Estruturas de Dados

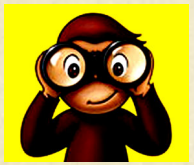
Prof. Guilherme Tomaschewski Netto
guilherme.netto@gmail.com

Roteiro

- Contextualização
- Apresentação
- Tipos de Dados
- Estrututas de Dados e seus tipos
- Listas Lineares
- Listas não lineares
- Algoritmos de Ordenação

Legendas

- Nesta apresentação serão utilizadas algumas legendas:



Indica uma referência, para quem ficou curioso e quer aprofundar mais seus conhecimentos sobre o assunto

Indica uma referência importante, leitura obrigatória.

Competências desejadas

Para compreênsão dos conceitos abordados é desejado que os alunos já tenham apropriado as seguintes competências:

- Conhecimentos básicos de algoritmos
- Compreenção de vetores e matrizes

Tipos de Dados

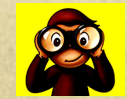
- Informática vem da contração das palavras “informação + automática”, e a forma mais primitiva da informação em um programa são as variáveis.

- Variáveis são espaços
  armazenar valores nu

Por exemplo, **int x;** espec
variável que pode armaze
valor.

```
public class Numero {
  public static void main(String args[]) {
   int x;
   x = 10+13;
   System.out.println("O valor numérico é " + x);
}}
```

Tipos de Dados



http://www.javaranch.com/campfire/StoryCups.jsp

Tipos de Dados Primitivos

Tipo	Descrição
boolean	Pode assumir o valor true ou o valor false
char	Caractere em notação Unicode de 16 bits. Serve para a armazenagem de dados alfanuméricos. Também pode ser usado como um dado inteiro com valores na faixa entre 0 e 65535.
byte	Inteiro de 8 bits em notação de complemento de dois. Pode assumir valores entre -27=-128 e 27-1=127.
short	Inteiro de 16 bits em notação de complemento de dois. Os valores possíveis cobrem a faixa de -2-15=-32.768 a 215-1=32.767
int	Inteiro de 32 bits em notação de complemento de dois. Pode assumir valores entre -231=2.147.483.648 e 231-1=2.147.483.647.
long	Inteiro de 64 bits em notação de complemento de dois. Pode assumir valores entre -263 e 263-1.
float	Representa números em notação de ponto flutuante normalizada em precisão simples de 32 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável por esse tipo é 1.40239846e-46 e o maior é 3.40282347e+38
double	Representa números em notação de ponto flutuante normalizada em precisão dupla de 64 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável é 4.94065645841246544e-324 e o maior é 1.7976931348623157e+308

BOENTE, 2003 e GOODRICH, 2002

Conjuntos de Variáveis (Vetores)

- Vetores são conjuntos de variáveis, tornam possível a manipulação de uma quantidade grande de dados.
- Ao vetor sempre é atribuido um nome, um tamanho, e um índice que irá distinguir cada elemento.
- Ex: char x[] = new char[5];

  x[0] = 'A';

Estruturas de Dados

- É o ramo da ciência da computação que estuda os diversos mecanismos de organização dos dados para atender os diferentes tipos de processamento.

Ex. Listas Simples, Filas, Pilhas, Árvores, Grafos.

