



# Listas Lineares e Operadores de Ordenação

**UFU – FACOM**

**João Henrique de Souza Pereira**

# Sumário

- **Introdução**
- **Algoritmos Básicos**
- **Complexidade**
- **Exercícios**
- **Referências**

# Introdução

- Ordenação é o processo de colocar uma lista linear em uma ordem (Aho, 1983).
- Qual a importância de ordenar?
- Para que serve a ordenação?

# Introdução

- Terminologia:

- **Chave:** Parte dos registros utilizadas para controlar a ordenação
- **Ordenação interna:** Memória principal
- **Ordenação externa:** Memória de armazenamento secundário
- **Ações:** Comparar ou Mover
- **Estabilidade:** Preservar a ordem relativa de chaves iguais no arquivo
  - » A maioria dos algoritmos básicos é estável
  - » A maioria dos algoritmos sofisticados não é estável

# Algoritmos Básicos

- Métodos elementares para pequenos conjuntos de dados, ou conjuntos com estruturas especiais (Sedgewick, 1990).
- Razões para estudar os algoritmos Básicos:
  - Entender a terminologia e mecanismos básicos
  - Úteis em algumas aplicações
    - Usado única vez, ou poucas vezes
    - Poucos elementos a ordenar
    - Arquivos ordenados, ou quase ordenados
    - Grande quantidade de chaves iguais
  - Estendem para melhores propósitos gerais

# Algoritmos Básicos

- Seleção
- Inserção
- Bolha

# Algoritmos Básicos

- Seleção: Visão Geral

- É um dos algoritmos mais simples.
- Considere uma Lista Linear com os elementos a serem ordenados.
- A Lista Linear deve ser percorrida a fim de encontrar o menor elemento.
- Assim que o mesmo for encontrado o mesmo é trocado com o elemento da primeira posição.
- Encontre o segundo menor elemento e troque-o com o elemento da segunda posição.
- Repita o processo até que toda Lista Linear esteja ordenada.

# Algoritmos Básicos

- Seleção: Implementação em Lista Linear Sequencial

```
selection(int a[], int N) {  
    int i, j, min, t;  
  
    for (i = 1; i < N; i++) {  
        min = i;  
        for (j = i+1; j <= N; j++)  
            if (a[j] < a[min]) min = j;  
        t = a[min]; a[min] = a[i]; a[i] = t;  
    }  
}
```



# Algoritmos Básicos

- Seleção: Análise

- O número de comparações no método de ordenação por seleção é da ordem de  $N^2$ , sendo  $N$  o número de elementos da Lista Linear.
- Funciona muito bem para arquivos pequenos.
- Apesar da evidente “força bruta” e simplicidade é uma boa opção quando os dados a serem movidos são muito grandes e as chaves pequenas, pois cada item é movido apenas uma vez (Sedgewick, 1990).

# Algoritmos Básicos

- Inserção: Visão Geral

- Tão simples quanto o algoritmo de seleção, no entanto, talvez um pouco mais flexível.
- Considere uma Lista Linear com os elementos a serem ordenados.
- Pegue um elemento de cada vez.
- Insira o elemento na posição correta (caso a Lista seja Sequencial, mova os elementos maiores para a direita e insira na posição vazia).
- Repita o processo até que toda a Lista Linear esteja ordenada.

# Algoritmos Básicos

- Inserção: Implementação em Lista Linear Sequencial

```
insertion(int a[], int N) {  
    int i, j, v;  
  
    for (i = 2; i <= N; i++) {  
        v = a[i]; j = i;  
        while (a[j-1] > v)  
            { a[j] = a[j-1]; j--; }  
        a[j] = v;  
    }  
}
```

# Algoritmos Básicos

- Inserção: Análise

- Semelhante ao algoritmo de seleção os elementos à esquerda do índice  $i$  estão ordenados durante a ordenação, no entanto, não estão necessariamente em sua posição final.
- O vetor estará ordenado quando o índice chegar ao final à direita.
- Quando  $v$  é o menor elemento do vetor o while passará o final à esquerda do vetor. Para corrigir isto é necessário colocar uma chave sentinela em  $a[0]$ , ou usar: `while j>1 && a[j-1] > v.`

# Algoritmos Básicos

- Bolha: Visão Geral

- É um algoritmo simples de permuta de elementos adjacentes.
- Considere um vetor com os elementos a serem ordenados.
- Percorra o vetor ordenando os elementos adjacentes.
- Os maiores elementos (mais pesados) serão movidos para o fundo e os menores (mais leves) subirão.
- Repita o processo até que o vetor esteja ordenado.

