



USP - ICMC - SSC SSC 0300 - 2o. Semestre 2013

Disciplina de Linguagem de Programação e Aplicações [Eng. Elétrica / Automação]

Prof. Dr. Fernando Santos Osório / PAE: Rafael Klaser (LRM / ICMC)

LRM - Laboratório de Robótica Móvel do ICMC / CROB-SC

Email: fosorio@icmc.usp.br ou fosorio@gmail.com

Página Pessoal: <http://www.icmc.usp.br/~fosorio/>

Material on-line:

Wiki ICMC - <http://wiki.icmc.usp.br/index.php>

Wiki SSC0300 - [http://wiki.icmc.usp.br/index.php/SSC-300-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-300-2013(fosorio))

Agenda:

- **Métodos de Ordenação**

Simples

- **Bubble Sort (Bolha)**
- **Selection Sort**
- **Insertion Sort**

Avançados

- **Merge Sort**
- **Quick Sort**

- **Exercícios**

Informações Complementares a Atualizadas:

Consulte REGULARMENTE o material disponível na WIKI

[http://wiki.icmc.usp.br/index.php/SSC-300-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-300-2013(fosorio))

Ordenação de Dados: Vetor ou Lista de Dados (encadeada)

Ordenar é o processo de rearranjar um conjunto de objetos em uma ordem ascendente ou descendente.

A ordenação visa facilitar a recuperação e/ou uso posterior de itens do conjunto ordenado.

Por exemplo:

Dificuldade de se utilizar um catálogo telefônico se os nomes das pessoas não estivessem listados em ordem alfabética.

03	11	25	37	41	56	83	A	C	D	L	P	R	Z
----	----	----	----	----	----	----	---	---	---	---	---	---	---

Ordenação de Dados: Métodos Simples (+ Lentos)

BUBBLE SORT

SELECTION SORT

E muitos outros...

INSERTION SORT

Ordenação de Dados: Métodos Otimizados (+ Rápidos)

MERGE SORT

E muitos outros...

QUICK SORT

Ordenação de Dados: Métodos Simples (+ Lentos)

1. BUBBLE SORT

2. SELECTION SORT

E muitos outros...

3. INSERTION SORT

Ordenação de Dados: Métodos Otimizados (+ Rápidos)

4. MERGE SORT

E muitos outros...

5. QUICK SORT

Ordenação de Dados: Métodos Simples

BUBBLE SORT

Exemplo: Suponha que se deseja classificar em ordem crescente o seguinte vetor de valores:

28	26	30	24	25
----	----	----	----	----

Primeira Varredura

28	26	30	24	25	compara par (28, 26) : troca
26	28	30	24	25	compara par (28, 30) : não troca
26	28	30	24	25	compara par (30, 24) : troca
26	28	24	30	25	compara par (30, 25) : troca
26	28	24	25	30	fim da primeira varredura

Ordenação de Dados: Métodos Simples

BUBBLE SORT

Exemplo: Suponha que se deseja classificar em ordem crescente o seguinte vetor de valores:

28	26	30	24	25
----	----	----	----	----

Segunda Varredura: 26 28 24 25 30

26 28 24 25 30 compara par (26, 28) : não troca


26 28 24 25 30 compara par (28, 24) : troca


26 24 28 25 30 compara par (28, 25) : troca


26 24 25 28 30 fim da segunda varredura

Ordenação de Dados: Métodos Simples

BUBBLE SORT

Exemplo: Suponha que se deseja classificar em ordem crescente o seguinte vetor de valores:

28	26	30	24	25
----	----	----	----	----

Terceira Varredura: 26 24 25 28 30

26 24 25 28 30 compara par (26, 24) : troca


24 26 25 28 30 compara par (26, 25) : troca


24 25 26 28 30 fim da terceira varredura

Ordenação de Dados: Métodos Simples

BUBBLE SORT

Exemplo: Suponha que se deseja classificar em ordem crescente o seguinte vetor de valores:

28	26	30	24	25
----	----	----	----	----

```
void bubbleSort(int a[])
{
    for (int i = a.length-1; i>0; i--) { // nro de varreduras (n-1)
        for (int j = 0; j<i; j++) { // percorre vetor
            if (a[j] > a[j+1]) { // troca par de posição
                int T = a[j];
                a[j] = a[j+1];
                a[j+1] = T;
            } // if
        } // for
    } // for
}
```

Ordenação de Dados: Métodos Simples

BUBBLE SORT

Exemplo: Suponha que se deseja classificar em ordem crescente o seguinte vetor de valores:

28	26	30	24	25
----	----	----	----	----

Comentários:

- Método muito simples de implementar
- Método muito lento de executar
- Melhorias:
 - * Se não ocorreram trocas em uma passagem, já está ordenado
 - * *Shaker Sort*: Bubble Sort que vai-e-volta deslocando dados