Estruturas de Dados

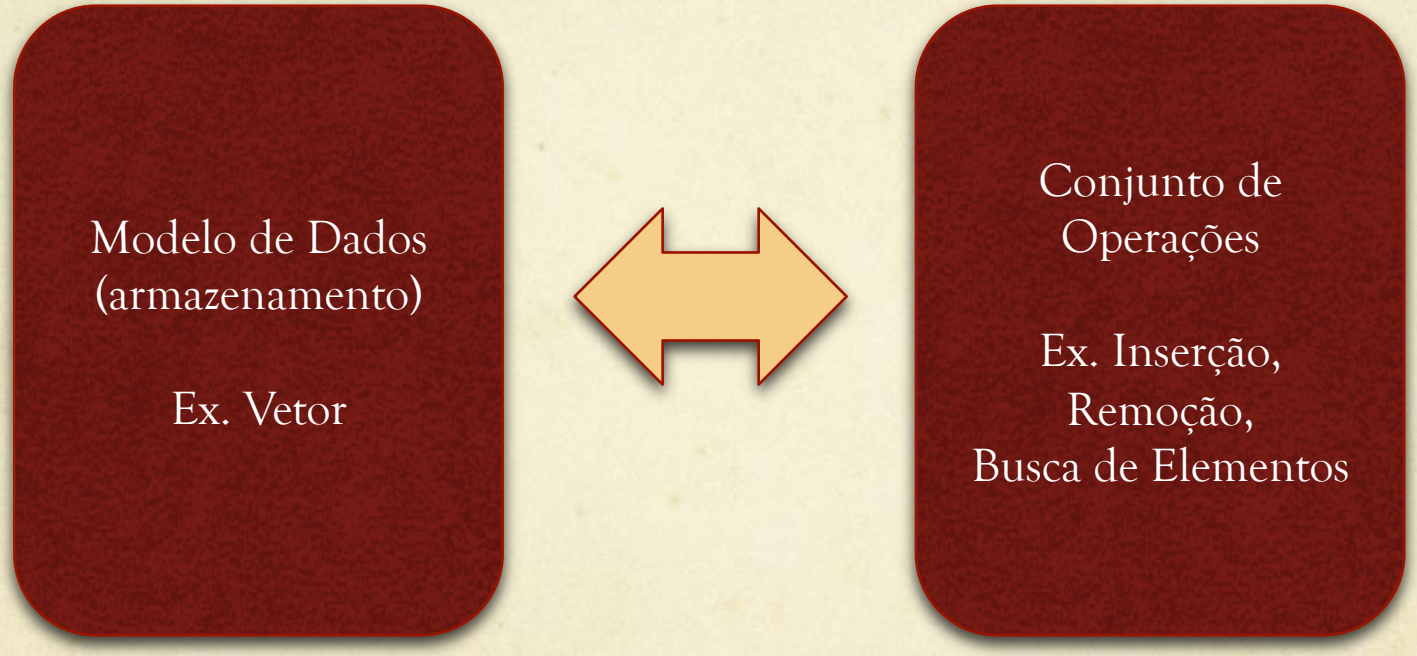
- **Listas Lineares**

  Listas simples, Filas, Pilhas

- **Lista não-Lineares**

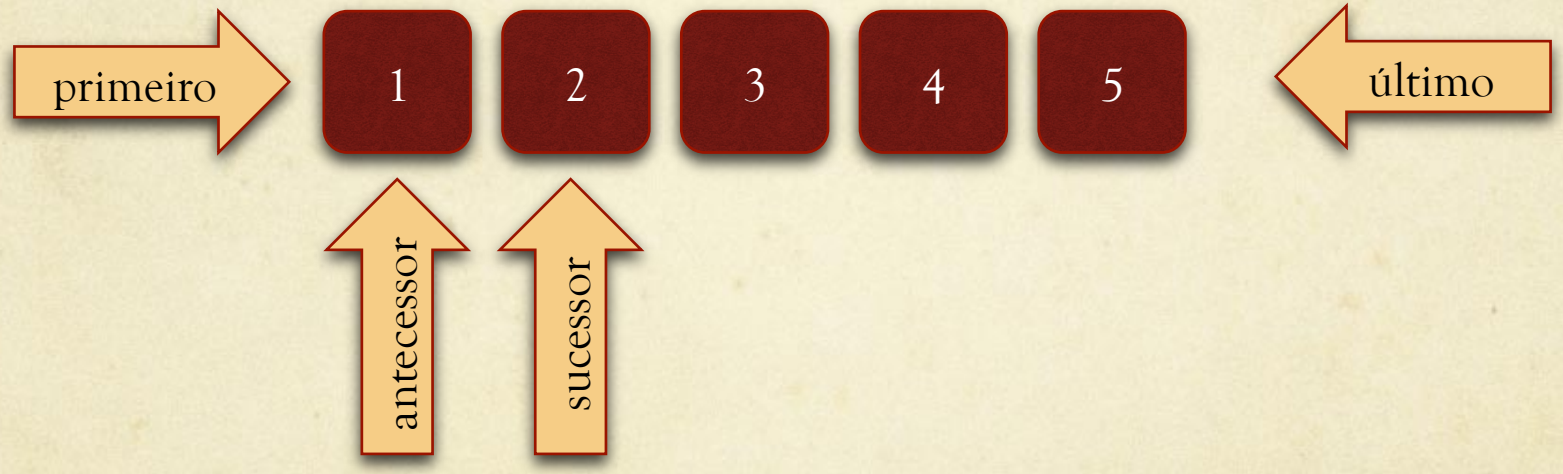
  Árvores, Grafos

Estruturas de Dados



Listas Lineares

- Conjunto de dados que mantém a relação de ordem Linear entre os componentes.



