



Estruturas de Dados I

Algoritmos de Ordenação

Prof. Dr. Lidio Mauro Lima de Campos

limadecampos@gmail.com

Universidade Federal do Pará – UFPA

ICEN

FACOMP

Agenda

- **Introdução (Sorting).**
- **Algoritmos Iterativos de Ordenação:**
 - **Ordenação por Trocas - Bolha (*Bubble Sort*).**
 - **Ordenação por Inserção (*Insertion Sort*).**
 - **Método da Inserção Direta.**
 - **Métodos de Incrementos Decrescentes - ShellSort.**
 - **Ordenação por Seleção (*Selection Sort*).**
 - **Método da Seleção Direta.**
 - **Merge Sort**
 - **Exercícios**

Exercícios

- 1) O bubble sort com frequência é considerado o método de ordenação mais ineficiente, já que ele precisa realizar a troca de itens sem saber qual será sua posição final. Essas trocas são muito custosas. Em particular, se durante uma passagem não houver trocas, então sabemos que a lista está ordenada. O bubble sort pode ser modificado para terminar antes se descobrir que a lista ficou ordenada. Isso significa que para listas que requerem apenas algumas passagens, o bubble sort pode ter a vantagem de reconhecer a lista ordenada e parar. Efetue as modificações no algoritmo para contemplar essa situação.

Exercícios

- 2) Determine qual das seguintes classificações é mais eficiente:
 - i. a classificação por inserção simples,
 - ii. a classificação por seleção direta,
 - iii. a classificação por bolha.
- <https://replit.com/@sandrooliveira/testetempo>

```
=====
Lista quase ordenada
=====
SelectionSort 2.69408
InsertionSort: 0.04739
BubbleSort: 3.71867
BubbleSortAlterado: 0.00765
=====
Lista aleatória
=====
SelectionSort 2.97445
InsertionSort: 3.26993
BubbleSort: 6.87409
BubbleSortAlterado: 7.66506
```

Exercícios

- 2) Determine qual das seguintes classificações é mais eficiente:

- i. a classificação por inserção simples,
- ii. a classificação por seleção direta,
- iii. a classificação por bolha.

- <https://math.hws.edu/eck/js/sorting/xSortLab.html>

Algorithm	Array Count	Array Size	Comparisons	Copies	Seconds
• -----	-----	-----	-----	-----	-----
• Bubble Sort	10	10000	499950000	749242188	4.368
• Selection Sort	10	10000	499950000	299970	2.854
• Insertion Sort	10	10000	250264399	250364486	2.012
• Bubble Sort	30	10000	1499850000	2248540947	32.000
• Selection Sort	30	10000	1499850000	899910	16.968
• Insertion Sort	30	10000	750286457	750586674	3.814

Exercícios

- 2) Determine qual das seguintes classificações é mais eficiente:
 - i. a classificação por inserção simples,
 - ii. a classificação por seleção direta,
 - iii. a classificação por bolha.
- ORDEM 1 - Ordem 1: lista ordenada em ordem crescente.

Tamanho do vetor: 100

Algoritmo	Tempo(ms)	Comparações	Movimentações
Bubble sort	0,0988	5050	0
Selection Sort	0,0602	4950	297
Insertion sort	0,0038	99	198
Quick sort	0,0141	606	189

Exercícios

- 2)Determine qual das seguintes classificações é mais eficiente:
 - i. a classificação por inserção simples,
 - ii. a classificação por seleção direta,
 - iii. a classificação por bolha.
- Ordem 2: lista ordenada em ordem decrescente.

Tamanho do vetor: 100

Algoritmo	Tempo(ms)	Comparações	Movimentações
Bubble sort	0,2045	5050	14850
Selection Sort	0,0750	4950	297
Insertion sort	0,1173	99	5148
Quick sort	0,0147	610	336

Exercícios

- 2) Determine qual das seguintes classificações é mais eficiente:
 - i. a classificação por inserção simples,
 - ii. a classificação por seleção direta,
 - iii. a classificação por bolha.
- Ordem 3: lista desordenada de números aleatórios.