Complexidade: $O(N*N)$ ou $O(N^2)$

Ordenação de Dados: Métodos Simples

SELECTION SORT

Princípio de classificação

- A seleção do menor dado é feita por pesquisa seqüencial;
- O menor dado encontrado é permutado com a que ocupa a posição inicial do vetor, que fica reduzido de um elemento;
- O processo de seleção é repetido para a parte restante do vetor, até que todos os dados tenham sido selecionados e colocados em suas posições definitivas

Ordenação de Dados: Métodos Simples

SELECTION SORT

Exemplo

Suponha que se deseja classificar o seguinte vetor:

9 25 10 18 5 7 15 3

Simulação das iterações necessárias
para a classificação... (teste de mesa)

Ordenação de Dados: Métodos Simples

SELECTION SORT

Iteração	Vetor	Menor Dado (seleciona)	Troca	Vetor orde- nado até a posição
1	9 25 10 18 5 7 15 3	3	9 e 3	
2	3 25 10 18 5 7 15 9	5	25 e 5	0
3	3 5 10 18 25 7 15 9	7	10 e 7	1
4	3 5 7 18 25 10 15 9	9	18 e 9	2
5	3 5 7 9 25 10 15 18	10	25 e 10	3
6	3 5 7 9 10 25 15 18	15	25 e 15	4
7	3 5 7 9 10 15 25 18	18	25 e 18	5
8	3 5 7 9 10 15 18 25			6

Ordenação de Dados: Métodos Simples

SELECTION SORT

```
void selectionSort (int a[])  
{  
    int min=0, ch;  
    for (int i=0; i<a.length-1; i++) {  
        min = i; // mínimo inicial  
        for (int j = i + 1; j<a.length; j++)  
            if (a [ j ] < a [ min ]) min = j; // acha o novo mínimo  
        ch = a [ i ];  
        a [ i ] = a [ min ];  
        a [ min ] = ch; // coloca o novo mínimo (min)  
        // na posição correta (i)  
    }  
}
```

Ordenação de Dados: Métodos Simples

SELECTION SORT

Comentários:

- Método bastante simples de implementar
- Método bastante lento de executar

Complexidade: $O(N*N)$ ou $O(N^2)$

Ordenação de Dados: Métodos Simples

INSERTION SORT

Inicialmente:

Divide o vetor em 2 segmentos:
o primeiro contendo os elementos já ordenados
o segundo contendo os elementos ainda não ordenados

No início: o 1º segmento terá apenas 1 elemento

Funcionamento:

Pega o primeiro elemento do segmento não ordenado
e procura seu lugar no segmento ordenado.

Ordenação de Dados: Métodos Simples

INSERTION SORT

Vetor original

18	15	7	9	23	16	14
----	----	---	---	----	----	----

Divisão inicial

18	15	7	9	23	16	14
----	----	---	---	----	----	----



Primeira iteração

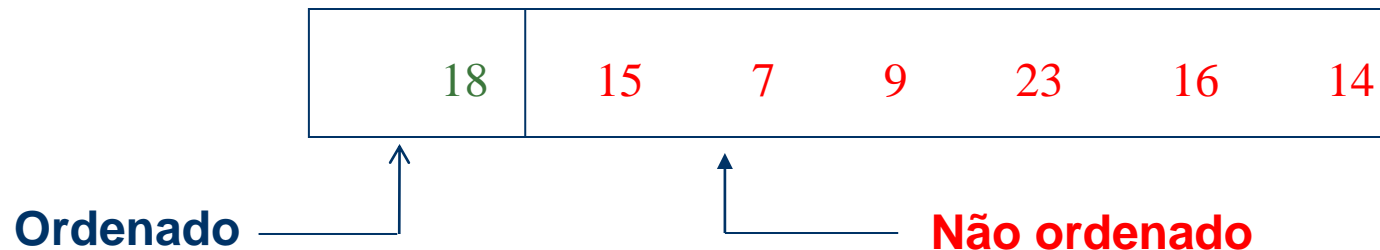
15	18	7	9	23	16	14
----	----	---	---	----	----	----

Segunda iteração

7	15	18	9	23	16	14
---	----	----	---	----	----	----

Ordenação de Dados: Métodos Simples

INSERTION SORT



- [15, 18, 7, 9, 23, 16, 14] após 1ª interação
- [7, 15, 18, 9, 23, 16, 14] após 2ª interação
- [7, 9, 15, 18, 23, 16, 14] após 3ª interação
- [7, 9, 15, 18, 23, 16, 14] após 4ª interação
- [7, 9, 15, 16, 18, 23, 14] após 5ª interação
- [7, 9, 14, 15, 16, 18, 23] após 6ª interação

Ordenação de Dados: Métodos Simples

INSERTION SORT

```
void insertionSort (int a[]) {  
    for (int i = 1; i < a.length; i++) {  
        int j = i; // pos do 1º elemento no seg. não ord.  
        int B = a[i]; // 1º elemento no seg. não ord.  
        while ((j > 0) && (a[j-1] > B)) {  
            a[j] = a[j-1];  
            j--;  
        }  
        a[j] = B;  
    }  
}
```

} buscando a posição
do 1º elemento do
segmento não
ordenado no
segmento ordenado

Ordenação de Dados: Métodos Simples

INSERTION SORT

Comentários:

- Método bastante simples de implementar
- Método bastante lento de executar
- A inserção com poucos elementos é interessante de ser usada, mas a medida que vamos aumentando o nro. de dados o algoritmo pode ir ficando bastante lento.