ALVES, 2009

Listas Lineares

- Operações

  Inserção, Remoção, Percurso

Pilha

- Uma Lista com princípio:

  o último a entrar é o primeiro a sair

  Last In First Out(LIFO)

Operções de inserção(push) e retirada(pop) somente no último elemento da lista (topo)

ALVES, 2009

Pilha

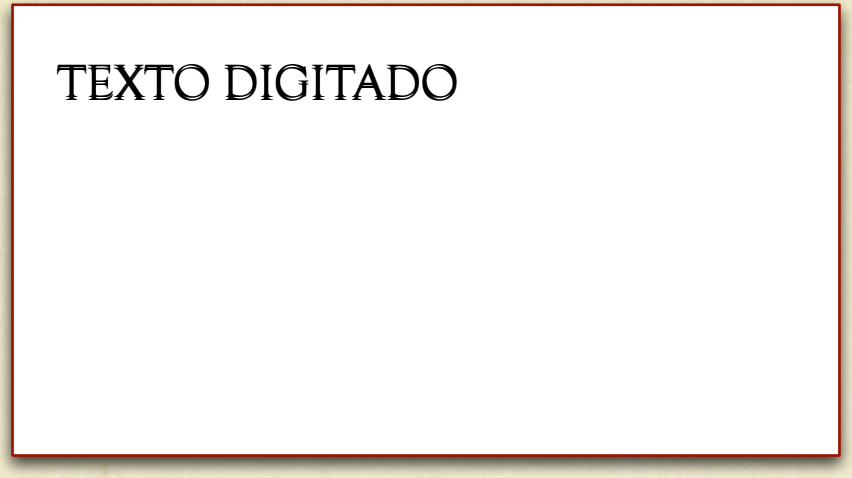
Para uma pilha P, são definidas as operações:

push(x), insere o elemento X na pilha
pop(), retira um elemento do topo da pilha

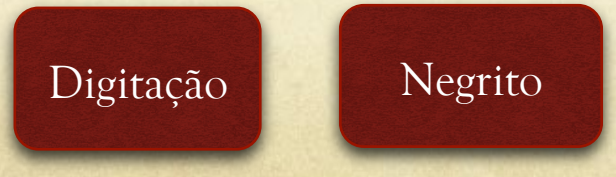
Operação	saída	P
push(5)	-	(5)
push(6)	-	(5,6)
pop()	6	(5)
push(3)	-	(5,3)
push(4)	-	(5,3,4)
pop()	4	(5,3)
pop()	3	(5)
pop()	5	()
pop()	erro	()

Pilha

Exemplo prático



Pilha que armazena as operações



Pilha

- Modelo de Implementação

```
public class Pilha{
    int v[] = new int[10];
    int topo=-1;

   public void push(int x){
                if(topo<v.length-1)
                    v[++topo]=x;
        }
    public int pop(){
          if(topo>-1)
                    return v[topo--];
        else
            return 0;
   }
}
```

Fila

Outra Lista com princípio:

o promeiro a entrar é o primeiro a sair

First In First Out(FIFO)

Operções de inserção(enqueue) sempre ao final e retirada(dequeue) sempre no inicio da lista.

