



Universidade Federal Rural de Pernambuco
(UFRPE)
- Unidade Acadêmica de Garanhuns-

Busca Linear e Binária

Estruturas de Dados

Prof. Priscilla Kelly Machado Vieira

Apresentação do Capítulo

2

- Introdução
- Busca em vetor
 - Busca Linear
 - Busca Binária

Introdução

- Buscar um elemento em um determinado conjunto de dados
 - Um programa de controle de estoque pode buscar, dado um código ou nome, a descrição e as características de um determinado produto
 - Método eficiente em grandes volumes de produtos
 - Tem que ser viável

Introdução

- Considerar que os dados estão armazenados em vetores
 - Dado um vetor com n elementos, desejamos saber se um determinado elemento está ou não no vetor
- Futuramente (AED 2) serão consideradas outras estruturas de dados

Busca Linear

5

- Percorre o vetor elemento a elemento, verificando se o elemento em questão existe ou não no vetor

```
int busca (int n, int* vet, int elem) {  
    int i;  
    for (i=0; i<n; i++) {  
        if (elem == vet[i]) {  
            return i;  
        }  
    }  
    return -1;  
}
```

Busca Linear

6

- Se o número de elementos for muito grande?
 - Pode percorrer todos os elementos (n comparações)
 - Em média $n/2$ comparações
 - Melhor caso 1 comparação
- Seria possível melhorar o algoritmo?

Ordenados!

Busca Linear

- Se os elementos estão ordenados

```
int busca_ord (int n, int* vet, int elem) {  
    int i;  
    for (i=0; i<n; i++) {  
        if (elem == vet[i]) {  
            return i;  
        } else if (elem < vet[i])  
            return -1;  
        }  
    return -1;  
}
```

Busca Linear

8

- Se os elementos estão ordenados
 - Um desempenho ligeiramente melhor que o interior
- Busca Binária

Busca Binária

- Testa o elemento que buscamos com o elemento central do vetor
- Se o elemento for menor que o elemento central, sabemos que se o elemento está no vetor, ele estará na primeira parte do vetor
- Se não, estará na segunda parte do vetor

Busca Binária

10

```
int busca_bin (int n, int* vet, int elem) {  
    int ini = 0; int fim = n-1; int meio;  
    while (ini <= fim) {  
        meio = (ini+fim)/2;  
        if (elem < vet [meio]) {  
            fim = meio-1; //ajusta fim  
        } else if (elem > vet [meio]) {  
            ini = meio + 1; //ajusta início  
        } else {  
            return meio; //elemento encontrado  
        }  
    }  
    return -1; //não encontrou  
}
```

Busca Binária

- Desempenho superior ao anterior
- A cada iteração descartamos metade do vetor
- Se os dados estão armazenados em uma lista encadeada, só temos a alternativa de implementar um algoritmo de busca linear, mesmo se os elementos estiverem ordenados

Exercício

12

- Implemente a busca binária com um código recursivo