

```

watershed.c > ...
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include "../utils/image
5
6  /*-----
7   * Fila de prioridade
8   *-----
9  typedef struct no *ptno;
10 typedef struct no
11 {
12     int i, j;
13     ptno next;
14 } no;
15
16 /*-----
17 * init *Q new in
18 * [a:] [b:] + [c:] -> [a:]
19 * ^
20 *-----
21 void InsQ(ptno *Q, int i, int j)
22 {
23     ptno new = malloc(sizeof(no))
24     new->i = i;
25     new->j = j;
26     if (!(*Q))
27         new->next = new;
28 }

```

Ordenação

Luiz Eduardo da Silva

Algoritmos e Estrutura de Dados I

Ciência da Computação

UNIFAL-MG

Agenda

1 Ordenação

2 Métodos

- A troca
- Ordenação por troca
- Ordenação por seleção
- Ordenação por inserção

Agenda

1 Ordenação

2 Métodos

Introdução

- Uma base de dados teria pouca utilidade se fosse muito difícil encontrar os dados armazenados.
- As tarefas mais comuns executadas em sistemas computacionais são a classificação e a pesquisa de dados armazenados.
- A classificação é importante para que possamos obter respostas do tipo: "Qual foi o produto mais vendido no último mês?".

Definição

Definição

Classificar ou ordenar um conjunto de dados consiste em receber como entrada uma sequência de n valores $\langle a_1, a_2, \dots, a_n \rangle$ e produzir como resultado uma permutação (reordenamento) dos valores de entrada $\langle a'_1, a'_2, \dots, a'_n \rangle$ de tal forma que $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

Agenda

1 Ordenação

2 Métodos

- A troca
- Ordenação por troca
- Ordenação por seleção
- Ordenação por inserção

Métodos

- São três as classes gerais para classificação interna de dados: ordenação por troca, por seleção e por inserção.
- Suponha o problema de ordenação das cartas do baralho:

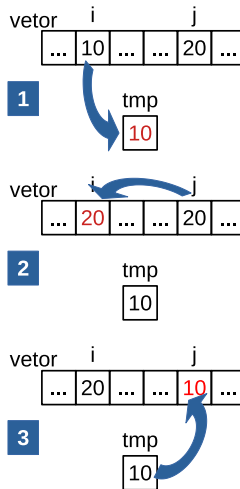
Por troca: Espalhe as cartas numa mesa voltadas para cima e então troque as cartas de ordem até que todo o baralho esteja ordenado.

Por seleção: Espalhe as cartas na mesa, selecione a carta de menor valor, retire-a do baralho e segure-a na sua mão. Este processo continua até que todas as cartas estejam na sua mão.

Por inserção: Segure todas as cartas na sua mão. Ponha uma carta por vez na mesa, sempre inserindo na posição correta. O maço estará ordenado quando não restarem mais cartas em sua mão.

Swap

- 1 $\text{tmp} = \text{vetor}[i];$
- 2 $\text{vetor}[i] = \text{vetor}[j];$
- 3 $\text{vetor}[j] = \text{tmp};$



Ordenação por troca

```

1  int troca , i , salva ;
2  do {
3      troca = FALSE;
4      for (i = 0; i < n-1; i++)
5          {
6              if (v[i] > v[i+1])
7                  {
8                      salva = v[i];
9                      v[i] = v[i+1];
10                     v[i+1] = salva;
11                     troca = TRUE;
12                 }
13         }
14     } while (troca);

```

Ordenação por seleção

```

1  int i, j, menor, salva;
2  for (i = 0; i < n-1; i++)
3  {
4      menor = i;
5      for (j = i+1; j < n; j++)
6          if (v[j] < v[menor])
7              menor = j;
8      salva = v[i];
9      v[i] = v[menor];
10     v[menor] = salva;
11 }

```

Ordenação por inserção

```

1  int   i, j, x;
2  i = 1;
3  while (i < n)
4  {
5      x = v[i];
6      j = i - 1;
7      while (j >= 0 && x < v[j])
8      {
9          v[j+1] = v[j];
10         j--;
11     }
12     v[j+1] = x;
13     i++;
14 }
```
