Trees

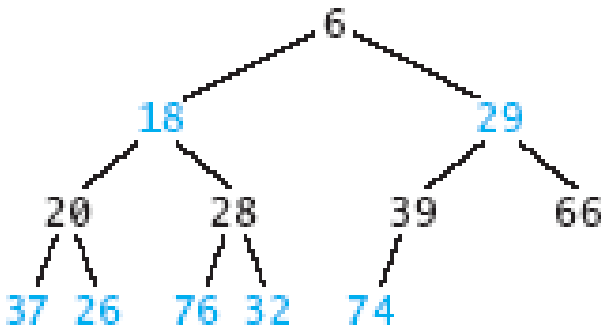
Chapter 4

(and some extra things)

Heaps and Priority Queues

- In a heap, the value in a node is less than all values in its two subtrees
- A heap is a complete binary tree with the following properties
  - The value in the root is the smallest item in the tree
  - Every subtree is a heap

FIGURE 8.20
Example of a Heap



Inserting an Item into a Heap

Algorithm for Inserting in a Heap

1. Insert the new item in the next position at the bottom of the heap.
2. `while` new item is not at the root and new item is smaller than its parent
3. Swap the new item with its parent, moving the new item up the heap.

FIGURE 8.21
Inserting 8 into a Heap

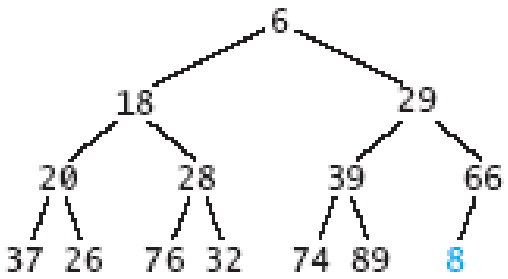


FIGURE 8.22
Swapping 8 and 66

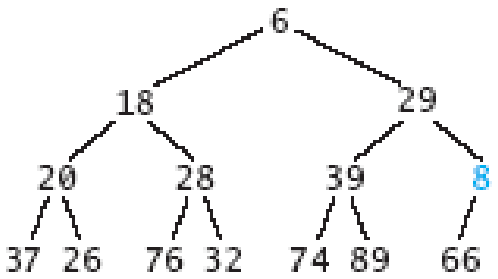
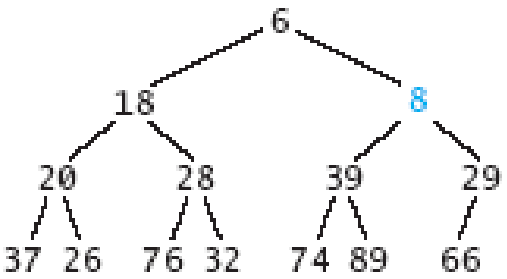


FIGURE 8.23
Swapping 8 and 29



Removing an Item from a Heap

Algorithm for Removal from a Heap

1. Remove the item in the root node by replacing it with the last item in the heap (LIH).
2. `while` item LIH has children and item LIH is larger than either of its children
3. Swap item LIH with its smaller child, moving LIH down the heap.

As an example, if we remove 6 from the heap shown in Figure 8.23, 66 replaces it as shown in Figure 8.24. Since 66 is larger than both of its children, it is swapped

FIGURE 8.24
After Removal of 6

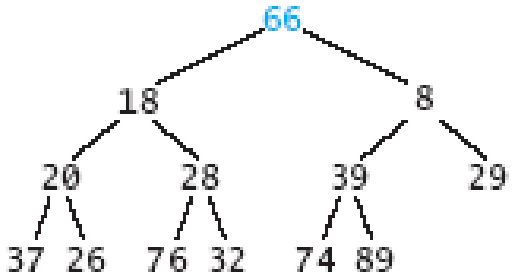


FIGURE 8.25
Swapping 66 and 8

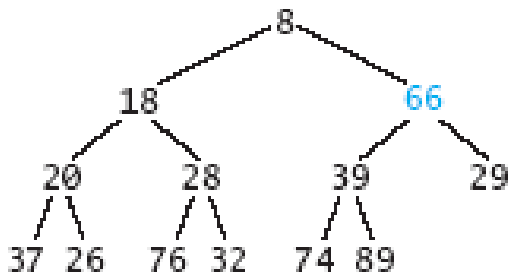
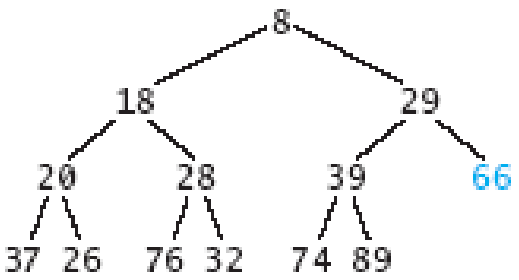


