



INSTITUTO SUPERIOR DE TRANSPORTES E COMUNICAÇÕES
DEPARTAMENTO DE TECNOLOGIAS DE INFORMAÇÃO E COMUNICAÇÃO
Licenciatura em Engenharia Informática e de Telecomunicações

Algoritmos e Estrutura de Dados

Introdução a algoritmos de pesquisa

Docente: Erick Mahanjane

Maputo, 04 de Abril de 2024

Conteúdo da aula

1. Introdução
2. Algoritmos de pesquisa
3. Pesquisa sequencial
4. Pesquisa binária
5. Exercícios

1. Introdução

- Durante as aulas, falamos de algoritmos que representam uma sequência de passos simples utilizados para resolver problemas específicos. No caso de problemas relacionados com base de dados, por exemplo, em que o utilizador precisa localizar determinado dado em um registo, devem ser aplicadas técnicas para pesquisar, de modo sistemático para encontrar o respectivo dado.
- Nesses termos, existem os algoritmos de pesquisa que permitem através de operações procurar a localização de um item numa colecção de dados, facilitando o processo de busca pelo utilizador.

2. Algoritmo de pesquisa

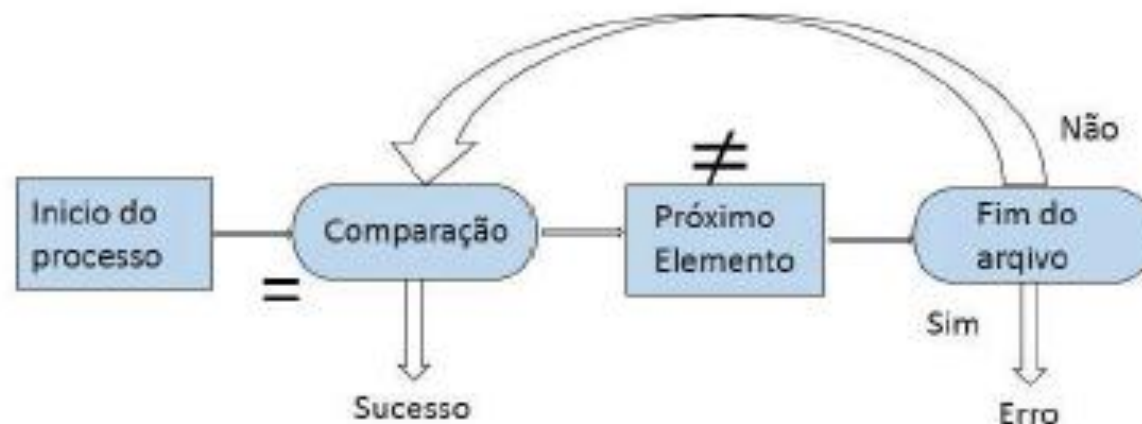
- Entende-se por pesquisa como sendo a operação que permite encontrar ou concluir que não existe um dado elemento num dado conjunto. A pesquisa de um elemento pode ser feita num conjunto de dados ordenados ou não.
- Existem dois tipos de realização de pesquisa, sequencial e binária.

3. Pesquisa sequencial

- Quando o conjunto de dados não está ordenado, o método a ser utilizado é o exaustivo, que consiste em percorrer sequencialmente todo o conjunto, isto é, desde o primeiro dado até ao último para se encontrar o elemento desejado ou, não encontrar caso não exista.
- Em termos de implementação este método é mais simples de pesquisa e consiste em varredura serial de uma tabela (vector, matriz ou arquivo), em que a chave de cada entrada é comparada até ser encontrada uma igual, ou ser atingido o final da tabela, caso a chave procurada não exista.

3. Pesquisa sequencial

- A função pode ser implementada de duas formas:
 - Retornar o próprio elemento encontrado; ou
 - Retornar o índice do elemento (no caso de ser um vector).



3. Pesquisa sequencial

- **Exemplo:** Considera-se a seguinte sequência: 1 2 3 4 5 6 7 8 9, pretende-se encontrar o número 4. Para se encontrar a solução deve-se começar a comparar do primeiro ao último elemento com 4, e se encontrar dar a solução.
- Embora seja dada o problema em termos ordenados, para a pesquisa sequencial não importa a ordem, sempre vai visitar os dados existentes de modo sequencial.

3. Pesquisa sequencial

Procurar um elemento X num vector V de tamanho Max

```
k ← -1 // denota que X ainda não foi encontrado em V
i ← 0 // índice dos elementos do vector V
enquanto (i < Max) e (k = -1) faça
se (V[i] = X) então
k ← i
senão
se (V[i] < Max) então
i ← i + 1
senão
```

```
k ← -2; // denota que X não está em V
se (k ≥ 0) then
X está na posição k
senão
X não está em V
```


4. Pesquisa binária

- Quando o conjunto de dados está ordenado, pode-se utilizar uma pesquisa binária, o que ajuda a localizar o dado mais rapidamente.
- Esse método é considerado muito superior e mais eficiente para encontrar o elemento procurado, através da utilização de uma abordagem dividir e conquistar.
- Ele primeiro verifica o elemento central, se esse elemento é maior que a chave, ele testa o elemento central da primeira metade, caso contrário, ele testa o elemento central da segunda metade. Esse procedimento é repetido até que o elemento seja encontrado ou que não haja mais elementos a testar.

4. Pesquisa binária

- **Exemplo:** Uma matriz contém os elementos 1 2 3 4 5 6 7 8 9 para encontrar o número 4, uma pesquisa binária testa o elemento médio nesse caso. Visto que 5 (elemento central) é maior que 4, a pesquisa continua com a primeira metade, isto é, 1 2 3 4. O elemento central agora é 3, que é menor que 4, então a primeira metade é descartada.
- A pesquisa continua com 4 e, nesse momento o elemento é encontrado.

4. Pesquisa binária

Procurar o elemento X no vector V de tamanho Max

```
início ← 0 //inicializar o início e o fim
fim ← tam - 1
k ← -1 // k recebe a posição de X
while ( (início ≤ fim) and (k = -1) ) do
meio ← (início + fim) / 2
if (X = V[meio]) then
k ← meio
else
```

```
if (X < V[meio]) então
fim ← meio - 1
else
início ← meio + 1
if (k ≥ 0) então X encontra-se em V na posição k
else
X não se encontra em V
```

5. Exercícios

1. Pretende-se encontrar o numero 34, como seria feita a pesquisas binárias e sequenciais para o seguinte conjunto de dados:

2	15	17	30	32	34	40	50	80	90	95	97	99	101	105
---	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----

2. Pretende-se encontrar o número 37, como seria feita a pesquisa binária:

1	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----

5. Exercícios

3. Pretende-se encontrar o número 37, como seriam feitas as pesquisas binárias e sequenciais:

2	15	17	30	32	34	40	50	80	90	95	97	99	101	105
---	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----

Nota: Para os três exercícios, determina o número de passos seguidos.

Bibliografia

- Ferrão, A. (n.d). *Estruturas de Dados e Algoritmos*. Universidade Virtual Africana. Link
- Parreira Júnior, W. (2015). *Apostila de algoritmos e estruturas de dados*.
http://www.waltenomartins.com.br/algdados_apostila_ord.pdf
- Ticongolo, I. (2021). Complexidade de Algoritmos–Aula 2 (Teórica). [Slides]

DÚVIDAS E/OU QUESTÕES

