

Aula 9: Ordenação de Listas Lineares

- ⇒ O Problema de ordenação
- ⇒ Ordenação por seleção
- ⇒ Ordenação pelo método da bolha

O Problema de Ordenação

➡ "Dada uma Lista Linear L com n elementos, deseja-se ordená-los crescentemente através de trocas de suas posições originais."

▢ Exemplo:

Lista L :

10	5	9	-2	13	-8	4	0	1	-5	7	6
----	---	---	----	----	----	---	---	---	----	---	---

Lista L , ordenada crescentemente:

-8	-5	-2	0	1	4	5	6	7	9	10	13
----	----	----	---	---	---	---	---	---	---	----	----

Ordenação por Seleção

- ➡ Idéia: Encontrar o menor elemento da lista e trocá-lo de posição com o primeiro. Repetir o processo a partir do segundo elemento, e assim sucessivamente.
- ➡ Exemplo: Observe as sucessivas etapas da ordenação por seleção na lista a seguir.

Ordenação por Seleção

- A parte hachurada é a parte já ordenada.
- O elemento marcado com * é o menor elemento da parte não ordenada:

10	5	9	-2	13	-8*	4	-5
-8	5	9	-2	13	10	4	-5*
-8	-5	9	-2*	13	10	4	5
-8	-5	-2	9	13	10	4*	5
-8	-5	-2	4	13	10	9	5*
-8	-5	-2	4	5	10	9*	13
-8	-5	-2	4	5	9	10*	13
-8	-5	-2	4	5	9	10	13*
-8	-5	-2	4	5	9	10	13*

[Animar](#)
[Voltar](#)

Algoritmo de Ordenação por Seleção

➡ Entrada : Vetor V com n elementos, desordenados

```

para i := 1 ... n faça      % considerar a lista da posição i em diante
    MENOR := i                % para cada valor de i
    para j := i + 1 ... n faça    % procurar o menor
        (*) .... se    V[ j ] < V[MENOR]
            então    MENOR := j
    fim-para
    AUX := V[ i ]              % fazer a troca
    V[ i ] := V[MENOR]
    V[MENOR] := AUX
fim-para

```

➡ Complexidade : Número de comparações na linha (*)
 Portanto, a complexidade é $\theta(n^2)$

Exercício

- ➡ Desenhar as trocas de elementos ao ordenar a lista abaixo utilizando ordenação por seleção.
Quantas trocas de elementos foram efetuadas?

20	13	17	-9	4	8	2	-1	-5	0	-11	6
----	----	----	----	---	---	---	----	----	---	-----	---

Tempo: 5 minutos

Solução



São efetuadas 12 trocas

(observe que a última troca é de um elemento com ele mesmo)

20	13	17	-9	4	8	2	-1	-5	0	-11*	6
-11	13	17	-9*	4	8	2	-1	-5	0	20	6
-11	-9	17	13	4	8	2	-1	-5*	0	20	6
-11	-9	-5	13	4	8	2	-1*	17	0	20	6
-11	-9	-5	-1	4	8	2	13	17	0*	20	6
-11	-9	-5	-1	0	8	2*	13	17	4	20	6
-11	-9	-5	-1	0	2	8	13	17	4*	20	6
-11	-9	-5	-1	0	2	4	13	17	8	20	6*
-11	-9	-5	-1	0	2	4	6	17	8*	20	13
-11	-9	-5	-1	0	2	4	6	8	17	20	13*
-11	-9	-5	-1	0	2	4	6	8	13	20	17*
-11	-9	-5	-1	0	2	4	6	8	13	17	20*
-11	-9	-5	-1	0	2	4	6	8	13	17	20

Ordenação pelo Método da Bolha

➡ Idéia: os elementos mais leves "sobem". Os elementos mais pesados "descem e vão para o fundo". Pode-se comparar a subida de um elemento como uma bolha, que tenta chegar a superfície.