FIGURE 8.26
Swapping 66 and 29



Implementing a Heap

- Because a heap is a complete binary tree, it can be implemented efficiently using an array instead of a linked data structure
- First element for storing a reference to the root data
- Use next two elements for storing the two children of the root
- Use elements with subscripts 3, 4, 5, and 6 for storing the four children of these two nodes and so on

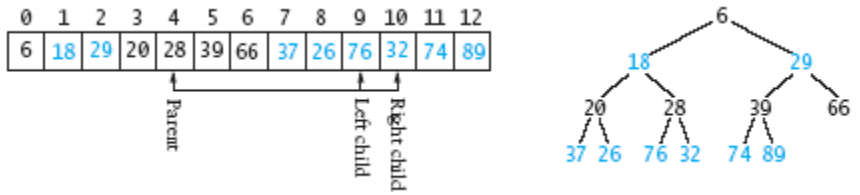


FIGURE 8.27
Internal Representation of the Heap

Inserting into a Heap Implemented as an ArrayList

FIGURE 8.28
Internal Representation of Heap After Insertion

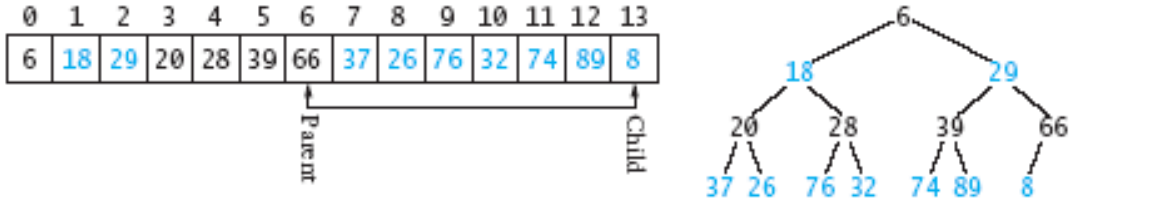


FIGURE 8.29
Internal Representation of Heap After First Swap

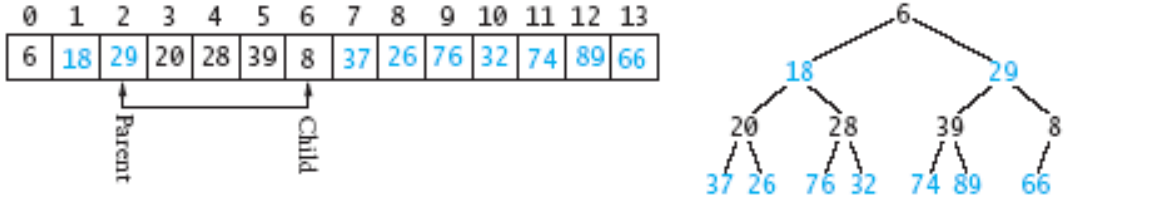
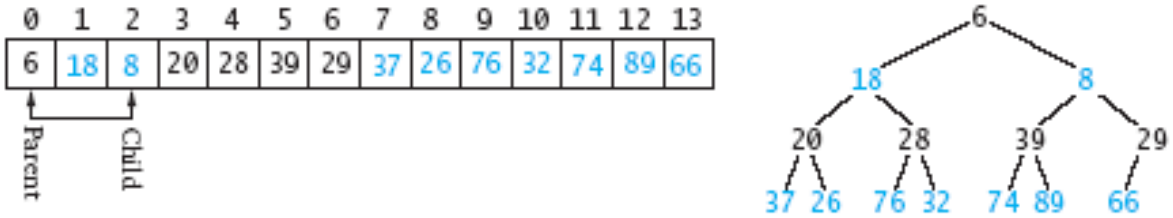


FIGURE 8.30
Internal Representation of Heap After Second Swap



Inserting into a Heap Implemented as an ArrayList (continued)

