

---

# ALGORITMOS E ESTRUTURAS DE DADOS III

Tutorial 15 (usa o compilador de linguagem C Dev-C++ versão 4.9.9.2)

Pesquisa em memória primária: pesquisa sequencial.

# 1 INTRODUÇÃO

Esta série de tutoriais sobre *Algoritmos e Estruturas de Dados III* foi escrita usando o **Microsoft Windows 7 Ultimate**, **Microsoft Office 2010**, **Bloodshed Dev-C++** versão 4.9.9.2 (pode ser baixado em <http://www.bloodshed.net>), **Code::Blocks** versão 10.05 (pode ser baixado em <http://www.codeblocks.org>) referências na internet e notas de aula do professor quando estudante. Ela cobre desde os algoritmos de ordenação, passando pela pesquisa em memória primária e culminando com a pesquisa em memória secundária.

Nós entendemos que você já conhece o compilador Dev-C++. No caso de você ainda não o conhecer, dê uma olhada nos tutoriais Dev-C++ 001 a 017, começando pelo [Tutorial Dev-C++ - 001 - Introdução](#).

Adotaremos o livro **Projeto de Algoritmos com Implementação em Pascal e C**, Editora Cengage Learning, de Nivio Ziviani, como livro-texto da disciplina. Nele você encontrará os métodos de ordenação que iremos estudar.

Se você seguiu todos os passos até aqui, está pronto para prosseguir com este tutorial.

## 2 PESQUISA EM MEMÓRIA PRIMÁRIA

### 2.1 BUSCA LINEAR

Busca linear (ou busca sequencial) para expressar um tipo de pesquisa em vetores ou listas de modo sequencial, isto é, elemento por elemento, de modo que a função do tempo em relação ao número de elementos é linear, ou seja, cresce proporcionalmente. Num vetor ordenado, essa não é a pesquisa mais eficiente, a pesquisa (ou busca) binária, por exemplo, é um tipo de pesquisa com o gráfico de tempo logarítimo.

### 2.2 ANÁLISE DE COMPLEXIDADE

No melhor caso, o elemento a ser buscado é encontrado logo na primeira tentativa da busca. No pior caso, o elemento a ser buscado encontra-se na última posição e são feitas  $N$  comparações, sendo  $N$  o número total de elementos. No caso

médio, o elemento é encontrado após  $N/2$  comparações. O algoritmo de busca linear é um algoritmo  $O(n)$ .

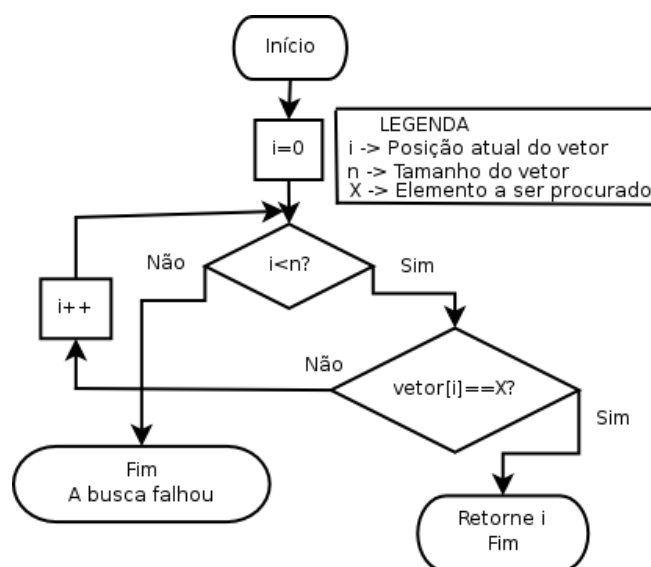
### 2.3 EXEMPLO DE CÓDIGO EM C

```
/**
 * Retorna -1 caso não encontre ou a posição, caso encontre.
 */
int procura(char vetor[], int tamanho, char itemProcurado) {
    int i;

    for (i = 0; i < tamanho; i++)
        if (vetor[i] == itemProcurado)
            return i;

    return -1;
}
```

### 2.4 FLUXOGRAMA



## 3 EXERCÍCIOS

1. Invente um vetor-exemplo de entrada e use o algoritmo de pesquisa sequencial para buscar um valor qualquer fornecido pelo usuário.
2. Considere uma matriz retangular. Modifique o algoritmo de busca sequencial para localizar um elemento da matriz. O algoritmo deve retornar a linha e a coluna onde o elemento foi encontrado.

## 4 TERMINAMOS

Terminamos por aqui. Mais em <http://flavioaf.blogspot.com>.

Corra para o próximo tutorial.