

FUNDAMENTOS MATEMÁTICOS PARA COMPUTAÇÃO

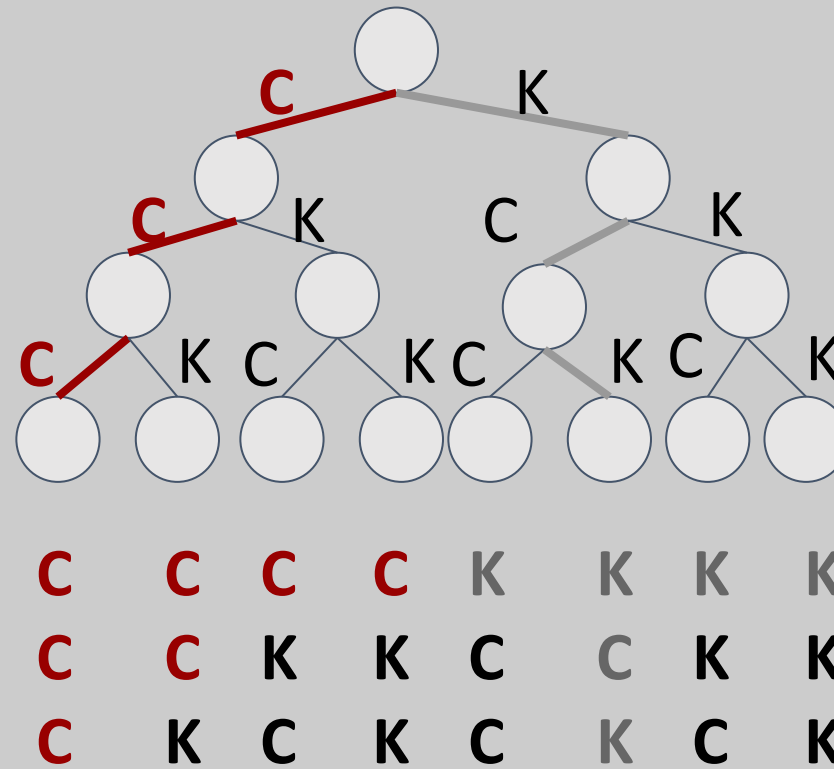
Árvores de decisão

SUMÁRIO

- Algoritmos de Busca
- Árvore de Busca Binária
- Algoritmos de Ordenação

Algoritmos de Busca

Árvore de decisão:
árvore na qual os nós
internos representam
ações, os arcos
representam os
resultados de uma
ação, e as folhas
representam
resultados finais.



Algoritmos de Busca

Busca Sequencial: Algoritmo mais simples de busca

1. Percorra toda a lista de valores comparando o valor procurado em cada posição.
2. Se for igual em alguma posição, então retorna a posição.
3. Se a lista foi toda percorrida e o valor não foi encontrado, retorna valor não encontrado.

