Ordenação de dados

Prof. Leandro O. Freitas
leandro@politecnico.ufsm.br

Ordenação de dados

Sistemas para Internet
UFSM

- Processo de arranjar/organizar um conjunto de dados em determinada ordem (crescente ou decrescente, ou ainda, outra pré-estabelecida pelo desenvolvedor);

- Coloca os elementos de uma sequência em ordem (ordenação parcial ou total);
  - Algoritmos de ordem numérica ou lexicográfica.

- Exemplo: sendo ***L*** uma lista ordenada de ***n*** elementos, então temos:
  - $L_1 <= L_2 <= L_3 <= ... <= L_n$

- Permite acessar seus dados com mais eficiência.

Ordenação de dados

Sistemas para Internet
UFSM

- *Bubble sort*
- Ordenação por seleção (*selection sort*)
- Ordenação por inserção (*insertion sort*)
- *Quick sort*
- ...

Bubble Sort

Sistemas para Internet
UFSM

- Processo de ordenação mais conhecido, devido sua simplicidade;
- Faz a comparação e troca de elementos **adjacentes** para ordenação de dados;
- Quando aplicado a um grande número de dados, possui um dos piores desempenhos.

Bubble Sort

Sistemas para Internet
UFSM

- Percorre a lista da esquerda para direita, comparando elementos vizinhos e os trocando de lugar se estiverem fora de ordem.
  - Exemplo: Colocar os 5 elementos de um conjunto L em ordem crescente.

Primeira varredura	**L1**	**L2**	**L3**	**L4**	**L5**
Compara L1 com L2	13	9	2	0	6
Compara L2 com L3	9	13	2	0	6
Compara L3 com L4	9	2	13	0	6
Compara L4 com L5	9	2	0	13	6
	9	2	0	6	13

- Ao final da primeira varredura apenas o maior elemento encontra-se em sua posição definitiva. Este processo deve ser repetido até que todos os elementos estejam em ordem.

Bubble Sort

- Na segunda varredura, o segundo maior elemento irá ocupar sua posição definitiva.

Segunda varredura	**L1**	**L2**	**L3**	**L4**	**L5**
Compara L1 com L2	9	2	0	6	13
Compara L2 com L3	2	9	0	6	13
Compara L3 com L4	2	0	9	6	13
Compara L4 com L5	2	0	6	9	13

- Quantas varreduras são necessárias para ordenar todos os elementos?
  - Resposta: total de elementos - 1.
  - Não recomendado para volume grande de dados devido ao tempo de execução.

Selection sort (seleção)

Sistemas para Internet
UFSM

- Consiste em identificar o menor elemento da lista;
- Trocar esse elemento com o da primeira posição;
- Depois, identificar o segundo menor elemento e trocá-lo com o da segunda posição;
- Repetir esse processo para os demais.

	L1	**L2**	**L3**	**L4**	**L5**
Troca L4 com L1	13	9	2	0	6
Troca L3 com L2	0	9	2	13	6
Troca L5 com L3	0	2	9	13	6
Troca L5 com L4	0	2	6	13	9
	0	2	6	9	13