Tamanho do vetor: 100

Algoritmo	Tempo(ms)	Comparações	Movimentações
Bubble sort	0,1596	5050	6777
Selection Sort	0,0698	4950	297
Insertion sort	0,0570	99	2457
Quick sort	0,0314	897	576

Exercícios

- 3) Suponha que você tenha a seguinte lista de números para ordenar:
- [5, 3, 1, 9, 8, 2, 4, 7] qual lista representa a lista parcialmente ordenada depois de três passagens completas do bubble sort?

```
i= 7
-----
j= 0
[3, 5, 1, 9, 8, 2, 4, 7]
j= 1
[3, 1, 5, 9, 8, 2, 4, 7]
j= 2
[3, 1, 5, 9, 8, 2, 4, 7]
j= 3
[3, 1, 5, 8, 9, 2, 4, 7]
j= 4
[3, 1, 5, 8, 2, 9, 4, 7]
j= 5
[3, 1, 5, 8, 2, 4, 9, 7]
j= 6
[3, 1, 5, 8, 2, 4, 7, 9]
```

```
i= 6
-----
j= 0
[1, 3, 5, 8, 2, 4, 7, 9]
j= 1
[1, 3, 5, 8, 2, 4, 7, 9]
j= 2
[1, 3, 5, 8, 2, 4, 7, 9]
j= 3
[1, 3, 5, 2, 8, 4, 7, 9]
j= 4
[1, 3, 5, 2, 4, 8, 7, 9]
j= 5
[1, 3, 5, 2, 4, 7, 8, 9]
```

```
i= 4
-----
j= 0
[1, 3, 2, 4, 5, 7, 8, 9]
j= 1
[1, 2, 3, 4, 5, 7, 8, 9]
j= 2
[1, 2, 3, 4, 5, 7, 8, 9]
j= 3
[1, 2, 3, 4, 5, 7, 8, 9]
```

```
i= 3
-----
j= 0
[1, 2, 3, 4, 5, 7, 8, 9]
j= 1
[1, 2, 3, 4, 5, 7, 8, 9]
j= 2
[1, 2, 3, 4, 5, 7, 8, 9]
```

Exercícios

- 3) Suponha que você tenha a seguinte lista de números para ordenar:
- [5, 3, 1, 9, 8, 2, 4, 7] qual lista representa a lista parcialmente ordenada depois de três passagens completas do bubble sort?

i= 2	i= 1
-----	-----
j= 0	j= 0
[1, 2, 3, 4, 5, 7, 8, 9]	[1, 2, 3, 4, 5, 7, 8, 9]
j= 1	
[1, 2, 3, 4, 5, 7, 8, 9]	

- 4) Suponha que você tenha a seguinte lista de números para ordenar:
- [5, 3, 1, 9, 8, 2, 4, 7] quantas iterações são realizadas para ordenar o array dado?

Exercícios

- 4) Suponha que você tenha a seguinte lista de números para ordenar:
- [5, 3, 1, 9, 8, 2, 4, 7] quantas iterações são realizadas para ordenar o array dado ?

```
i= 7
-----
j= 0
[3, 5, 1, 9, 8, 2, 4, 7]
j= 1
[3, 1, 5, 9, 8, 2, 4, 7]
j= 2
[3, 1, 5, 9, 8, 2, 4, 7]
j= 3
[3, 1, 5, 8, 9, 2, 4, 7]
j= 4
[3, 1, 5, 8, 2, 9, 4, 7]
j= 5
[3, 1, 5, 8, 2, 4, 9, 7]
j= 6
[3, 1, 5, 8, 2, 4, 7, 9]
-----
i= 6
-----
j= 0
[1, 3, 5, 8, 2, 4, 7, 9]
j= 1
[1, 3, 5, 8, 2, 4, 7, 9]
j= 2
[1, 3, 5, 8, 2, 4, 7, 9]
j= 3
[1, 3, 5, 2, 8, 4, 7, 9]
j= 4
[1, 3, 5, 2, 4, 8, 7, 9]
j= 5
[1, 3, 5, 2, 4, 7, 8, 9]
```

```
i= 5
-----
j= 0
[1, 3, 5, 2, 4, 7, 8, 9]
j= 1
[1, 3, 5, 2, 4, 7, 8, 9]
j= 2
[1, 3, 2, 5, 4, 7, 8, 9]
j= 3
[1, 3, 2, 4, 5, 7, 8, 9]
j= 4
[1, 3, 2, 4, 5, 7, 8, 9]
-----
i= 4
-----
j= 0
[1, 3, 2, 4, 5, 7, 8, 9]
j= 1
[1, 2, 3, 4, 5, 7, 8, 9]
j= 2
[1, 2, 3, 4, 5, 7, 8, 9]
j= 3
[1, 2, 3, 4, 5, 7, 8, 9]
-----
```

```
i= 3
-----
j= 0
[1, 2, 3, 4, 5, 7, 8, 9]
j= 1
[1, 2, 3, 4, 5, 7, 8, 9]
j= 2
[1, 2, 3, 4, 5, 7, 8, 9]
[1, 2, 3, 4, 5, 7, 8, 9]
```

