11 Trees

11.2 Applications of Trees

1. Binary Search Trees presents an algorithm to searching for items in a list (one of the most important tasks that arises in computer science)

   - each child of a vertex is designated as a right or left child
   - no vertex has more than one right child or left child
   - each vertex is labeled with a key, which is one of the items
   - the key of a vertex is both larger than the keys of all vertices in its left subtree and smaller than the keys of all vertices in its right subtree.

ALGORITHM 1 Locating an Item in or Adding an Item to a Binary Search Tree.

procedure *insertion*(T: binary search tree, x: item)
v := root of T
{a vertex not present in T has the value *null*}
while $v \neq$ *null* and *label*(v) $\neq x$
  if $x <$ *label*(v) **then**
    if left child of $v \neq$ *null* **then** v := left child of v
    else add *new vertex* as a left child of v and set v := *null*
  else
    if right child of $v \neq$ *null* **then** v := right child of v
    else add *new vertex* as a right child of v and set v := *null*
if root of T = *null* **then** add a vertex v to the tree and label it with x
else if v is null or *label*(v) $\neq x$ **then** label *new vertex* with x and let v be this new vertex
return v {v = location of x}

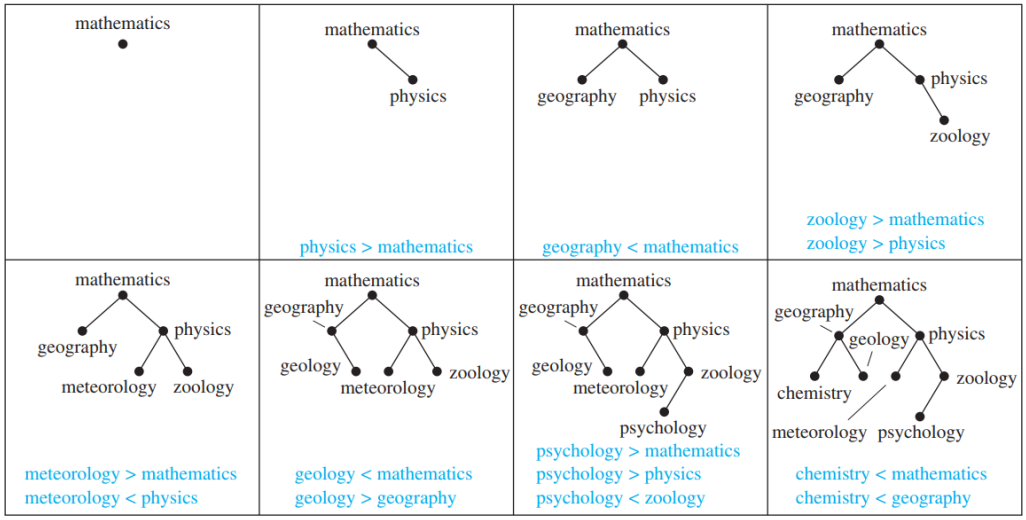

FIGURE 1 Constructing a Binary Search Tree.

2. Prefix Codes use bit strings to encode the letters of the English alphabet (where no distinction is made between lowercase and uppercase letters) represent each letter with a bit string of length five, because there are only 26 letters and there are 32 bit strings of length five.

   Note that prefix codes are codes for which

   - each letter has a unique binary string that represents it and
   - no bit string corresponds to more than one sequence of letters (so the bit string for a letter never occurs as the first part of the bit string for another letter).

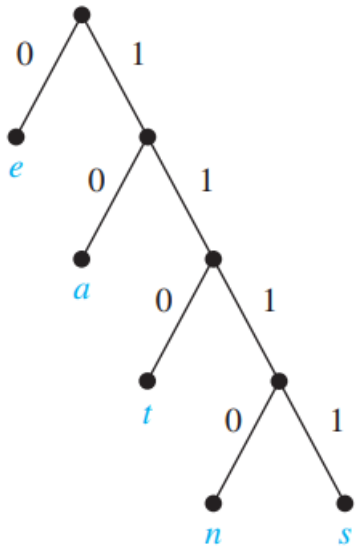

Figure 1: Example Binary Tree with a prefix code

This bit string can be decoded to a letter by starting at the root, using the sequence of bits to form a path that stops when a leaf is reached. For example letter a is the code 10. What words do these two strings represent:

110010

111110110

110101110

3. Huffman Coding is an algorithm that takes as input the frequencies (which are the probabilities of occurrences) of symbols in a string and produces as output a prefix code that encodes the string using the fewest possible bits, among all possible binary prefix codes for these symbols:

ALGORITHM 2 Huffman Coding.

procedure *Huffman*(C: symbols a_i with frequencies w_i, $i = 1, \ldots, n$)
$F :=$ forest of n rooted trees, each consisting of the single vertex a_i and assigned weight w_i
while F is not a tree
  Replace the rooted trees T and T' of least weights from F with $w(T) \geq w(T')$ with a tree having a new root that has T as its left subtree and T' as its right subtree. Label the new edge to T with 0 and the new edge to T' with 1.
  Assign $w(T) + w(T')$ as the weight of the new tree.
{the Huffman coding for the symbol a_i is the concatenation of the labels of the edges in the unique path from the root to the vertex a_i}

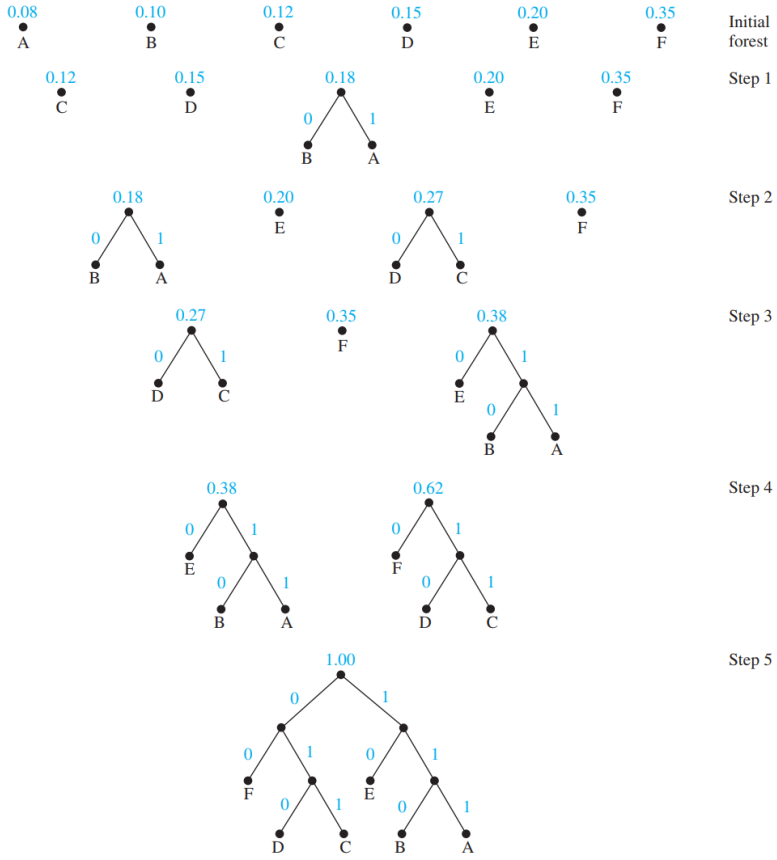

FIGURE 6 Huffman Coding of Symbols in Example 4.