ALVES, 2009

Fila

Para uma fila F, são definidas as operações:

enqueue(x), insere o elemento X na fila
dequeue(), retira um elemento da fila

Operação	saída	F(inicio,fim)
enqueue(5)	-	(5)
enqueue(6)	-	(5,6)
dequeue()	5	(6)
enqueue(3)	-	(6,3)
enqueue(4)	-	(6,3,4)
dequeue()	6	(3,4)
dequeue()	3	(4)
dequeue()	4	()
dequeue()	erro	()

Filas

Exemplo prático

Fila

```
public class Fila {
    double [] v = new double[10];
    int primeiro;
    int ultimo;

    public void enqueue(double elemento) {
        int m = v.length;

        if (ultimo != m-1) {
           ultimo++;
           v[ultimo] = elemento;
        } else {
           System.out.println("Status: Lote Cheio");
        }
    }
```

```
public double dequeue() {
        double elemento;

        if (primeiro <= ultimo) {
            elemento = v[primeiro];
            primeiro++;
            if (primeiro > ultimo) {
                primeiro = 0;
                ultimo = -1;
            }
            return(elemento);
        } else {
            return(-1);
        }
    }
```

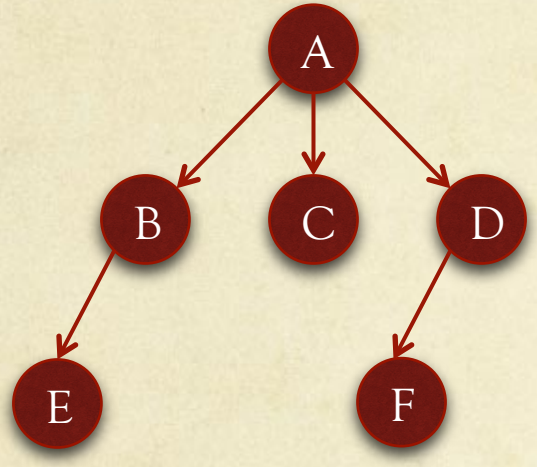
Árvore

Na Árvore a relação existente entre os dados (denominados nós) é uma relação de hierarquia, onde um conjunto de dados é hierarquicamente subordinado a outro. É uma lista não linear.

Uma Árvore é um conjunto finito T de um ou mais nós, tais que:

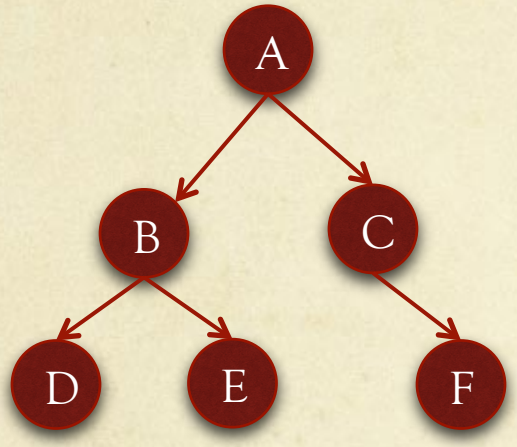
- existe um nó raiz da árvore
- os demais nós formam conjuntos disjuntos, onde cada um destes
- conjuntos é uma árvore (denominada de sub-árvores)

Árvore



- número de subárvores de um nó é o grau daquele nó;
- cada nó da Árvore é a raiz de uma subárvore, os nós inferiores são denominados “filhos”;
- um nó de grau zero é denominado folha ou nó terminal;
- nível do nó: é igual ao número de "linhas" que o liga à raiz (a raiz tem nível zero);
- altura da Árvore é definida como sendo o nível mais alto da Árvore;
- floresta é um conjunto de zero ou mais Árvores disjuntas (se eliminarmos o nó raiz de uma árvore, o que dela restar forma uma floresta);

