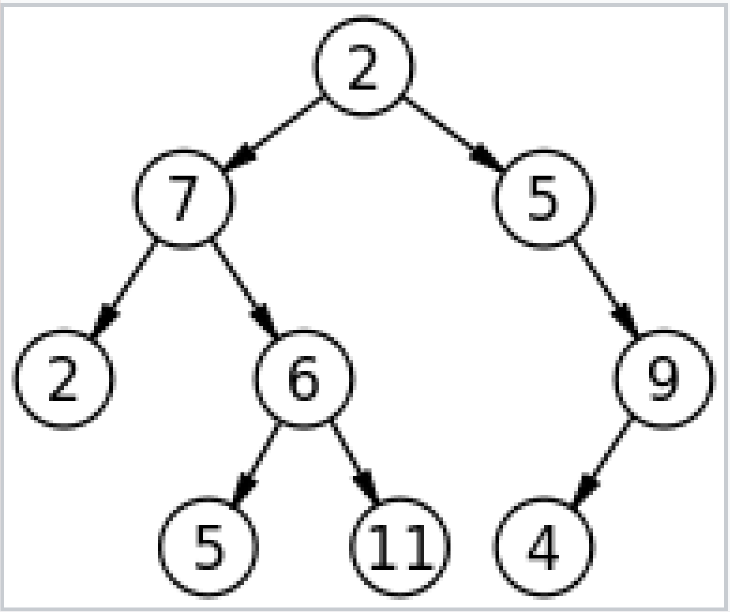
Binary Trees

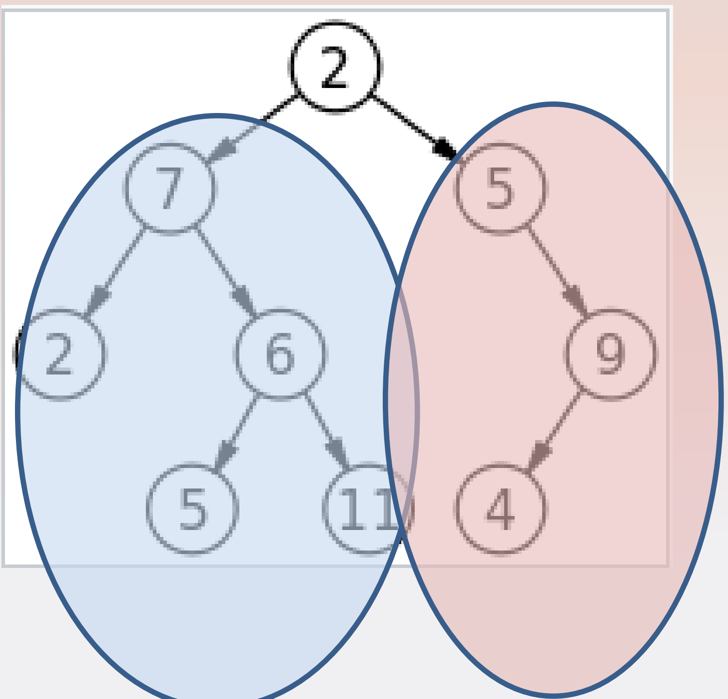
In computer science, a binary tree is a tree data structure, in which each node has at most two children.



Binary Trees

The number of leaves in a binary tree of height **h** is at most $\mathbf{2^h}$.

T



Proof (by induction on h). (l - num. of leaves)

Let T be a tree of height h.

Base Case: h = 0, tree has 1 leaf, and $l \leq 2^0 = 1$

Ind. assumption: for any $h \leq k$: $l \leq 2^k$

Prove that if $h = k + 1$, $\mathbf{l \leq 2^{k+1}}$ **.**

The number of leaves in T is equal to the sum of the number of leaves in blue and red subtrees:

