UNINOVE

Estrutura de Dados

Prof. Mailson

Email: mailson.silva@uninove.br

UNINOVE
Universidade Nove de Julho

Pesquisa Sequencial

Lembram-se dessa pergunta?

- Por que ordenar?
  - ⇨ “Ordenar os dados também pode ser uma etapa preliminar para procurá-los, que é muito mais rápido do que um procura linear”;

Pesquisa Sequencial

- Uma forma simples de procurar consiste em:
  - Começar no inicio do vetor; comparar sucessivamente o dado procurado com o dado do vetor; repetir o processo até encontrar o dado ou atingir o fim do vetor.
- Como percorremos sequencialmente o vetor, a este algoritmo chamamos pesquisa seqüencial.
- Exemplo: Pretende-se saber qual a posição que o valor 75 (se existir) ocupa num vetor de números.
  - Comparo 75 com 10 => São diferentes => Incremento a posição.
  - Comparo 75 com 20 => São diferentes => Incremento a posição.
  - Comparo 75 com 75 => São iguais => 75 ocupa a 3ª posição no vetor.

V [0]	V [1]	V [2]	V [3]	V [4]	V [5]
10	20	75	9	4	3

- A busca será mais rápida se o vetor estiver com seus elementos ordenados.
- O tempo de busca pode ser reduzido significantemente devido a relação entre os elementos da lista.
- Será feita a busca pelo valor V em um conjunto de N elementos ordenados.
- O procedimento retorna o valor da posição no vetor se encontrar V e -1 (um negativo) caso não encontre.

Pesquisa Sequencial

- Buscar o valor 5 no vetor;
- Vetor desordenado:

V [0]	V [1]	V [2]	V [3]	V [4]	V [5]
10	20	75	9	4	3

- Vetor Ordenado:

V [0]	V [1]	V [2]	V [3]	V [4]	V [5]
3	4	9	10	20	75

Pesquisa Sequencial

Vetor Desordenado

```
int pesqseq(int v[], int busca)
{
   int i;
   for (i = 0; i <= MAX; i++)
        if (v[i] == busca)  return i;
   return -1;
}
```

Vetor Ordenado

```
int pesqseq(int v[], int busca)
{
     int i;
     for (i = 0; i <= MAX; i++)
     {
          if (v[i] == busca)
               return i;
          else
               if (v[i]>busca)
                    return -1;
     }
     return -1;
}
```

Pesquisa Sequencial

```
// função para ordenar o vetor

void bsort(char vet[ ], int t)
{
   int i, j;
   char k;
   for (i = 0; i < t - 1; i++)
   {
      for (j = 0; j < t-( i + 1); j++)
      {
         if (vet[ j ]>vet[ j + 1 ])
         {
            k = vet[ j ] ;
            vet[ j ] = vet[ j + 1 ];
            vet[ j + 1 ] = k ;
         }
      }
   }
}
```

```
// função de busca no vetor

int pesqseq(char v[ ], char busca)
{
   int i;
   for (i = 0; i <= MAX; i++)
   {
      if (v[ i ] == busca)
         return i;
      else
         if (v[ i ]>busca)
            return -1;
   }
   return -1;
}
```

Exercícios

- 1 – Desenvolva um programa que receba 20 números inteiros quaisquer digitados pelo usuário. Ao final peça para ele digitar outro número e usando a função “Pesquisa Sequencial Ordenada” informe em que posição do vetor o número se encontra e ao final mostre o vetor ordenado;

Exercícios

- 2 – Desenvolva um programa que receba 20 nomes quaisquer digitados pelo usuário. Ao final peça para ele digitar outro nome e usando a função “Pesquisa Sequencial Ordenada” informe em que posição do vetor o número se encontra e ao final mostre o vetor ordenado; É necessário adaptar o algoritmo de busca para trabalhar com string;