Algoritmos e Estruturas de Dados

Vetores: Pesquisa

Prof. Mauri Ferrandin

Pesquisa em vetores

Comumente grandes quantidades de dados são armazenados em vetores.

- V[0] = 44
- V[2] = 230
- V[3] = 54
- ...
- V[998] = 6000
- V[999] = 37

Como verificar se este vetor contém um determinado valor?

Pesquisa em vetores

Problema:

- Verificar se um valor existe no vetor
  A. no caso de existir, indicar sua posição.

- No caso de vetores com valores repetidos:
  A. indicar a posição da primeira ocorrência
  B. indicar a posição da última ocorrência
  C. indicar a posição de uma ocorrência qualquer

Pesquisa

Duas técnicas para pesquisa em vetores

- Pesquisa Linear/Sequencial
- Pesquisa Binária

Pesquisa Sequencial

Uma solução possível consiste em percorrer sequencialmente todas as posições do vetor.

Para cada posição *i*, compara-se *vetor[i]* com o *valor* desejado.

- Se forem iguais diz–se que o *valor* existe
- Se chegarmos ao fim do vetor sem sucesso diz-se que o *valor* não existe

Pesquisa Sequencial

1º Passo: Inicialização

```
int i = 0;
int encontrado = 0; /* falso */
```

Pesquisa Sequencial

2° Passo: Pesquisa

```
while (i < tamanho && !encontrado){
   if(vetor[i]== valor){
        encontrado = 1; /* Verdadeiro */
   }
   i++;
}
```

Pesquisa Sequencial

3° Passo: tratamento dos resultados

```
if (encontrado){
   printf("Valor %d está na posição %d\n",
        vetor[i],i);
}else{
    printf("Valor %d nao encontrado\n",
valor);
}
```

Pesquisa Sequencial

Quanto tempo a busca sequencial demora para executar? Em outras palavras, quantas vezes a comparação valor == vetor[i] é executada?

- Caso valor não esteja presente no vetor:
  - n vezes.
- Caso valor esteja presente no vetor:
  - Melhor caso: 1 vez (valor está na primeira posição).
  - Pior caso: n vezes (valor está na ultima posição).
  - Caso médio: n/2 vezes

Implementação da Pesquisa Sequencial em C

```c
/* Procura um valor inteiro (x) num vetor (v). Retorna o índice
da sua primeira ocorrência, se encontrar; senão, retorna -1. */

int pesquisaSequencial(int *v, int tamanho, int x){
   int i;
   for (i = 0; i < tamanho; i++)
       if (v[i] == x)
           return i; // encontrou

   return -1; // não encontrou
}
```

Pesquisa em Vetores Ordenados

Supondo que o vetor inicial está ordenado em ordem crescente, é possível resolver o problema de modo mais eficiente?

- Caso o vetor esteja ordenado:
  dados i e j, se i < j, então: A[i] ≤ A[j]
- Portanto, comparando um determinado elemento com o elemento procurado, saberemos:
  - se o elemento procurado é o elemento comparado;
  - se ele está antes do elemento comparado ou;
  - se está depois.

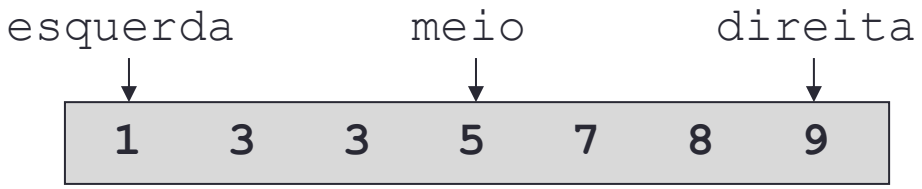
Pesquisa em vetores ordenados

- Se fizermos isso sempre com o elemento do meio da lista, a cada comparação dividiremos o vetor em dois, reduzindo nosso tempo de busca.
- Se em um determinado momento o vetor, após sucessivas divisões, tiver tamanho zero, então o elemento não está no vetor.