Árvore Binária

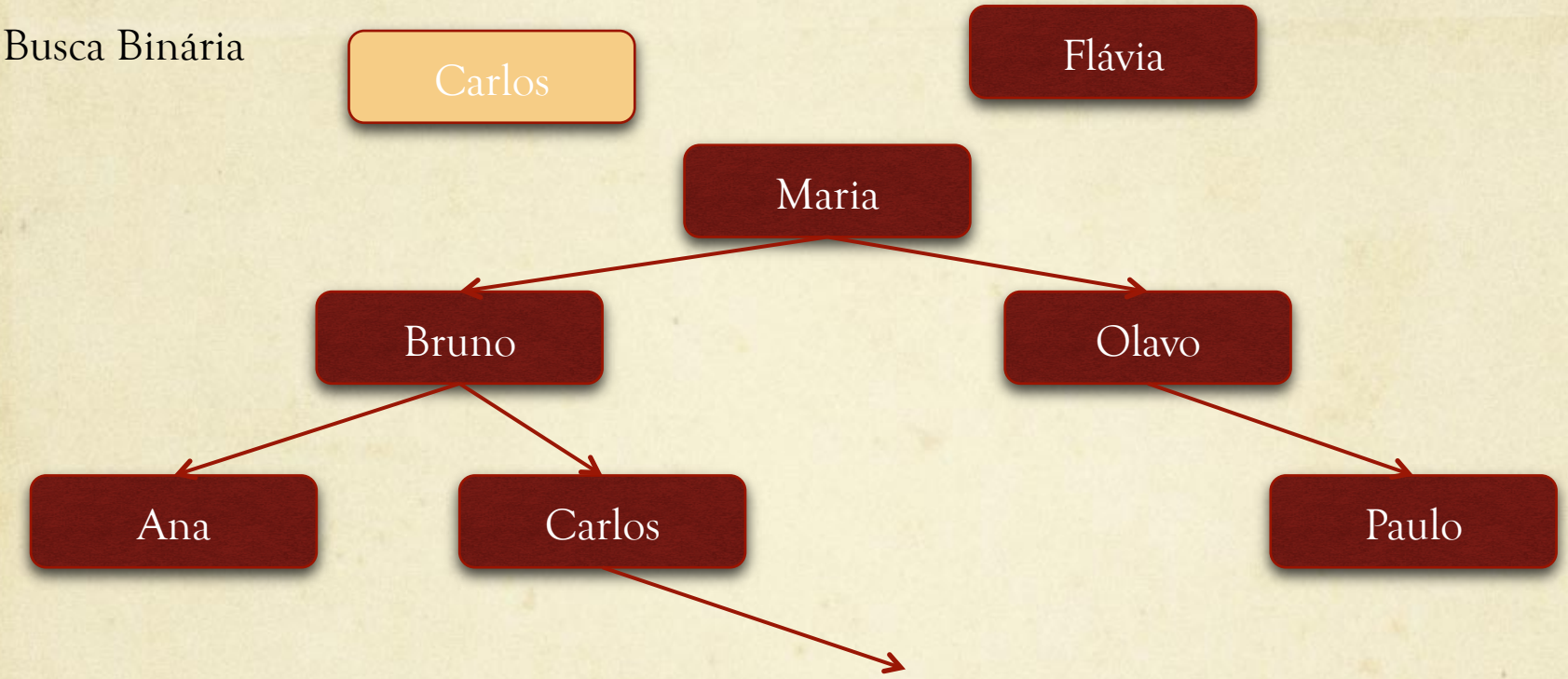


- Árvore de grau 2
- Muito utilizada em buscas
- Índices para SGBD's

ALVES, 2009;

Árvore Binária

Busca Binária



Ordenação de Dados

Umas das funções mais importantes dentro da estruturas de dados devido a grande utilização destes métodos para organização de índices. Existem vários métodos, por exemplo:

Bubble Sort

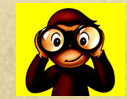
Inserção Direta

QuickSort

ALVES, 2009; BOENTE, 2003; GOODRICH, 2002

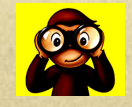
Bubble Sort

O *bubble sort*, ou ordenação por flutuação (literalmente "por bolha"), é um algoritmo de ordenação dos mais simples. A ideia é percorrer a lista diversas vezes, a cada passagem fazendo flutuar para o topo o maior elemento da sequência. Essa movimentação lembra a forma como as bolhas em um tanque de água procuram seu próprio nível, e disso vem o nome do algoritmo.



http://pt.wikipedia.org/wiki/Bubble_sort

Bubble Sort



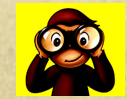
http://www.youtube.com/watch?v=rCKbfdvnhxI

Quick Sort

- O algoritmo **Quicksort** é um método de ordenação muito rápido e eficiente, inventado por C.A.R. Hoare em 1960. Ele criou o '*Quicksort* ao tentar traduzir um dicionário de inglês para russo, ordenando as palavras, tendo como objetivo reduzir o problema original em subproblemas que possam ser resolvidos mais fácil e rapidamente. Foi publicado em 1962 após uma série de refinamentos.