Exercícios

- 5) Suponha que você tenha a seguinte lista de números para ordenar: [11, 7, 12, 14, 19, 1, 6, 18, 8, 20] qual lista representa a lista parcialmente ordenada depois de três passagens completas da ordenação por seleção?
- [1, 7, 12, 14, 19, 11, 6, 18, 8, 20]
- [1, 6, 12, 14, 19, 11, 7, 18, 8, 20]
- [1, 6, 7, 14, 19, 11, 12, 18, 8, 20]
- 6) Suponha que você tenha a seguinte lista de números para ordenar:
- [19, 1, 9, 7, 3, 10, 13, 15, 8, 12] qual lista representa a lista parcialmente ordenada depois de três passagens completas do bubble sort?
- $i=9 \dots 0$, $j=0$ até $i-1$
 - $j=0$
 - [1, 19, 9, 7, 3, 10, 13, 15, 8, 12]
 - $j=1$
 - [1, 9, 19, 7, 3, 10, 13, 15, 8, 12]
 - $j=2$
 - [1, 9, 7, 19, 3, 10, 13, 15, 8, 12]

Exercícios

- 7)(Bubble Sort) Considerando uma lista (vetor) com os seguintes elementos: 1 - 8 - 9 - 2 - 15 - 20. Após percorrer pela primeira vez a lista, efetuando as comparações indicadas no algoritmo, quantas trocas serão realizadas? Considere o objetivo de ordenar a lista em ordem crescente.
- $i = n - 1 = 6 - 1 = 5$
- Para cada i , $j = 0 \dots i - 1$
- $i = 5$
 - $j = 0..4$
 - $j = 0$ 1, 8, 9, 2, 15, 20
 - $j = 1$ 1, 8, 9, 2, 15, 20
 - $j = 2$ 1, 8, 9, 2, 15, 20 troca 9 com 2
 - $j = 3$ 1, 8, 2, 9, 15, 20
 - $j = 4$ 1, 8, 2, 9, 15, 20

Exercícios

- 8)(Bubble Sort) Considerando uma lista (vetor) com os seguintes elementos: 1 - 8 - 9 - 2 - 15 - 20. Após percorrer pela segunda vez a lista, Como ficaria a lista após percorrer o vetor duas vezes, realizando as comparações e trocas?
- $i = n - 1 = 6 - 1 = 5$
- Para cada i , $j = 0 \dots i - 1$
- $i = 4$
 - $j = 0..3$
 - $j = 0$ 1, 8, 2, 9, 15, 20
 - $j = 1$ 1, 8, 2, 9, 15, 20 troca 8 com 2
 - $j = 2$ 1, 2, 8, 9, 15, 20
 - $j = 3$ 1, 2, 8, 9, 15, 20

Exercícios

- 9) Considerando as afirmações abaixo, assinale a(s) alternativa(s) correta(s).
- I - O Bubble Sort é um método difícil de ser implementado porque apresenta muitas comparações e a necessidade de uso de duas repetições
- II - No método abordado nesta atividade as comparações são feitas de forma sucessiva, o primeiro elemento com o segundo, o segundo com o terceiro, o terceiro com o quarto e assim por diante.
- III - O Bubble Sort é considerado pouco eficiente, principalmente quando a lista de valores é grande.
- Resp: II e III

Exercícios

- 10) Você encontra-se diante do seguinte problema. Há uma lista de valores de salários de funcionários e a empresa deseja ver a listagem em uma ordem que permita verificar quais ganham salários maiores e posteriormente aqueles que ganham menos. O objetivo é verificar possíveis incompatibilidades nos casos dos que são mais bem remunerados. A lista dos salários está armazenada na memória em um vetor denominado `v_salario`. Para solucionar o problema será utilizado o Bubble Sort. Assinale a alternativa abaixo que mostra as instruções de comparação e troca que resolvem corretamente o problema proposto. Considere que `i` representa a posição atual da lista.