Complexidade: $O(N*N)$ ou $O(N^2)$

Ordenação de Dados: Métodos Simples

BUBBLE SORT

SELECTION SORT

INSERTION SORT

Demonstração prática:

Visualização através de animações

<http://www.sorting-algorithms.com/>

Ordenação de Dados: Métodos Otimizados

MERGE SORT

Funcionamento:

- “Dividir para conquistar”
- Divide os dados até que tenha blocos pequenos (2 dados), ordena estes blocos individualmente e depois junta os blocos
- Juntar dois blocos ordenados é mais fácil que ordenar eles completamente!

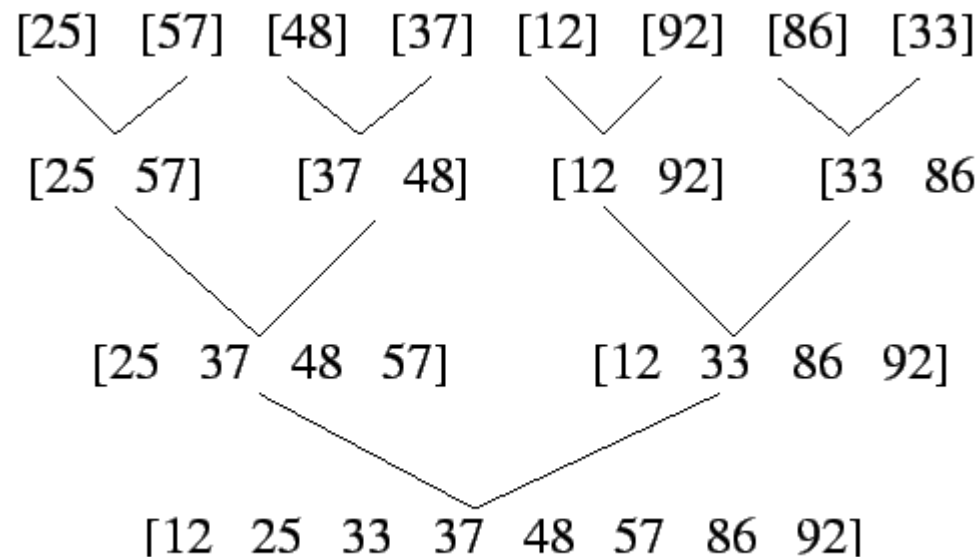
Possivelmente você já fez isto na prática...

Professores quando ordenam as provas em ordem alfabética, usualmente dividem em pilhas menores, ordenam estas pilhas individualmente (conjuntos menores), e depois juntam as pilhas mantendo a ordenação.

Ordenação de Dados: Métodos Otimizados

MERGE SORT

Exemplo:



Ordenação de Dados: Métodos Otimizados

MERGE SORT

Comentários:

- Método mais complexo de implementar (algoritmo recursivo)
- Método mais rápido de executar
- Juntar dois blocos ordenados, mantendo a ordenação, é bem mais rápido e eficiente que ordenar tudo junto.

Complexidade: $O(N \cdot \log N)$

Informações adicionais:

<http://www.lcad.icmc.usp.br/~nonato/ED/Ordenacao/node52.htm>

Demo: <http://www.sorting-algorithms.com/>

Ordenação de Dados: Métodos Otimizados

QUICK SORT

Comentários:

- Método mais complexo de implementar (algoritmo recursivo)
- Método mais rápido de executar
- Algoritmo “Clássico” de ordenação rápida disponível em diversas bibliotecas.

Informações adicionais:

<http://www.lcad.icmc.usp.br/~nonato/ED/Ordenacao/node54.htm>

Demo: <http://www.sorting-algorithms.com/>



INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO

Web institucional: <http://www.icmc.usp.br/>

Página pessoal: <http://www.icmc.usp.br/~fosorio/>

Página do Grupo de Pesquisa: <http://www.lrm.icmc.usp.br/>

E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Linguagem de Programação e Aplicações SSC300

WIKI - [http://wiki.icmc.usp.br/index.php/SSC-300-2013\(fosorio\)](http://wiki.icmc.usp.br/index.php/SSC-300-2013(fosorio))

- > Programa, Material de Aulas, Critérios de Avaliação,**
- > Trabalhos Práticos, Datas das Provas, Notas**