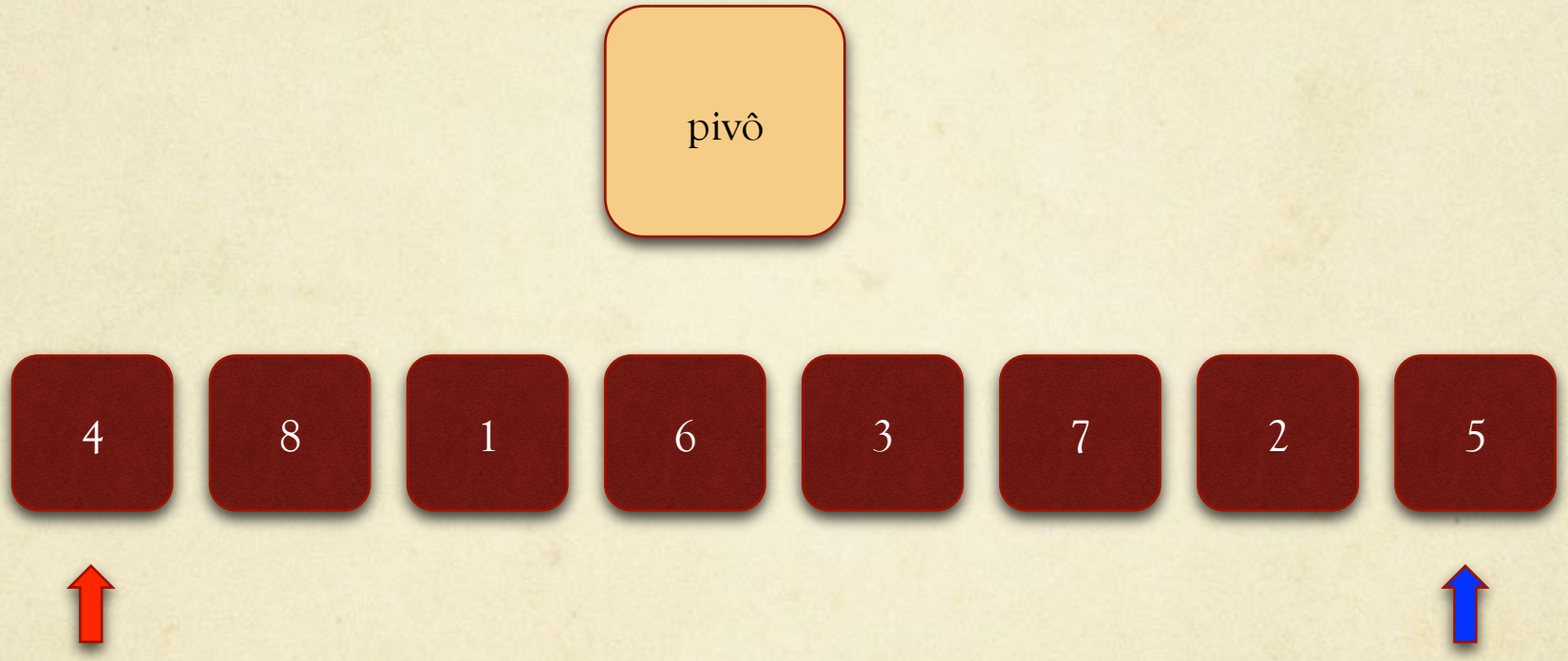
Quick Sort

O Quicksort adota a estratégia de divisão e conquista. A estratégia consiste em rearranjar as chaves de modo que as chaves "menores" precedam as chaves "maiores". Em seguida o Quicksort ordena as duas sublistas de chaves menores e maiores recursivamente até que a lista completa se encontre ordenada.