- ➡ A **bolha** é marcada com " * " ;
- ➡ A extremidade esquerda da lista corresponde à "superfície";
- ➡ A extremidade direita da lista corresponde ao "fundo";
- ➡ A bolha vai subindo até encontrar sua posição;
 - ➡ Observe a próxima transparência

Ordenação pelo Método da Bolha

"Superfície" 10*	5	9	-2	13	-8	4	12 "Fundo"
10	5*	9	-2	13	-8	4	12
5*	10	9	-2	13	-8	4	12
5	10	9*	-2	13	-8	4	12
5	9*	10	-2	13	-8	4	12
5	9	10	-2*	13	-8	4	12
5	9	-2*	10	13	-8	4	12
5	-2*	9	10	13	-8	4	12
-2*	5	9	10	13	-8	4	12
-2	5	9	10	13*	-8	4	12
-2	5	9	10	13	-8*	4	12

Ordenação pelo Método da Bolha

"Superfície"	-2	5	9	10	-8*	13	4	12	"Fundo"
	-2	5	9	-8*	10	13	4	12	
	-2	5	-8*	9	10	13	4	12	
	-2	-8*	5	9	10	13	4	12	
	-8*	-2	5	9	10	13	4	12	
	-8	-2	5	9	10	13	4*	12	
	-8	-2	5	9	10	4*	13	12	
	-8	-2	5	9	4*	10	13	12	
	-8	-2	5	4*	9	10	13	12	
	-8	-2	4*	5	9	10	13	12	
	-8	-2	4	5	9	10	13	12*	
	-8	-2	4	5	9	10	12*	13	

FIM

Algoritmo de Ordenação pelo Método da Bolha

➡ Entrada: Vetor V com n elementos desordenados

```

para i : = 1 ... n faça
    BOLHA : = i
    enquanto BOLHA > 1 faça
        (*)   se v[BOLHA] < v[BOLHA-1]
                então
                    aux          : = v[BOLHA-1]      % troca
                    v[BOLHA-1] : = v[BOLHA]
                    v[BOLHA]   : = aux
                    BOLHA      : = BOLHA - 1        % bolha "sobe"
                senão
                    BOLHA : = 1                    % fim da "subida"

```

➡ Complexidade:

⇒ No pior caso são efetuadas **i-1** comparações na linha (*), para cada valor de **i** que inicia a variável BOLHA. Portanto, temos $0+1+2+\dots+n-1 = n(n-1)/2 = O(n^2)$ comparações

Exercício

➡ Exercício

- Desenhar as trocas de elementos ao ordenar a lista abaixo utilizando ordenação pelo método da bolha
- Quantas trocas de elementos foram efetuadas?

20	13	17	-9	4	8	12
----	----	----	----	---	---	----

Tempo: 5 minutos

Solução

"Superfície"	20*	13	17	-9	4	8	12	"Fundo"
	20	13*	17	-9	4	8	12	
	13*	20	17	-9	4	8	12	
	13	20	17*	-9	4	8	12	
	13	17*	20	-9	4	8	12	
	13	17	20	-9*	4	8	12	
	13	17	-9*	20	4	8	12	
	13	-9*	17	20	4	8	12	
	-9*	13	17	20	4	8	12	

Solução

"Superfície"	-9	13	17	20	4*	8	12	"Fundo"
	-9	13	17	4*	20	8	12	
	-9	13	4*	17	20	8	12	
	-9	4*	13	17	20	8	12	
	-9	4	13	17	20	8*	12	
	-9	4	13	17	8*	20	12	
	-9	4	13	8*	17	20	12	
	-9	4	8*	13	17	20	12	

Solução

"Superfície"	-9	4	8	13	17	20	12*	"Fundo"
	-9	4	8	13	17	12*	20	
	-9	4	8	13	12*	17	20	
	-9	4	8	12*	13	17	20	

➡ Foram efetuadas 14 trocas de elementos

Exercício Final

➡ Compare as ordenações por seleção e pelo método da bolha com relação ao número de trocas entre elementos no pior caso.

- ☐ Para finalizar a aula, uma excelente animação dos algoritmos de ordenação apresentados está disponibilizada na Internet e pode ser vista ao clicar no botão abaixo, aproveite.

ir