

## Chapter 12 Trees

### Section 12.1

#### Definitions

tree, forest, spanning tree, spanning forest

#### Theorem

- 12.1  $T = (V, E)$  tree.  $\forall a, b \in V$  there is a unique path from  $a$  to  $b$ .
- 12.2  $G$  undirected graph.  $G$  connected  $\Leftrightarrow G$  has a spanning tree.
- 12.3  $T$  tree  $\Rightarrow |V| = |E| + 1$ .
- 12.4  $|V| \geq 2 \Rightarrow T$  at least 2 pendant vertices.
- 12.5 Five equivalent statements for a loop-free undirected graph  $G$ .
  - $G$  tree
  - $G$  connected. Removal of any edge disconnects  $G$  into two trees.
  - $G$  acyclic,  $|V| = |E| + 1$ .
  - $G$  connected,  $|V| = |E| + 1$ .
  - $G$  acyclic. Adding any edge, not in  $G$ , makes a cycle.

#### Exercises

8, 17, 18, 19

### Section 12.2

#### Definitions

directed tree, rooted tree, root, leaf, internal vertex, level, child, parent, descendants, ancestors, siblings, subtree, ordered rooted tree, universal address system, lexicographic order, binary rooted tree, complete binary rooted tree, infix, prefix, preorder transversal, postorder transversal, inorder transversal, backtrack, depth-first search, breadth-first search, queue, m-ary tree, hight, balanced

#### Theorem

- 12.6  $T$  complete  $m$ -ary tree,  $l$  leaves,  $i$  internal vertices.  
 $|V| = mi + 1$ ,  $l = (m - 1)i + 1$ , and  $i = \frac{l-1}{m-1} = \frac{n-1}{m}$ .
- 12.7  $T$  complete  $m$ -ary tree of hight  $h$ , then  $l \leq m^h$ .

#### Algorithms

Depth-first search algorithm  
Breadth-first search algorithm

#### Exercises

1, 3, 5, 9, 10, 13, 18

## **Section 11.3**

### **Algorithms**

Merge two lists (Lemma 12.1)  
Merge Sort Algorithm  
Quick Sort Algorithm

### **Exercises**

1, 2, 3

## **Section 11.4**

### **Definitions**

prefix code, full binary tree, the weight of a tree  $W(T)$ , Huffman Tree

### **Algorithms**

Huffman Tree Construction (Lemma 12.2, Theorem 12.8)

### **Exercises**

1, 3, 7

## **Section 11.5**

### **Definitions**

articulation point, biconnected, biconnected component,

### **Algorithm**

Search for biconnected components (Lemma 12.3)

### **Exercises**

1, 5, 6, 9