



Busca

Prof. Aneu de Andrade Lopes

ICMC - USP - São Carlos

Algoritmo de Procura

- Algoritmo de Procura
 - O problema de procurar, pesquisar alguma informação numa tabela ou num catálogo é muito comum
- Exemplo:
 - procurar o telefone de uma pessoa no catálogo
 - procurar o nº da conta de um certo cliente
 - consultar o seu saldo em um terminal automático

Algoritmo de Procura

- A tarefa de “pesquisa”, “procura” ou “busca” é, como se pode imaginar, *uma das funções mais utilizadas*
- rotinas que a executem de uma forma *eficiente*
- *Eficiente*: uma rotina que faça a busca no menor tempo possível
- O TEMPO GASTO pesquisando dados em tabelas depende do TAMANHO da tabela.

Algoritmo de Procura

O tempo gasto
para se
descobrir um
telefone na
lista de São
Paulo

>

O tempo gasto
para se descobrir
um telefone na
lista de uma
cidade do interior
com 3000
habitantes

Algoritmo de Procura

- TEMPO GASTO pode variar muito dependendo do algoritmo utilizado
- Para os algoritmos de busca que se seguem vamos denotar por:
 - **Tab** - um vetor de n posições
 - **Dado** - elementos que devemos procurar em **Tab**
 - **Achou** - indica o sucesso ou falha na pesquisa
 - **Ind** - aponta para a posição do elemento encontrado

Parte que se repetirá...

```
Const N=100;
```

```
Type
```

```
    indice : integer;
```

```
    vetor : array [1..N] of integer;
```

```
( * ) Variáveis
```

```
    Tab : vetor;           {tabela de pesquisa}
```

```
    Ind : indice; {retorna a posição do elemento}
```

```
    Achou : boolean; {sucesso ou falha na busca}
```

```
    Dado : integer; {valor a ser procurado}
```

```
    I : integer;      {auxiliar}
```

Algoritmo 1 - Busca em Tabela

Alg1;

(*)

início

msg('Entre com os N valores da tabela');

for (I=0; I < N; I++)

ler(Tab[I]);

msg('Entre com o valor a ser procurado');

ler(Dado);

Achou=false;

for (I=0; I<N; I++) {

if Tab[I] = Dado

início

Achou=true;

Ind= I;

end;

}

if Achou

msg(Dado, 'se encontra na posição', Ind)

else

msg(Dado, 'não encontra na tabela');

end.

N comparações

(~ percorrer 1 dicionário todo)

Algoritmo 2 - Busca em tabela

- Para-se o processo de busca quando o dado for encontrado.

- 1º modo

```
I ← 1
Enquanto (Tab [I] != dado) and (I<=n) faça
    I = I + 1
```

- Este algoritmo não funciona pois

```
I ← N + 1
```

Leva a

Tab [N+1] - referência inválida

2º Modo (Com o uso da variável BOOLEAN)

Program Alg2;

(*)

início

msg ('Entre com os valores da tabela');

for (i=1;i<n;i++) ler (Tab[I]);

msg ('Entre com o valor a ser procurado');

ler (Dado);

Achou= false; Procura=

Ind= 0;

while Procura **do**

begin

Ind= Ind + 1;

if Ind > N

then Procura= false;

else Procura= Tab[Ind] <> Dado

end;

if Ind <= N

Achou = true;

if Achou

msg (Dado, 'se encontra na posição', Ind)

else msg (Dado, 'não encontra na tabela');

fim.

**Dado pode estar na 1ª posição
ou Dado pode estar na última
Na média: $N/2$ comparações**

Obs: 2 testes

- Procura = true
- Ind > N

Algoritmo 3: Busca com Sentinela

- Se tivermos certeza que o dado se encontra na tabela não precisaríamos fazer o teste
 $\text{Ind} > N$
- Única forma de ter certeza: **INSERIR** o Dado no final da tabela

```
Program Alg3; {Bem + simples}  
(*  
begin  
    msg ('Entre com os valores da tabela');  
    for (I=1;I<N-1;I++)  
        ler (Tab[I]);  
    msg ('Entre com o valor a ser procurado');  
    ler (Dado);  
    Achou= false;  
    Ind= 1;  
    Tab[N]= Dado;  
    while Tab[Ind] <> Dado do  
        Ind= Ind + 1;  
    Achou = Ind <> N  
    if Achou  
        msg (Dado,'se encontra na posição', Ind)  
    else msg (Dado,'não se encontra na tabela');  
end.
```

Algoritmo 4 - Busca binária (+ eficiente)

Dicionário - *Tarol*

- Abre-se o dicionário ao meio → letra J
- Abandonamos a 1ª metade
- Tomamos a metade a partir de J → letra P
- Abandonamos a 1ª metade
- Tomamos a metade a partir de P → letra S
(pág. 1318)
- Dividimos novamente, chegamos a palavra *Tomo* (pág. 1386)

∴ palavra está entre 1318 e 1386

Algoritmo 4 - Busca binária

- A cada passo dividimos a área de pesquisa à metade

- Caso o dicionário

$$11 \text{ pesquisas} = \log_2 1500$$

$$N \Rightarrow \log_2 N$$

$$32.000 \Rightarrow 15 \text{ comparações}$$

$$1500/2 \rightarrow 750$$

$$24/2 \rightarrow 12$$

$$750/2 \rightarrow 375$$

$$12/2 \rightarrow 6$$

$$375/2 \rightarrow 187,5$$

$$6/2 \rightarrow 3$$

$$188/2 \rightarrow 94$$

$$3/2 \rightarrow 1,5$$

$$94/2 \rightarrow 47$$

$$2/2 \rightarrow 1$$

$$47/2 \rightarrow 23,5$$

```

Program Alg4;
(*)
Inicio, Fim, Meio: Integer;
begin
    msg ('Entre com os valores da tabela');
    for (i=1;i<n;i++)
        ler (Tab[I]);
    msg ('Entre com o valor a ser procurado');
    ler (Dado);
    Achou= false;
    Inicio=1; Fim= N; Meio= (1+N) div 2;
    while (Dado <> Tab[Meio]) and (Inicio <> Fim)
        begin
            if Dado > Tab[Meio]
                Inicio = Meio + 1;
            else Fim = Meio;
            Meio= (Inicio + Fim) div 2
        end;
    Achou = Dado = Tab[Meio];
    if Achou
        msg (Dado, 'se encontra na posição', Meio)
    else msg (Dado, 'não encontra na tabela');
end.

```

Contém material preparado pela
Profª Roseli A. Francelin Romero

Fim

