Key Terms and Results

TERMS

tree: a connected undirected graph with no simple circuits

forest: an undirected graph with no simple circuits

rooted tree: a directed graph with a specified vertex, called the root, such that there is a unique path to every other vertex from this root

subtree: a subgraph of a tree that is also a tree

parent of v in a rooted tree: the vertex u such that (u, v) is an edge of the rooted tree

child of a vertex v in a rooted tree: any vertex with v as its parent

sibling of a vertex v in a rooted tree: a vertex with the same parent as v

ancestor of a vertex v in a rooted tree: any vertex on the path from the root to v

descendant of a vertex v in a rooted tree: any vertex that has v as an ancestor

internal vertex: a vertex that has children

leaf: a vertex with no children

level of a vertex: the length of the path from the root to this vertex

height of a tree: the largest level of the vertices of a tree

m-ary tree: a tree with the property that every internal vertex has no more than m children

full m-ary tree: a tree with the property that every internal vertex has exactly m children

binary tree: an m-ary tree with $m = 2$ (each child may be designated as a left or a right child of its parent)

ordered tree: a tree in which the children of each internal vertex are linearly ordered

balanced tree: a tree in which every leaf is at level h or $h - 1$, where h is the height of the tree

binary search tree: a binary tree in which the vertices are labeled with items so that a label of a vertex is greater than the labels of all vertices in the left subtree of this vertex and is less than the labels of all vertices in the right subtree of this vertex

decision tree: a rooted tree where each vertex represents a possible outcome of a decision and the leaves represent the possible solutions of a problem

game tree: a rooted tree where vertices represent the possible positions of a game as it progresses and edges represent legal moves between these positions

prefix code: a code that has the property that the code of a character is never a prefix of the code of another character

minmax strategy: the strategy where the first player and second player move to positions represented by a child with maximum and minimum value, respectively

value of a vertex in a game tree: for a leaf, the payoff to the first player when the game terminates in the position represented by this leaf; for an internal vertex, the maximum or minimum of the values of its children, for an internal vertex at an even or odd level, respectively

tree traversal: a listing of the vertices of a tree

preorder traversal: a listing of the vertices of an ordered rooted tree defined recursively—the root is listed, followed by the first subtree, followed by the other in the order they occur from left to right

inorder traversal: a listing of the vertices of an ordered rooted tree defined recursively—the first subtree is listed, followed by the root, followed by the other subtrees in the order they occur from left to right

postorder traversal: a listing of the vertices of an ordered rooted tree defined recursively—the subtrees are listed in the order they occur from left to right, followed by the root

infix notation: the form of an expression (including a full set of parentheses) obtained from an inorder traversal of the binary tree representing this expression

prefix (or Polish) notation: the form of an expression obtained from a preorder traversal of the tree representing this expression

postfix (or reverse Polish) notation: the form of an expression obtained from a postorder traversal of the tree representing this expression

spanning tree: a tree containing all vertices of a graph

minimum spanning tree: a spanning tree with smallest possible sum of weights of its edges

RESULTS

A graph is a tree if and only if there is a unique simple path between every pair of its vertices.

A tree with n vertices has $n - 1$ edges.

A full m-ary tree with i internal vertices has $mi + 1$ vertices.

The relationships among the numbers of vertices, leaves, and internal vertices in a full m-ary tree (see Theorem 4 in Section 11.1)

There are at most m^h leaves in an m-ary tree of height h.

If an m-ary tree has l leaves, its height h is at least $\lceil \log_m l \rceil$. If the tree is also full and balanced, then its height is $\lceil \log_m l \rceil$.

Huffman coding: a procedure for constructing an optimal binary code for a set of symbols, given the frequencies of these symbols

depth-first search, or backtracking: a procedure for constructing a spanning tree by adding edges that form a path until this is not possible, and then moving back up the path until a vertex is found where a new path can be formed

breadth-first search: a procedure for constructing a spanning tree that successively adds all edges incident to the last set of edges added, unless a simple circuit is formed

Prim's algorithm: a procedure for producing a minimum spanning tree in a weighted graph that successively adds edges with minimal weight among all edges incident to a vertex already in the tree so that no edge produces a simple circuit when it is added

Kruskal's algorithm: a procedure for producing a minimum spanning tree in a weighted graph that successively adds edges of least weight that are not already in the tree such that no edge produces a simple circuit when it is added

Review Questions

1. a) Define a tree. b) Define a forest.
2. Can there be two different simple paths between the vertices of a tree?
3. Give at least three examples of how trees are used in modeling.
4. a) Define a rooted tree and the root of such a tree.
   b) Define the parent of a vertex and a child of a vertex in a rooted tree.
   c) What are an internal vertex, a leaf, and a subtree in a rooted tree?
   d) Draw a rooted tree with at least 10 vertices, where the degree of each vertex does not exceed 3. Identify the root, the parent of each vertex, the children of each vertex, the internal vertices, and the leaves.
5. a) How many edges does a tree with n vertices have?
   b) What do you need to know to determine the number of edges in a forest with n vertices?
6. a) Define a full m-ary tree.
   b) How many vertices does a full m-ary tree have if it has i internal vertices? How many leaves does the tree have?
7. a) What is the height of a rooted tree?
   b) What is a balanced tree?
   c) How many leaves can an m-ary tree of height h have?
8. a) What is a binary search tree?
   b) Describe an algorithm for constructing a binary search tree.
   c) Form a binary search tree for the words *vireo, warbler, egret, grosbeak, nuthatch*, and *kingfisher*.
9. a) What is a prefix code?
   b) How can a prefix code be represented by a binary tree?
10. a) Define preorder, inorder, and postorder tree traversal.
    b) Give an example of preorder, postorder, and inorder traversal of a binary tree of your choice with at least 12 vertices.
11. a) Explain how to use preorder, inorder, and postorder traversals to find the prefix, infix, and postfix forms of an arithmetic expression.
    b) Draw the ordered rooted tree that represents $((x - 3) + ((x/4) + (x - y) \uparrow 3))$.
    c) Find the prefix and postfix forms of the expression in part (b).
12. Show that the number of comparisons used by a sorting algorithm to sort a list of n elements is at least $\lceil \log n! \rceil$.
13. a) Describe the Huffman coding algorithm for constructing an optimal code for a set of symbols, given the frequency of these symbols.
    b) Use Huffman coding to find an optimal code for these symbols and frequencies: A: 0.2, B: 0.1, C: 0.3, D: 0.4.
14. Draw the game tree for nim if the starting position consists of two piles with one and four stones, respectively. Who wins the game if both players follow an optimal strategy?
15. a) What is a spanning tree of a simple graph?
    b) Which simple graphs have spanning trees?
    c) Describe at least two different applications that require that a spanning tree of a simple graph be found.
16. a) Describe two different algorithms for finding a spanning tree in a simple graph.
    b) Illustrate how the two algorithms you described in part (a) can be used to find the spanning tree of a simple graph, using a graph of your choice with at least eight vertices and 15 edges.
17. a) Explain how backtracking can be used to determine whether a simple graph can be colored using n colors.
    b) Show, with an example, how backtracking can be used to show that a graph with a chromatic number equal to 4 cannot be colored with three colors, but can be colored with four colors.
18. a) What is a minimum spanning tree of a connected weighted graph?
    b) Describe at least two different applications that require that a minimum spanning tree of a connected weighted graph be found.
19. a) Describe Kruskal's algorithm and Prim's algorithm for finding minimum spanning trees.
    b) Illustrate how Kruskal's algorithm and Prim's algorithm are used to find a minimum spanning tree, using a weighted graph with at least eight vertices and 15 edges.