

Linguagem C: Ordenação e Busca –Laboratório–

Prof. Paulo R. S. L. Coelho
`paulo@facom.ufu.br`

Faculdade de Computação
Universidade Federal de Uberlândia

GEQ007



Organização

- 1 Introdução
- 2 Busca Binária
- 3 Exercícios



Organização

- 1 Introdução
- 2 Busca Binária
- 3 Exercícios



Introdução

- Uma das vantagens da manutenção de um vetor em ordem crescente é a facilidade de busca de informação nesse vetor.
- A busca pode ser realizada de duas maneiras:
 - 1 A mais elementar e menos eficiente, denominada **busca sequencial**, consiste em percorrer o vetor desde a posição 0 (zero) comparando cada elemento com o valor desejado, retornando sua posição caso seja encontrado, ou -1, caso contrário.
 - 2 A segunda maneira, mais inteligente, explora o fato de o vetor estar ordenado, realizando um busca otimizada conhecida como **busca binária**.



Busca Sequencial

- Busca sequencial:

```
//buscar numero x no vetor v de tamanho n

int buscaSequencial(int v[], int n, int x) {
    int i;
    for(i = 0; i < n; i++) {
        if(v[i] == x) {
            return i;
        }
    }
    return -1;
}
```



Organização

- 1 Introdução
- 2 Busca Binária**
- 3 Exercícios



Busca Binária I

- Se baseia na idéia comumente usada para encontrar um nome em uma lista telefônica.
- A busca binária parte do pressuposto de que o vetor está ordenado e realiza sucessivas divisões do espaço de busca comparando o elemento buscado (chave) com o elemento no meio do vetor.
- Se o elemento do meio do vetor for a chave, a busca termina com sucesso.
- Caso contrário, se o elemento do meio vier antes do elemento buscado, então a busca continua na metade posterior do vetor.
- E finalmente, se o elemento do meio vier depois da chave, a busca continua na metade anterior do vetor.

Fonte: wikipedia, verbete: busca binária



Busca Binária II

- EXEMPLO:

| | | | | | | | | |
|---|---|----|----|----|----|-----|-----|-----|
| 3 | 9 | 10 | 18 | 20 | 78 | 110 | 136 | 200 |
| e | | | | m | | | | d |

X = 136

| | | | | | | | | |
|---|---|----|----|----|----|-----|-----|-----|
| 3 | 9 | 10 | 18 | 20 | 78 | 110 | 136 | 200 |
| | | | | | e | m | | d |

| | | | | | | | | |
|---|---|----|----|----|----|-----|-----|-----|
| 3 | 9 | 10 | 18 | 20 | 78 | 110 | 136 | 200 |
| | | | | | | | e/m | d |

ACHOU!



Organização

- 1 Introdução
- 2 Busca Binária
- 3 Exercícios**



Exercícios

- 1 Implemente um programa em C que leia o tamanho desejado para um vetor de números reais e seus elementos. A leitura deve ser realizada por uma função `leVetor`, a qual deve receber como argumentos o vetor `v` e seu tamanho `n`.
- 2 Modifique a função `leVetor` criada anteriormente de maneira que cada elemento lido seja inserido em uma posição do vetor que o mantenha ordenado.
- 3 Utilizando o programa anterior, crie uma função `buscaBinaria` para realizar uma pesquisa no vetor ordenado lido na função `leVetor`.
A função `buscaBinaria` deve receber como argumentos o vetor `v`, seu tamanho `n` e o valor a ser buscado (`x`). Ela deve retornar a posição em que `x` foi localizado ou `-1` caso não seja encontrado.
A função principal deve solicitar que o usuário informe o valor que deseja pesquisar no vetor e imprimir a posição em que se encontra o elemento ou uma mensagem de erro caso não seja encontrado.



Respostas I

```
#include <stdio.h>
#include <stdlib.h>

void leVetor(int v[], int n) {
    int tamanho = 0, i, j, valor;
    printf("Entre os %d elementos do vetor:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &valor);
        j = tamanho - 1;
        while (j >= 0 && valor < v[j]) {
            v[j+1] = v[j];    // empurra elementos
            j--;              // maiores para frente
        }
        v[j+1] = valor;      // insere na pos. correta
        tamanho++;
    }
}

int buscaBinaria(int v[], int n, int x) {
    int esquerda = 0, direita = n-1, meio;
    while (esquerda <= direita) {
        meio = (esquerda + direita) / 2;
        if(x == v[meio])    // se achou, retorna posicao
            return meio;
        if(x > v[meio])    // metade da direita
            esquerda = meio + 1;
        else                // metade da esquerda
            direita = meio - 1;
    }
    return -1; // se nao achou
}
```

Respostas II

```
}  
  
int main() {  
    int v[200], n, i, x, pos_x;  
    char continuar = 's';  
    printf("Entre tamanho desejado do vetor: ");  
    scanf("%d", &n);  
    leVetor(v, n);  
    printf("\n\nVetor ordenado:\n");  
    for(i = 0; i < n; i++) {  
        printf("%d\t", v[i]);  
    }  
    printf("\n\n");  
    while (continuar != 'n') {  
        printf("Entre numero que deseja buscar no vetor: ");  
        scanf("%d", &x);  
        pos_x = buscaBinaria(v, n, x);  
        if(pos_x == -1) {  
            printf("Numero nao encontrado.\n");  
        } else {  
            printf("Numero %d encontrado na posicao %d.\n", x, pos_x);  
        }  
        printf("Deseja continuar (s/n)? ");  
        fflush(stdin);  
        scanf("%c", &continuar);  
    }  
    return 0;  
}
```