$$l = l_b + l_r$$

Each of the subtrees has the height h = k, and by assumption

$$l_b \leq 2^k \text{ and } l_r \leq 2^k$$

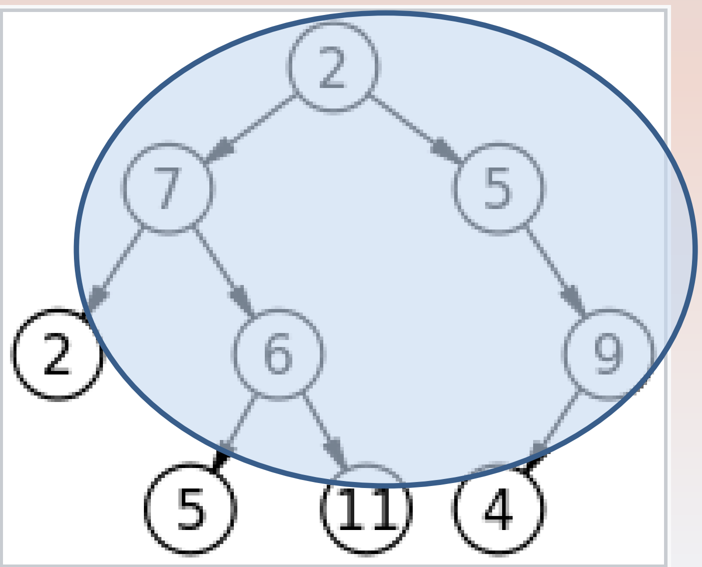
Then $\quad l \leq 2^k + 2^k = 2*2^k = 2^{k+1},$

$$l \leq 2^{k+1}$$

Binary Trees

A binary tree of height **h** has at most $\mathbf{2^{h+1}} - 1$ nodes

T



Proof (by induction on h).

Let T be a tree of height h.

Base Case: $h = 0$, tree has 1 node, and $n = 2^{0+1} - 1 = 1$

Ind. assumption: for any $h \leq k$: $n \leq 2^{k+1} - 1$

Prove that if $h = k + 1$, $\mathbf{n \leq 2^{k+2} - 1}$ **.**

The number of nodes in T is equal to the sum of all the leaf nodes and all the non-leaf nodes:

$$n = l + n'$$

By removing all leaf nodes, we get a tree of height $h = k$ with $n' \leq 2^{k+1} - 1$ nodes (by assumption).