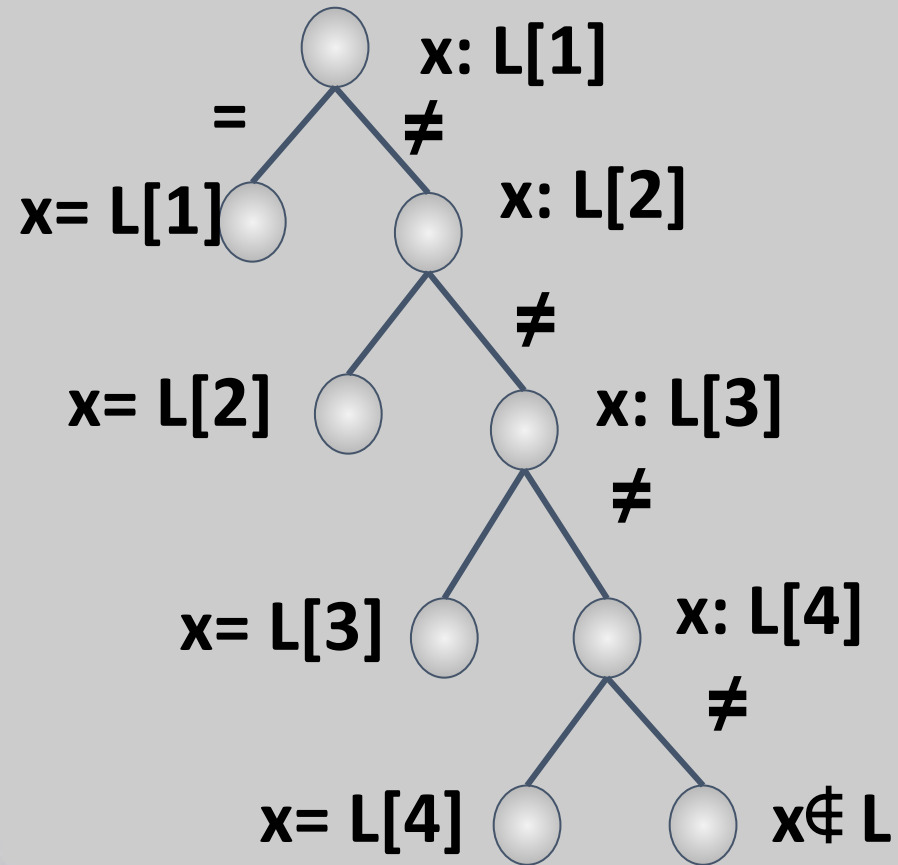
Algoritmos de Busca

Algoritmo BuscaSequencial(lista L de valores, tamanho n, valor x da busca)

- 1. Para i: 1 até n faça**
- 2. Se $L[i] == x$**
- 3. retorna i**
- 4. retorna $x \notin L$**

Algoritmos de Busca

Exemplo:



Algoritmos de Busca

Busca Binária: age em uma lista L ordenada com três resultados possíveis na busca por x :

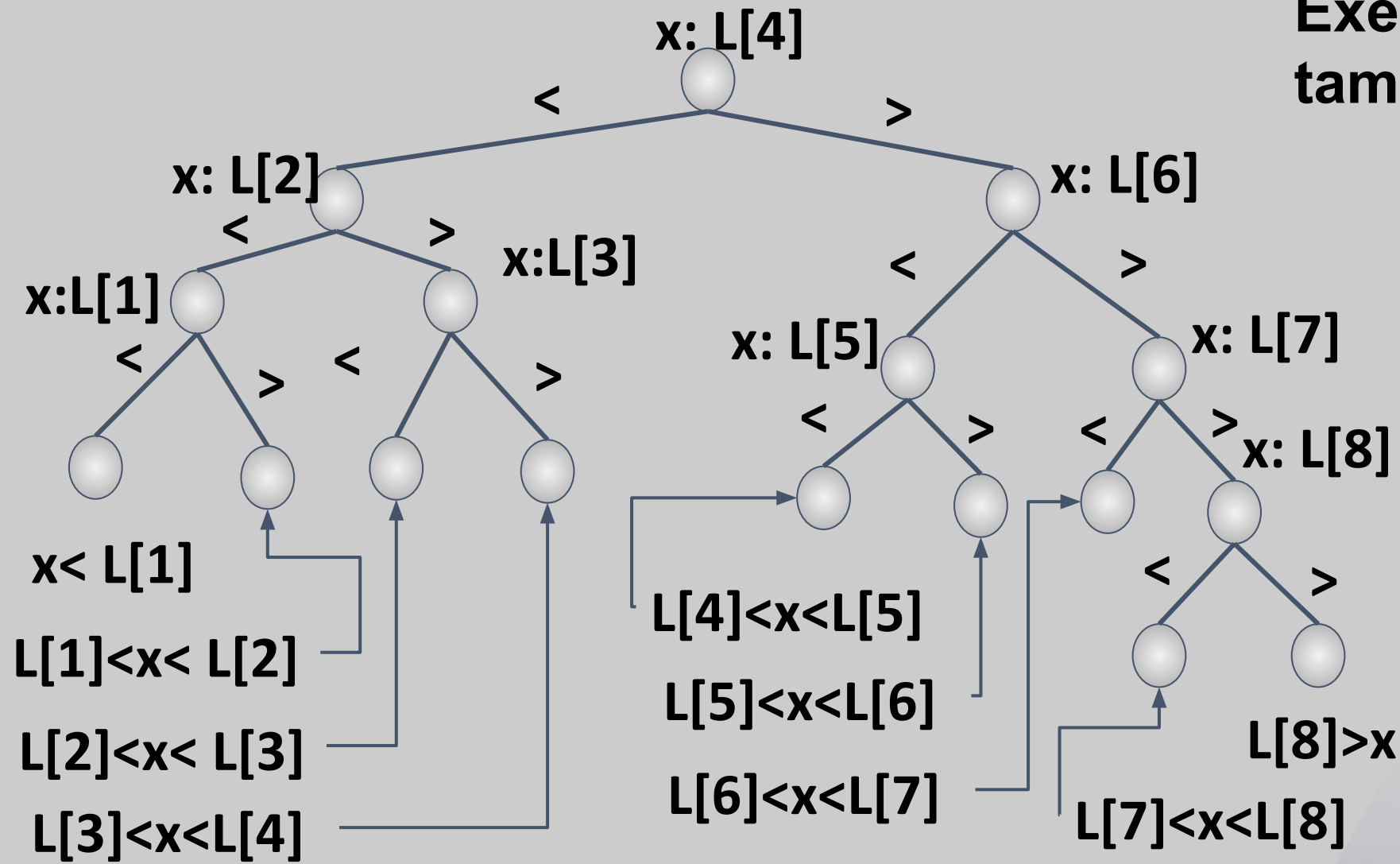
- $x = L[i]$: termina.
- $x < L[i]$: busca na metade esquerda de L
- $x > L[i]$: busca na metade direita de L