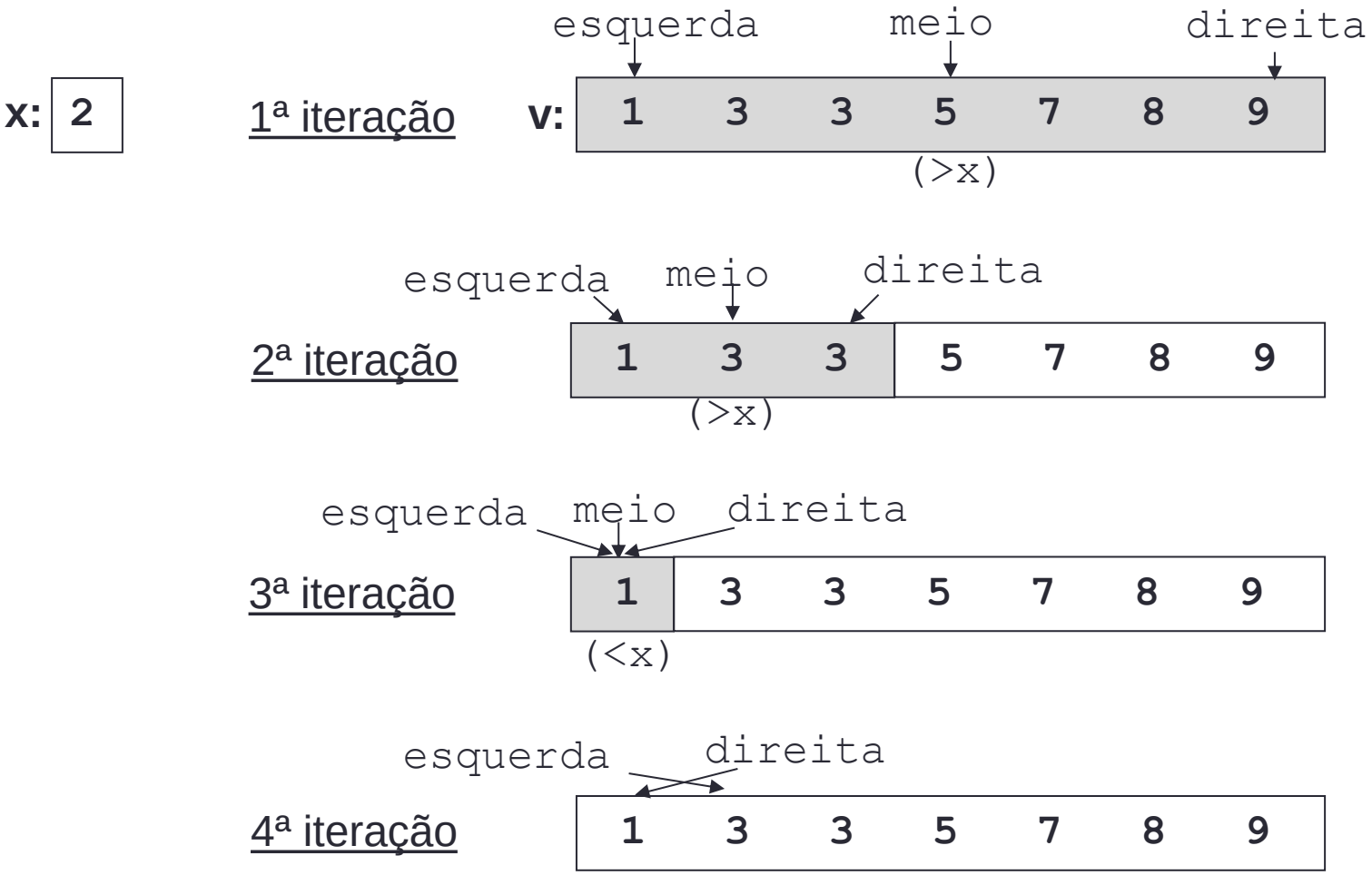
Pesquisa Binária

Algoritmo:

- Comparar o valor que se encontra a meio do vetor com o valor procurado:
  A. = valor procurado → está encontrado
  B. >valor procurado → continuar a procurar (do mesmo modo) no sub-vetor à esquerda da posição inspecionada
  C. <valor procurado → continuar a procurar (do mesmo modo) no sub-vetor à direita da posição inspecionada.
- Se o vetor a inspecionar se reduzir a um vetor vazio, conclui-se que o valor procurado não existe.



Exemplo de Pesquisa Binária



vetor a inspecionar vazio ⇒ o valor 2 não existe no vetor inicial !

Pesquisa Binária

1º Passo: Inicialização

```
int direita, esquerda, meio;
int encontrado = 0; /* falso */

esquerda = 0;
direita = tamanho - 1;
```

Pesquisa Binária

2° Passo: Pesquisa

```
while(esquerda<=direita && !encontrado){
    meio=(direita+esquerda)/2;
    if (vetor[meio] == valor)
        encontrado = 1; /*Verdadeiro*/
    else if (valor < vetor[meio])
        direita = meio - 1;
    else
        esquerda = meio + 1;
}
```

Pesquisa Binária

3° Passo: tratamento dos resultados

```
if(encontrado){
    printf ("Valor %d encontrado na posicao
%d\n",vetor[meio], meio);
}else{
    printf ("Valor %d nao encontrado\n",
        valor);
}
```

Implementação da Pesquisa Binária em C

```c
/* Procura um valor inteiro (x) num vetor (v) previamente
ordenado. Retorna o índice de uma ocorrência, se encontrar;
senão, retorna -1. */

int pesquisaBinaria(int *v, int tamanho, int x)
{
    int left = 0, right = tamanho - 1;
    while (left <= right)
    {
        int middle = (left + right) / 2;
        if (x == v[middle])
            return middle; // encontrou
        else if (x < v[middle])
            right = middle - 1;
        else left = middle + 1;
    }
    return -1;  // não encontrou
}
```