From the prev. theorem, the number of leaves in T

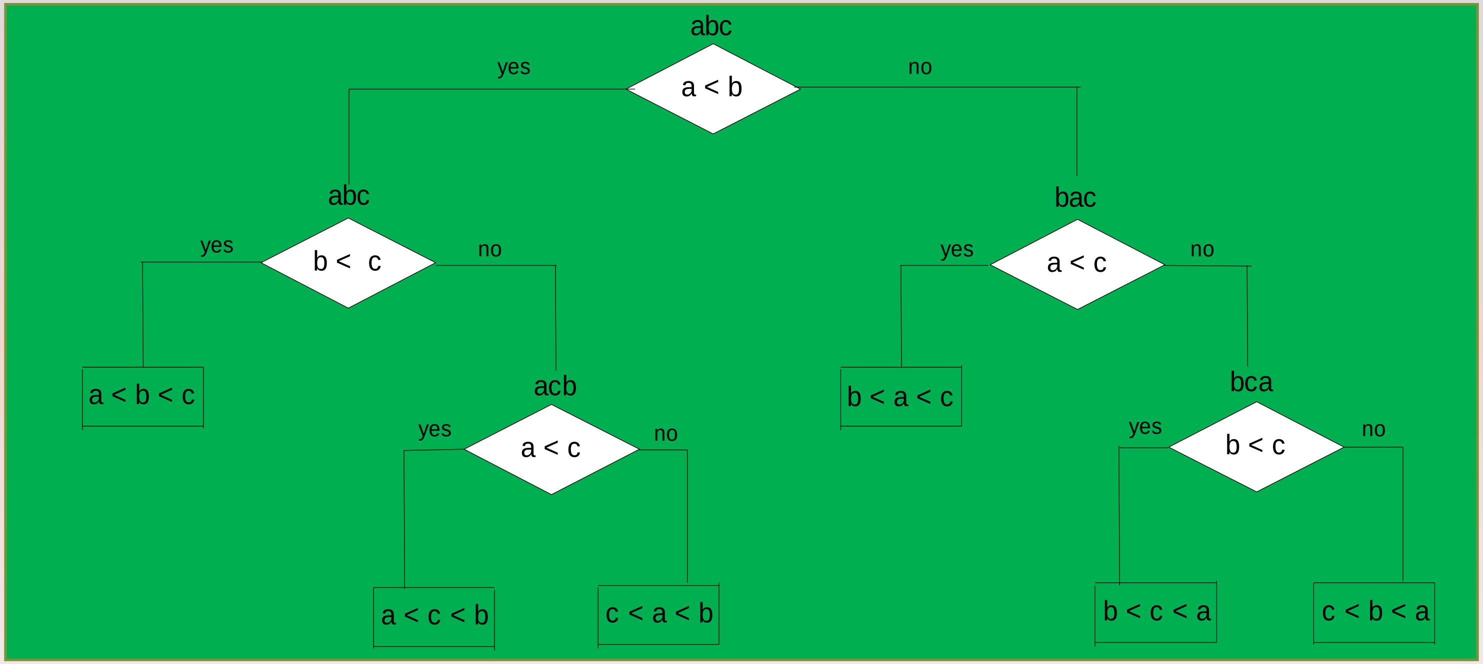
$$l \leq 2^{k+1}$$

Then $\quad n \leq 2^{k+1} + 2^{k+1} - 1 = 2 * 2^{k+1} - 1 = 2^{k+2} - 1$

$$\mathbf{n \leq 2^{k+2} - 1}$$

Decision Trees

Decision tree for 3-element insertion sort



Decision Trees

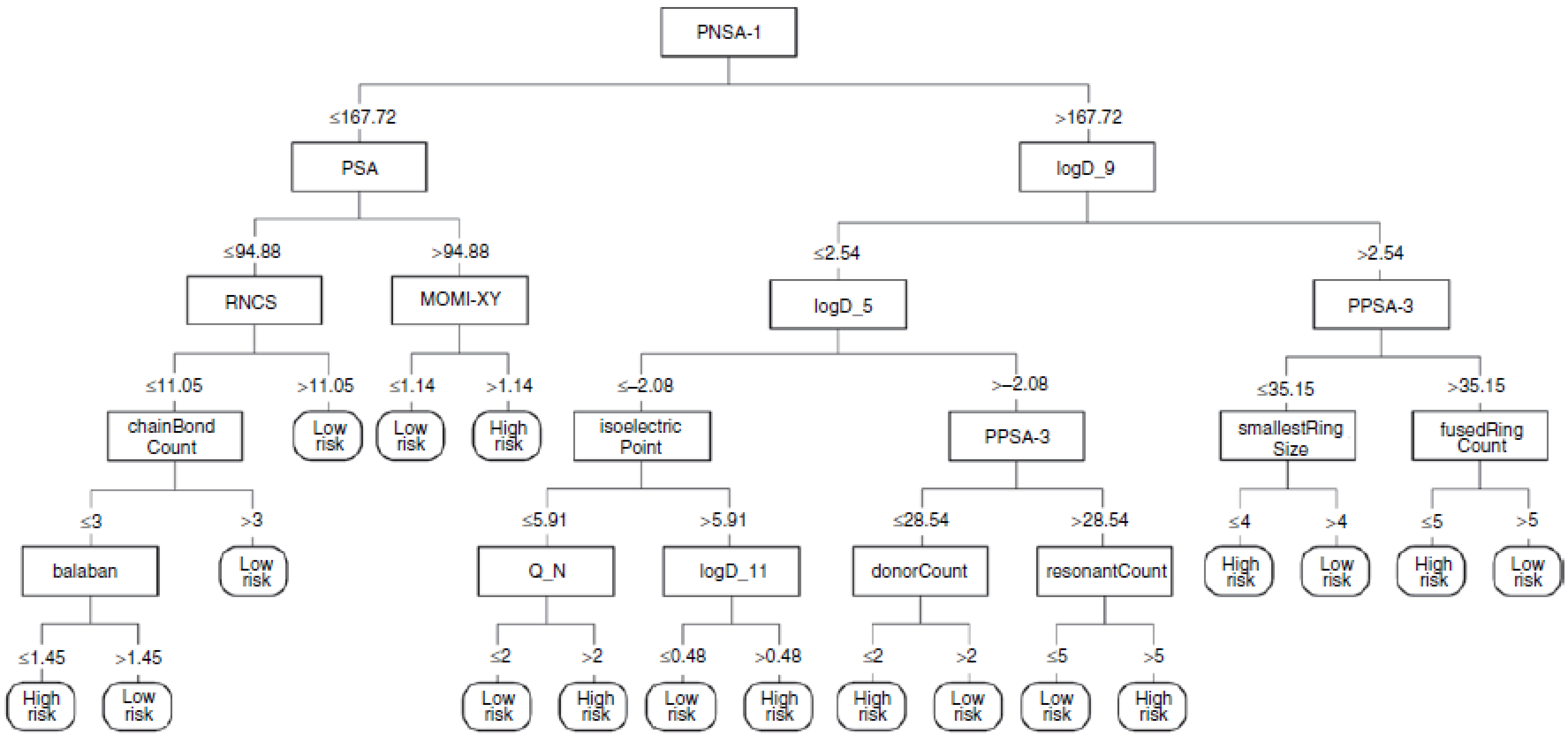
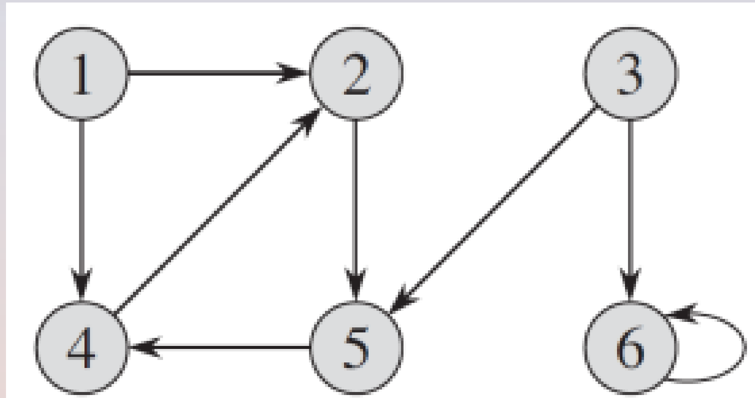
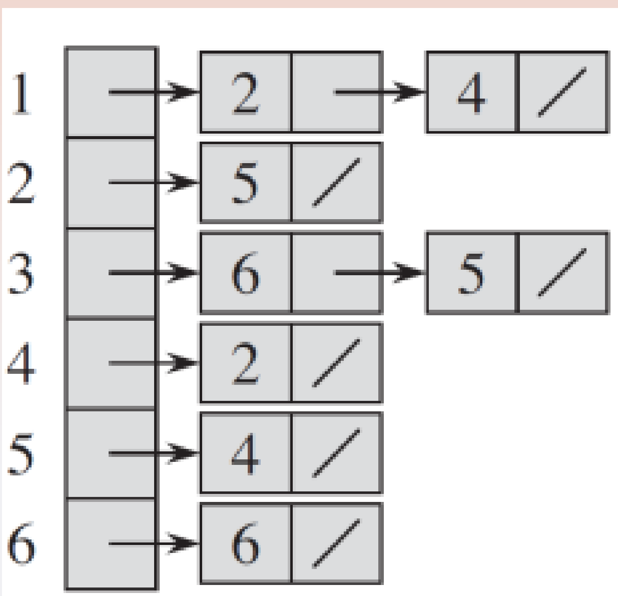