A. <input type="checkbox"/> if (<code>v_salario[i] < v_salario[i + 1]</code>) { <code>aux = v_salario[i];</code> <code>v_salario[i] = v_salario[i + 1];</code> <code>v_salario[i + 1] = aux;</code> <code>troca = true;</code> }	B. <input checked="" type="checkbox"/> if (<code>v_salario[i] > v_salario[i + 1]</code>) { <code>aux = v_salario[i];</code> <code>v_salario[i] = v_salario[i + 1];</code> <code>v_salario[i + 1] = aux;</code> <code>troca = true;</code> }	C. <input checked="" type="checkbox"/> if (<code>v_salario[i + 1] < v_salario[i]</code>) { <code>aux = v_salario[i];</code> <code>v_salario[i] = v_salario[i + 1];</code> <code>v_salario[i + 1] = aux;</code> <code>troca = true;</code> }	D. <input checked="" type="checkbox"/> if (<code>v_salario[i] > v_salario[i + 1]</code>) { <code>v_salario[i] = v_salario[i + 1];</code> <code>aux = v_salario[i];</code> <code>v_salario[i + 1] = aux;</code> <code>troca = true;</code> }
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Resp: A

Exercícios

- 11)(**Insertion Sort**) A cada etapa de ordenação é eleito um valor que será comparado com os anteriores para que seja colocado em uma posição ordenada. Considerando uma lista (vetor) com os seguintes elementos: 9 - 1 - 8 - 2 - 15 - 20, qual seria o primeiro elemento eleito?. Mostre todas as etapas da ordenação.
- 9-**1**-8-2-15-20 i=1
- 1-9-**8**-2-15-20 i=2
- 1-8-9-**2**-15-20 i=3
- 1-2-8-9-**15**-20 i=4
- 1-2-8-9-15-**20** i=5
- 1-2-8-9-15-20 i=6

Exercícios

- **12)(Insertion Sort)**

- Considerando as afirmações abaixo, assinale a(s) alternativa(s) correta(s).
I - O Insertion Sort é um método difícil de ser implementado, pois apresenta muitas comparações entre elementos de posições diferentes.
II - No método abordado nesta atividade um elemento é selecionado e comparado com os das posições anteriores e também com os armazenados nas posições seguintes.
III - O Insertion Sort é adequado na maioria dos casos quando a lista a ser ordenada possui poucos elementos.
- Resp: III

Exercícios

• 13)

Analise a seguinte figura.

	1	2	3	4	5	6
Chaves iniciais:	O	R	D	E	N	A
h = 4	N	A	D	E	O	R
h = 2	D	A	N	E	O	R
h = 1	A	D	E	N	O	R

O método de ordenação indicado pela figura é o

- ☐ A *Shellsort.*
- ☐ B *Bubblesort.*
- ☐ C *Insertsort.*
- ☐ D *Quicksort.*
- ☐ E *Heapsort.*

Exercícios

14) Dada a sequência de números: 3 4 9 2 5 18.

Ordene em ordem crescente utilizando o algoritmo aprendido em sala (MergeSort), apresentando a sequência dos números a cada passo.

15) Dada a sequência de números: 1 4 8 3 6 5 2 7

Ordene em ordem crescente utilizando o algoritmo aprendido em sala (MergeSort), apresentando a sequência dos números a cada passo.

Exercícios

- 16) [POSCOMP 2013 – COPS – UEL] Sobre o comportamento assintótico do algoritmo de ordenação MergeSort (veja um exemplo de implementação abaixo), assinale a alternativa que apresenta, corretamente, sua complexidade.

```
MERGESORT(v, i, j)
    Se (i<j) então
        m = (i+j)/2;
        MERGESORT(v, i, m);
        MERGESORT(v, m+1, j);
        MESCLAR(v, i, m, j);
    Fim;
```

- a) $O(\log n)$
- b) $O(n \log n)$
- c) $O(n^2)$
- d) $O(n^3)$
- e) $O(2n)$

Exercícios

Algoritmo	Comparações			Movimentações			Espaço
	Melhor	Médio	Pior	Melhor	Médio	Pior	
Bubble	$O(n^2)$			$O(n^2)$			$O(1)$
Selection	$O(n^2)$			$O(n)$			$O(1)$
Insertion	$O(n)$	$O(n^2)$		$O(n)$	$O(n^2)$		$O(1)$
Merge	$O(n \log n)$			–			$O(n)$
Quick	$O(n \log n)$		$O(n^2)$	–			$O(n)$
Shell	$O(n^{1.25})$ ou $O(n (\ln n)^2)$			–			$O(1)$

Referências

- Thomas H. [Cormen](#), Charles E. [Leiserson](#), Ronald L. [Rivest](#), and Clifford Stein. Algoritmos – Teoria e Prática, Tradução da Segunda Edição. Campus, 2016.
- Ziviani, N. Projeto de Algoritmos Com Implementações em Pascal e C, Pioneira Thomson Learning, 4ed. 2009.