FIGURE 8.31
Internal Representation of Heap After 6 Is Removed

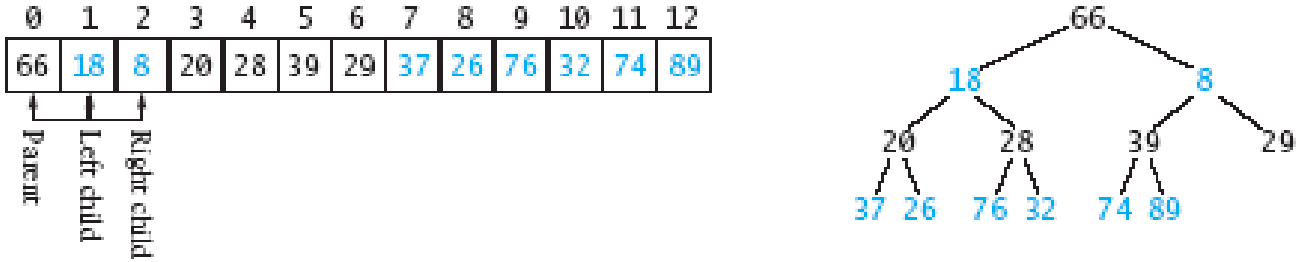


FIGURE 8.32
Internal Representation of Heap After 8 and 66 Are Swapped

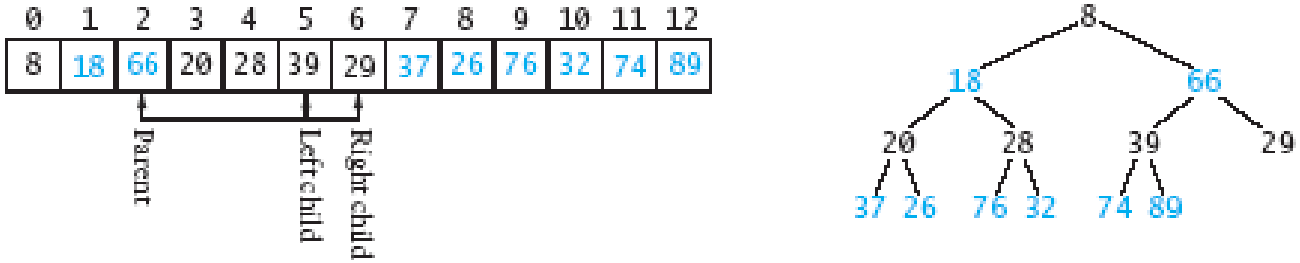
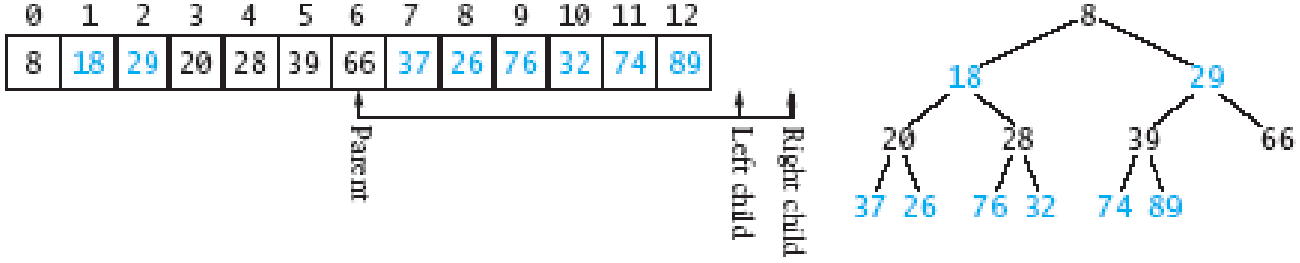


FIGURE 8.33
Internal Representation of Heap After Swap of 66 and 29



Priority Queues

- The heap is used to implement a special kind of queue called a priority queue
- The heap is not very useful as an ADT on its own
  - Will not create a Heap interface or code a class that implements it
  - Will incorporate its algorithms when we implement a priority queue class and Heapsort
- Sometimes a FIFO queue may not be the best way to implement a waiting line
- A priority queue is a data structure in which only the highest-priority item is accessible

Huffman Trees

- A Huffman tree can be implemented using a binary tree and a PriorityQueue
- A straight binary encoding of an alphabet assigns a unique binary number to each symbol in the alphabet
  - Unicode for example
- The message “go eagles” requires 144 bits in Unicode but only 38 using Huffman coding

Huffman Trees (continued)

