Inteira;  
Escreva ("Digite o número de elementos: ");  
Leia (N);  
Repita Para I = 1, N  
    VETOR[I] = GERARVR(2000);  
FimRepita;  
SUBROTINA CLASSIFICA (VETOR[1000], N);  
Escreva ("Digite o Valor de Pesquisa: ");  
Leia (VP);  
LI = 1;  
LF = N;  
CONT = 1;  
Repita Enquanto CONT = 1  
    IM = ( LI + LF ) / 2;  
    Se VETOR[IM] = VP  
        Então  
            Escreva ("Posição = ", IM, " Valor = ", VP);  
            CONT = 0;  
        SeNão  
            Se LI = LF  
                Então  
                    Escreva ("Valor Procurado Inexistente: ", VP);  
                    CONT = 0;  
                SeNão  
                    Se VP > VETOR[IM]  
                        Então  
                            LI = IM;  
                        SeNão  
                            LF = IM;  
                    FimSe;  
            FimSe;  
        FimSe;  
FimRepita;  
Fim.
```

X-X-X-X-X-X-X-X-X-X-X-X-X