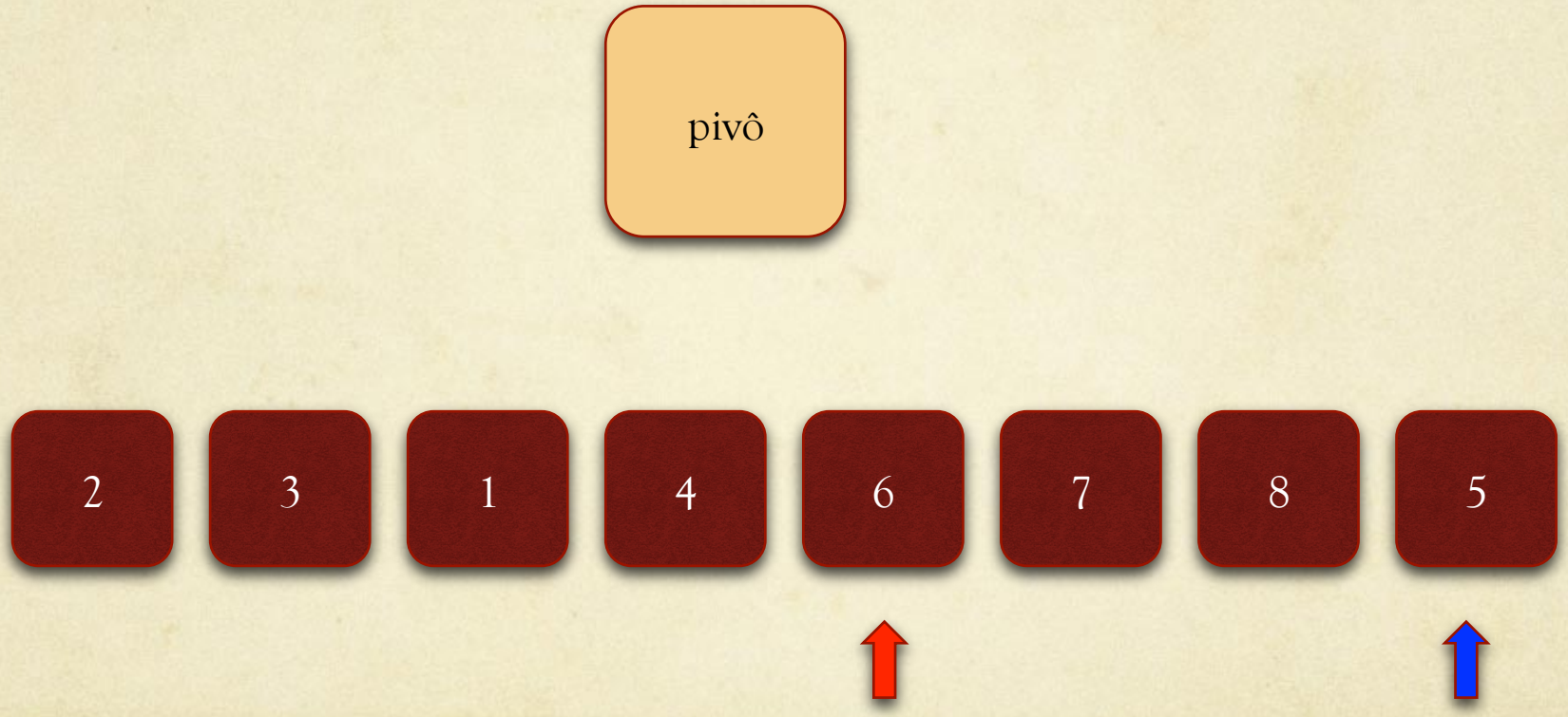


http://www.youtube.com/watch?v=vxENKlcs2Tw

Quick Sort



Quick Sort



Bibliografia

- ALVES, W.P. Banco de Dados – Teoria e Desenvolvimento.Editora Érica, 2009.
- BOENTE, A. Construindo algoritmos computacionais. Brasoft Livros e Multimidia Ltda., Rio de Janeiro(Brasil), 2003.
- CORMEN, Thomaz H., LEISERSON, Charles E., RIVEST, Ronald, L., STEIN, Clifford. Introduction to algorithms. Prentice Hall, 2001.
- ZIVIANI, Nivio. Projeto de Algoritmos: Com Implementações em Pascal e C. São Paulo: Pioneira Thomson Learning, 2002.
- AHO, Alfred V., HOPCROFT, John F., UILMAN, Jeffrey D. Data Structure and Algorithms. Massachussetts:
- GOODRICH, Michael T., TAMASSIA, Robert. Estruturas de Dados e Algoritmos em Java.Porto Alegre: Bookman, 2002.

” that's all folks! ”