Algoritmos de Busca

Algoritmo BuscaBinária(lista L **ordenada**, tamanho n,
valor x da busca)

1. posInicial=0, posFinal=tam-1, posTemp;
2. Enquanto(posInicial <= posFinal) faça
3. posTemp = (posInicial+posFinal)/2
4. Se(L[posTemp] == x), então
5. retorne posTemp
6. Senão Se(L[posTemp] > x), então
7. posFinal = posTemp - 1
8. Senão
9. posInicial = posTemp + 1
10. retorne $x \notin L$

Exemplo: L com tamanho 7



Árvore Binária de Busca

Dados arbitrários podem ser organizados em uma estrutura chamada de **árvore binária de busca**, que pode ser pesquisada usando o algoritmo de busca em árvore binária

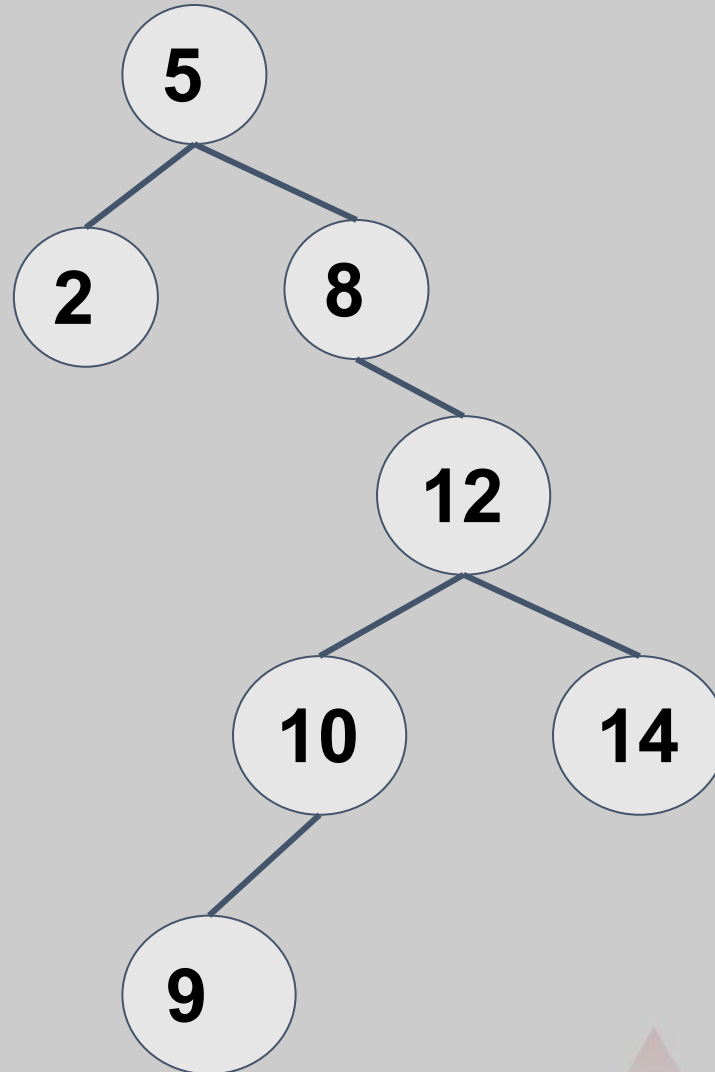
Árvore Binária de Busca

Construção da árvore binária de busca.