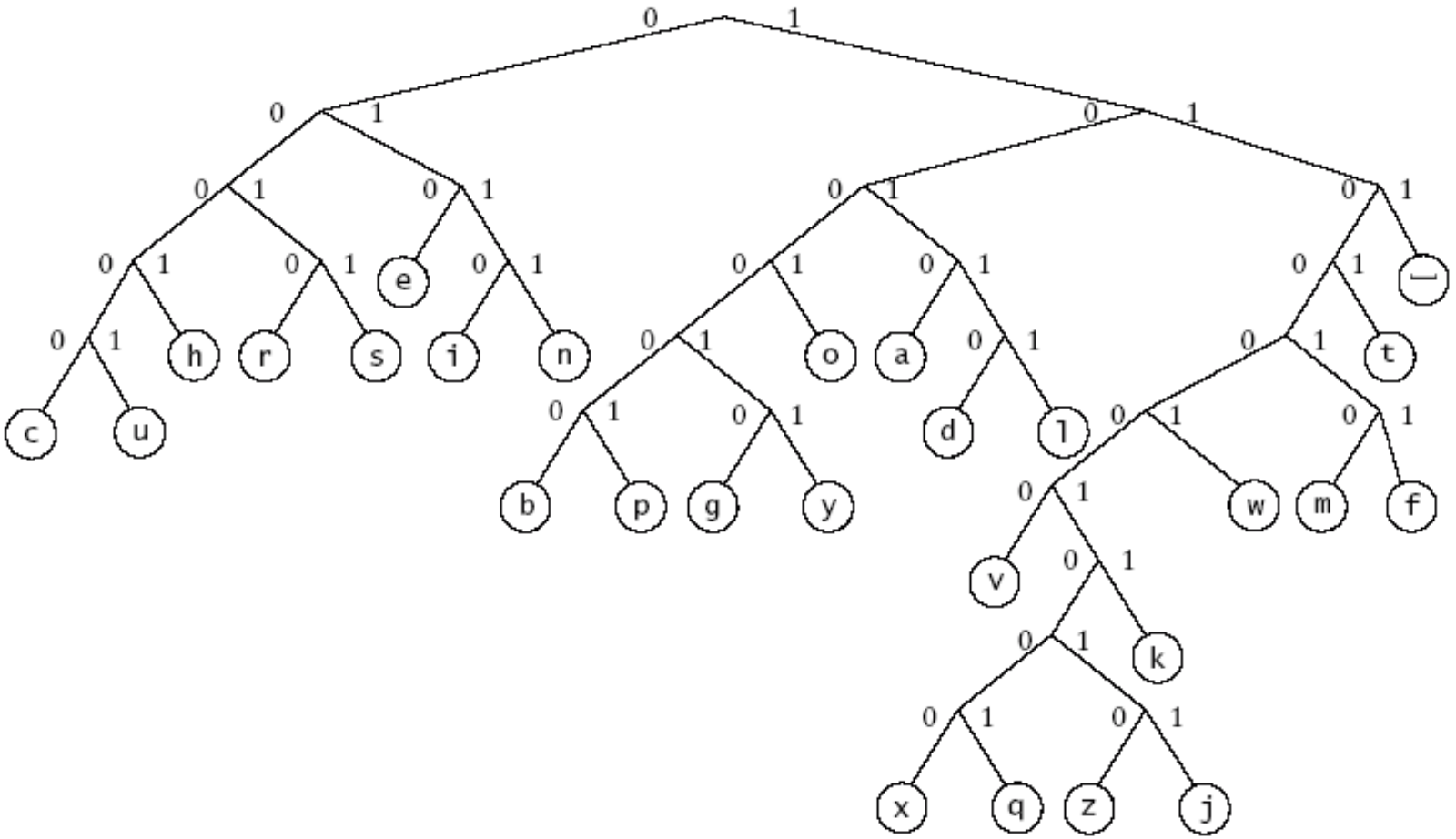
TABLE 8.8
Frequency of Letters in English Text

Symbol	Frequency	Symbol	Frequency	Symbol	Frequency
_	186	h	47	g	15
e	103	d	32	p	15
t	80	l	32	b	13
a	64	u	23	v	8
o	63	c	22	k	5
i	57	f	21	j	1
n	57	m	20	q	1
s	51	w	18	x	1
r	48	y	16	z	1

Huffman Trees (continued)

FIGURE 8.35

Huffman Tree Based on Frequency of Letters in English Text



Chapter Review

- A tree is a recursive, nonlinear data structure that is used to represent data that is organized as a hierarchy
- A binary tree is a collection of nodes with three components: a reference to a data object, a reference to a left subtree, and a reference to a right subtree
- In a binary tree used for arithmetic expressions, the root node should store the operator that is evaluated last
- A binary search tree is a tree in which the data stored in the left subtree of every node is less than the data stored in the root node, and the data stored in the right subtree is greater than the data stored in the root node

Chapter Review (continued)

- A heap is a complete binary tree in which the data in each node is less than the data in both its subtrees
- Insertion and removal in a heap are both O(log n)
- A Huffman tree is a binary tree used to store a code that facilitates file compression