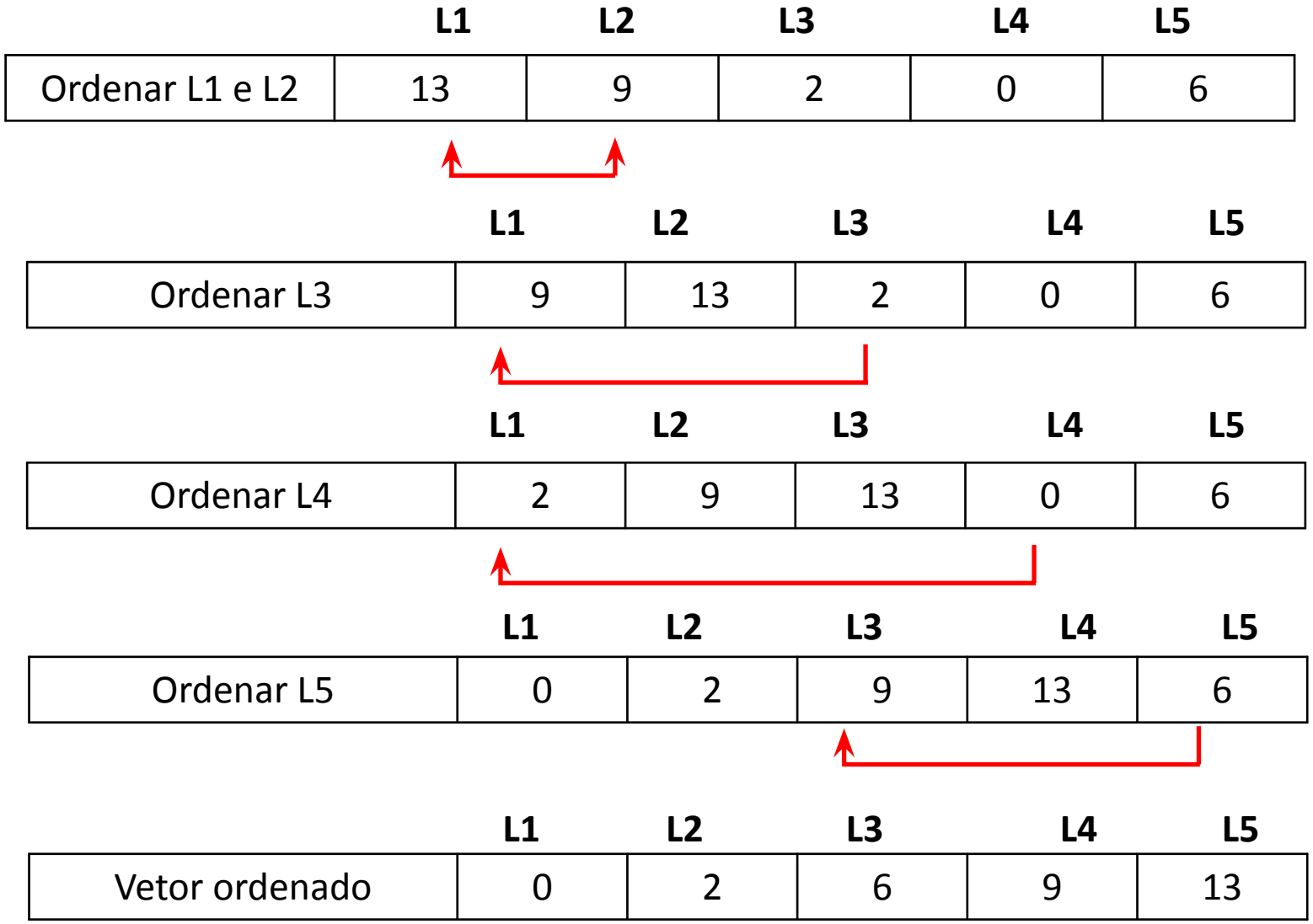
Insertion sort (inserção)

Sistemas para Internet
UFSM

- Consiste em:
  1. Colocar em ordem crescente os dois primeiros elementos;
  2. O terceiro elemento é inserido na posição correta, de acordo com os **dois** primeiros, ou seja, antes do primeiro, entre os dois, ou depois do segundo;
  3. Inserir o quarto elemento na posição correta, de acordo com os **três** primeiros;
  4. Repetir este processo até que todos estejam ordenados.

Insertion sort (inserção)

Sistemas para Internet
UFSM



Exercícios de aprendizagem

- Em um papel, crie um conjunto de 5 valores inteiros e simule a ordenação etapa por etapa através dos três métodos vistos até aqui. O resultado deve englobar todas as análises e trocas realizadas;

- Depois, desenvolva funções para ordenar os valores de um conjunto de 5 elementos. Ao final, exibir os valores ordenados. Implementar os algoritmos de *bubble sort, selection sort e insertion sort.*

Bubble Sort (solução)

Sistemas para Internet
UFSM

- Ordenar os elementos de um vetor de 5 posições.

```
11 void bubble(int lista[], int qtd) {
12     int i, j, aux;
13     for(i=0; i<qtd; i++) {
14         for(j=0; j<qtd-1; j++) {
15             if (lista[j] > lista[j+1]){
16                 aux = lista[j];
17                 lista[j] = lista[j+1];
18                 lista[j+1] = aux;
19             }
20         }
21     }
22 }
```

Ordenação por seleção (solução)

Sistemas para Internet
UFSM

- Ordenar os elementos de um vetor de 5 posições.

```
24 void selection(int lista[], int qtd) {
25     int i, j, aux, min;
26     for(i=0; i<qtd-1; i++) {
27         min = i;
28         for(j=i+1; j<qtd; j++) {
29             if (lista[j] < lista[min]){
30                 min = j;
31             }
32         }
33         aux = lista[i];
34         lista[i] = lista[min];
35         lista[min] = aux;
36     }
37 }
```

Ordenação por inserção (solução)

Sistemas para Internet
UFSM

- Ordenar os elementos de um vetor de 5 posições.

```
39 void insertion(int lista[], int qtd) {
40     int i, j, aux;
41     for(i=1; i<qtd; i++) {
42         aux = lista[i];
43         for(j=i-1; j>=0 && aux<lista[j]; j--){
44             lista[j+1] = lista[j];
45         }
46         lista[j+1] = aux;
47     }
48 }
```

Exercício com listas encadeadas

- Utilizando os conceitos de *Bubble sort, selection sort e insertion sort,* desenvolva um algoritmo com funções que ordenem uma lista encadeada de 5 elementos inteiros.