1. O primeiro dado é a raiz da árvore.
2. Novos dados são inseridos comparando-os com os nós já existentes, a começar pela raiz.
3. Se um dado for menor do que um nó, o próximo nó a ser testado é o filho esquerdo.
4. Senão, o filho direito é testado.
5. Quando o nó não tem filho, o novo dado torna-se um filho.

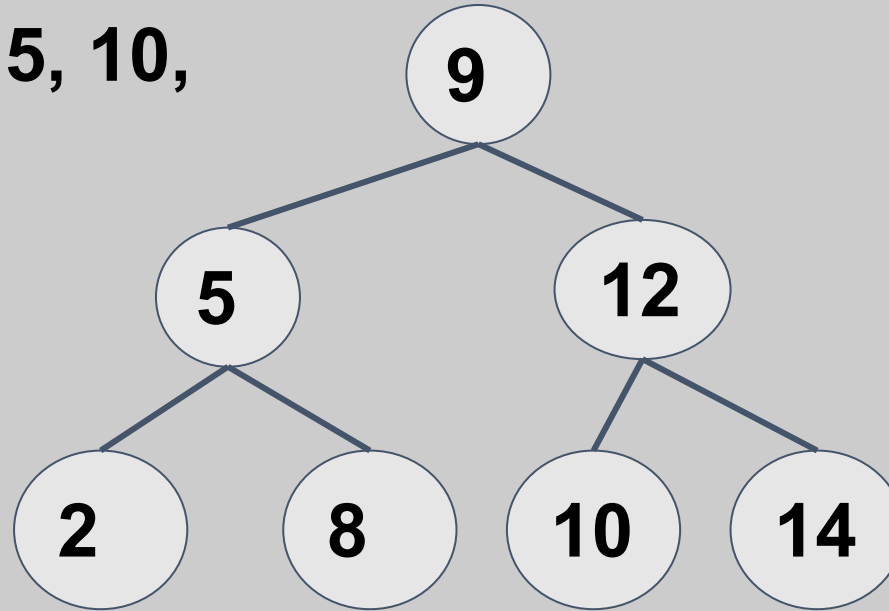
Árvore Binária de Busca

Exemplo: 5, 8, 2, 12,
14, 10, 9



Árvore Binária de Busca

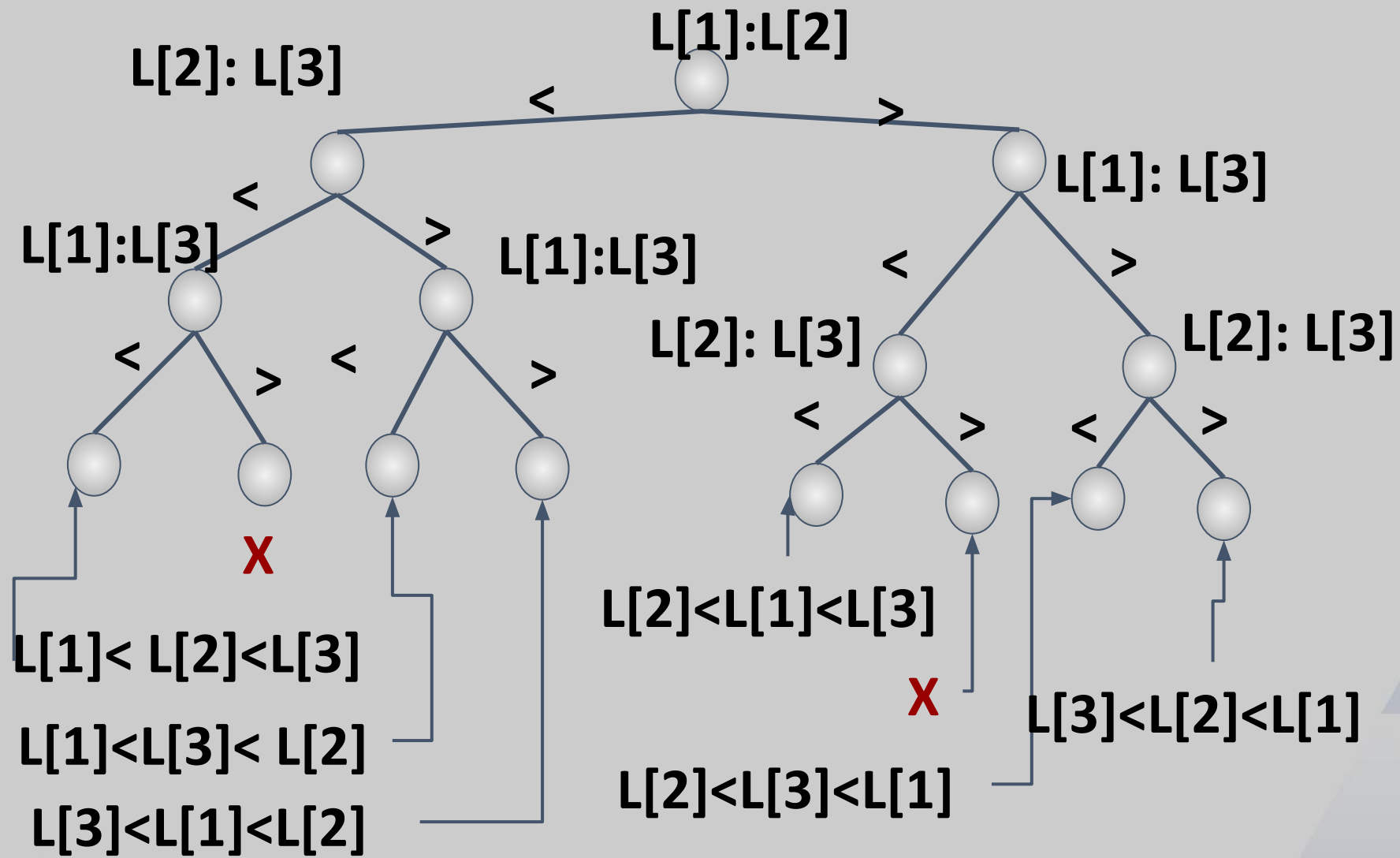
Exemplo: 9, 12, 5, 10,
8, 2, 14



Algoritmo de Ordenação

Podem ser modelados por árvores de decisão.
Supor que não há dados duplicados.

1. Se $L[i] < L[j]$, o algoritmo prossegue para a comparação indicada pelo filho esquerdo desse nó.
2. Senão, se $L[i] > L[j]$, o algoritmo vai para o filho direito.



Algoritmo de Ordenação

Teorema: Qualquer algoritmo que ordena uma lista de n elementos comparando pares de elementos na lista tem que fazer, pelo menos, $\lceil \log n! \rceil$ comparações no pior caso.

Prova:

As combinações possíveis de resultados nos nós folhas para n elementos são $n!$

O último nível da árvore tem 2^d nós folhas, onde d é a altura da árvore.

Seja p o número de folhas, temos:

$$p \leq 2^d$$

Algoritmo de Ordenação

Prova:

As combinações possíveis de resultados nos nós folhas para n elementos são $n!$

O último nível da árvore tem 2^d nós folhas, onde d é a altura da árvore.

Seja p o número de folhas, temos: $p \geq n!$

No pior caso, a ordenação é obtida no tamanho da árvore. Logo, $p \leq 2^d$

$$\log(p) \leq \log(2^d)$$

$$\log(p) \leq d$$

$$d \geq \log(p) \geq \log(n!)$$

Os conceitos e exemplos apresentados nesses slides são baseados no conteúdo da seção 6.4 do material-base “Fundamentos Matemáticos para a Ciência da Computação”, J.L. Gersting, 7a edição, LTC editora.

FUNDAMENTOS MATEMÁTICOS PARA COMPUTAÇÃO

Árvores de decisão