Figure 2 CART (classification and regression tree) decision tree model for the hepatic class of adverse drug reactions for 109 active and 177 inactive compounds ($n = 286$), with a corrected classification rate of 90.22%.

Graph Representation



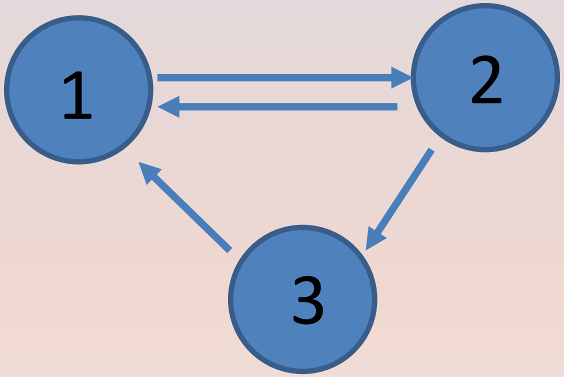
Adjacency list



Adjacency matrix

$$
\begin{array}{c|cccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 & 1 \\
4 & 0 & 1 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
\end{array}
$$

Adjacency matrix



$$M = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

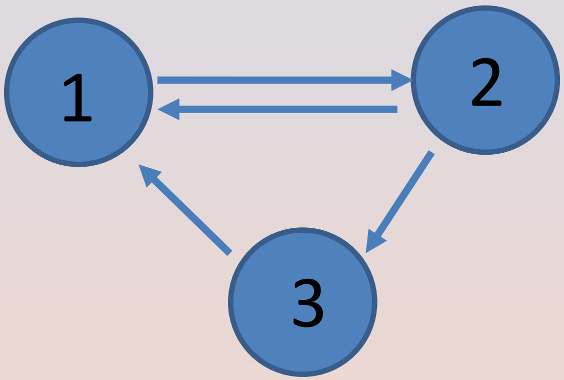
Adjacency matrix

- If you multiply an adjacency matrix A by itself, you get $\mathrm{A} \times A$ or A^2.

- A given entry a_{ij} from the matrix A^2 gives the number of simple paths of length 2 from node i to node j.

- More generally, an entry a_{ij} of A^k gives the number of simple paths of length exactly k from node i to node j.

Adjacency matrix



$$M = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

$$M^2 = \begin{pmatrix} \boxed{0 \quad 1 \quad 0} \\ 1 \quad 0 \quad 1 \\ 1 \quad 0 \quad 0 \end{pmatrix} \begin{pmatrix} \boxed{0} & 1 & 0 \\ \boxed{1} & 0 & 1 \\ \boxed{1} & 0 & 0 \end{pmatrix} = \begin{pmatrix} \boxed{1} & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$M^3 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$