# Algoritmos Básicos

- Bolha: Implementação em Lista Linear Sequencial

```
bubble(int a[], int N) {  
    int i, j, t;  
  
    for (i = N; i >= 1; i--)  
        for (j = 2; j <= i; j++)  
            if (a[j-1] > a[j]) {  
                t = a[j-1]; a[j-1] = a[j]; a[j] = t;  
            }  
}
```

# Algoritmos Básicos

- Bolha: Análise

- Quando o maior elemento é encontrado na primeira passagem pelo vetor, este é trocado com cada um dos elementos seguintes até o final do vetor. Na segunda passagem ocorre o mesmo com o segundo maior, e assim por diante.
- É semelhante ao algoritmo de seleção, porém, com mais trabalho para colocar cada elemento em sua posição.
- Utilizado para fins didáticos.

# Complexidade

- Algoritmos básicos  $N^2$
- Algoritmos mais elaborados  $N \log N$
- Alguns casos  $N$
- Notas:
  - O algoritmo de seleção usa tempo linear quando os dados são grandes e as chaves pequenas.
  - Os algoritmos de seleção e inserção são bons se os dados estiverem quase ordenados.



# Exercícios

Descreva a sequência, passo a passo, para o algoritmo seleção, dados os 6 elementos abaixo:

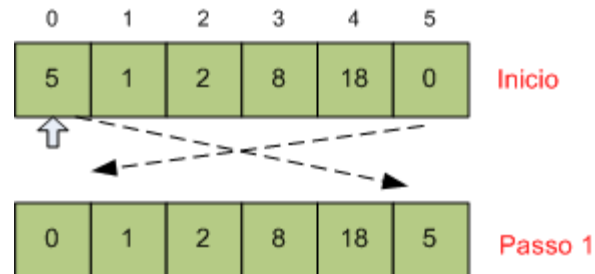
|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 2 | 3 | 4  | 5 |
| 5 | 1 | 2 | 8 | 18 | 0 |

Início

# Exercícios

Descreva a sequência, passo a passo, para o algoritmo seleção, dados os 6 elementos abaixo:

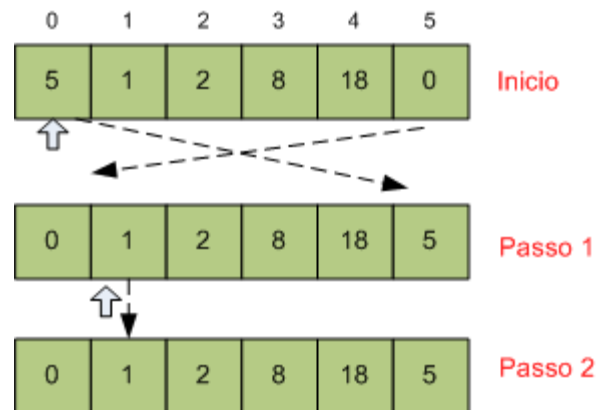
Solução:



# Exercícios

Descreva a sequência, passo a passo, para o algoritmo seleção, dados os 6 elementos abaixo:

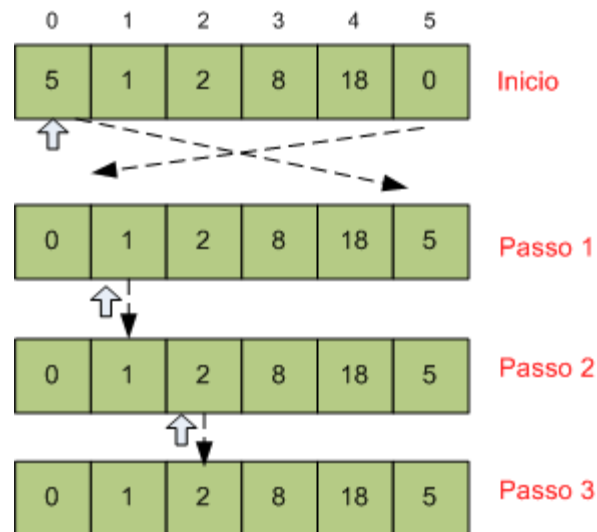
Solução:



# Exercícios

Descreva a sequência, passo a passo, para o algoritmo seleção, dados os 6 elementos abaixo:

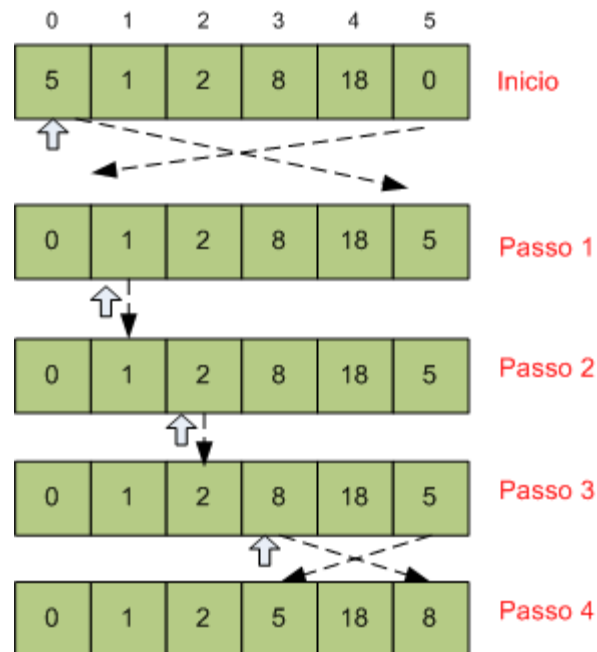
Solução:



# Exercícios

Descreva a sequência, passo a passo, para o algoritmo seleção, dados os 6 elementos abaixo:

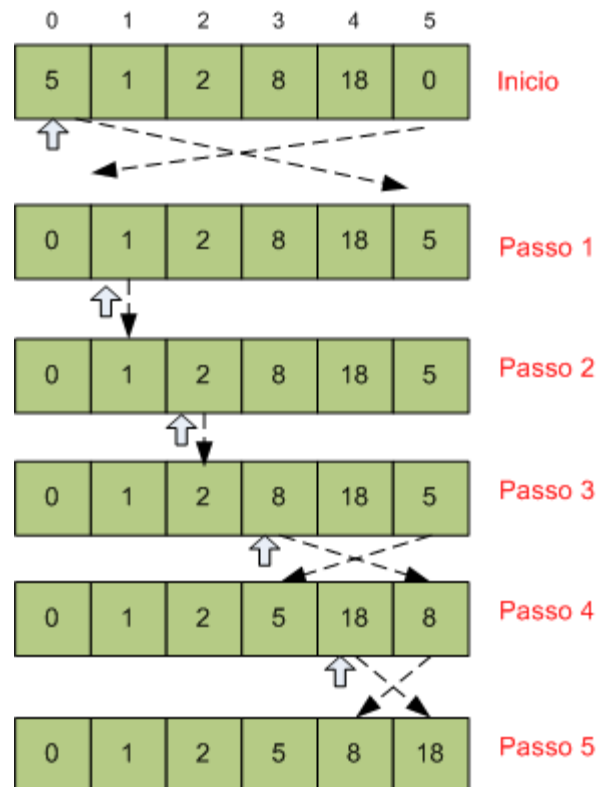
Solução:



# Exercícios

Descreva a sequência, passo a passo, para o algoritmo seleção, dados os 6 elementos abaixo:

Solução:



# Exercícios

- Implemente as operações de ordenação (ascendente e descendente) com algoritmos de “seleção”, “inserção” e “bolha” com uso de Listas Lineares Sequenciais.

# Exercícios

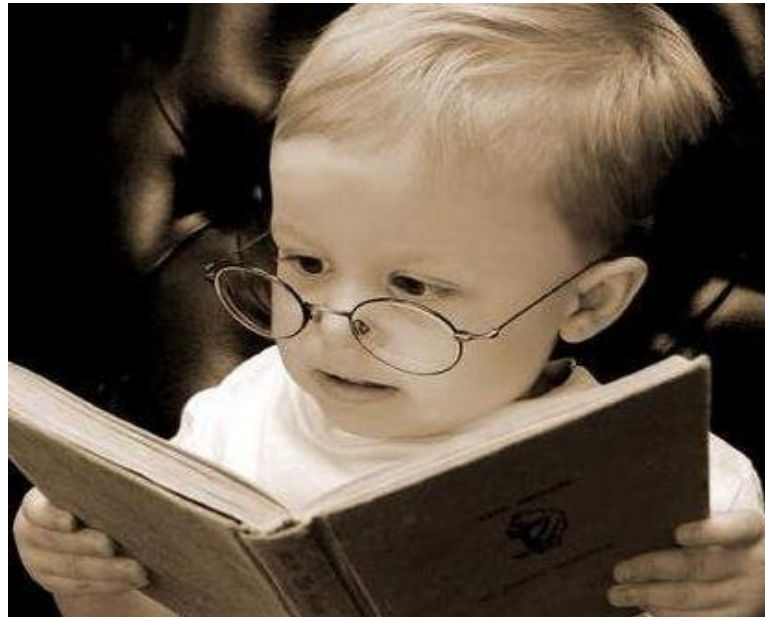
- Implemente as operações de ordenação com algoritmos de “seleção”, “inserção” e “bolha” com uso de TAD e Listas Lineares Dinamicamente Alocadas.



# Referências

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Algoritmos: Teoria e Prática, Editora Campus, 2a Edição, 2002.
2. Robert Sedgewick. Algorithms in C, Addison-Wesley, 1990.
3. Alfred V. Aho, Jeffrey D. Ullman, John E. Hopcroft. Data Structures and Algorithms. Addison-Wesley, 1st edition, 1983.
4. Autran Macêdo. Plano de Ensino 2012/2 – Disciplina: Algoritmos e Estrutura de Dados 2, <http://www.facom.ufu.br/~autran/apoio/ed/bcc-aed2-2012-2-plano-ensino>, acessado em 31/01/2013, 16:30h.

*Não é o mais forte que sobrevive, nem o mais inteligente,  
mas o que melhor se adapta às mudanças (Darwin).*



Obrigado!  
[joaohs@ufu.br](mailto:joaohs@ufu.br)