SELF-ASSESSMENT PROCEDURE XIII

A self-assessment procedure dealing with binary search trees and B-trees

by Gopal K. Gupta

What is Self-Assessment Procedure XIII?

This is the thirteenth self-assessment procedure. All the previous ones are listed on the next page. The first seven are collected in a single volume available from ACM.*

This procedure provides an elementary review of fundamental topics concerning binary search trees and B-trees. A reader who is acquainted with the subject as presented in any standard undergraduate data structures text should be able to answer most of the questions quickly. Data structure terminology is not standard. The procedure uses the following definitions:

The *root of a binary tree* is defined to be at level 0. The *level* of a direct descendent of a node of level i is $i + 1$. The maximum level of any element of a binary tree is said to be its *depth* or *height*. If an element has no descendents it is called a *leaf*. It is often convenient to extend a binary tree by adding special nodes (called *external nodes*) wherever a null subtree was present in the original tree. The new tree thus obtained is called an *extended binary tree*.

A *full binary tree* is defined as a binary tree in which each node either is a leaf or has exactly two nonempty descendents. A *complete binary tree* is defined as a binary tree with leaves on at most two adjacent levels $l - 1$ and l in which the leaves at the bottommost level l lie in the leftmost positions of l. Full binary tree is not defined by Knuth (1973b). Our definition of full binary tree is from Standish (1980) and Tremblay and Sorensen (1984). It differs from that of Horowitz and Sahni (1976). The definition of complete binary tree varies; our definition agrees with that used by Knuth (1973a, b) and Standish (1980).

Heap is a binary tree as defined by Standish (1980, p. 84) and Knuth (1973b, p. 145).

The terms *AVL tree* and *height-balanced* tree are used synonymously. The terms *weight-balanced* trees and *bounded-balanced* trees are also used synonymously to refer to trees defined by Nievergelt and Reingold (1973). Several books including Knuth (1973b) do not differentiate between weight-balanced and bounded-balanced trees although we recognize that the weight-balanced trees defined by Baer (1975) are different.

We will use the definition of *B-tree* as presented by Knuth (1973b). Comer (1979) and Wirth (1976, p. 246) use a somewhat different definition.

Most questions have only one correct response. A few questions do have two correct responses.

The next few paragraphs repeat the introduction and instructions given with earlier procedures. Those who read them before may branch directly to the questions.

What is Self-Assessment?

Self-assessment is based on the idea that a question and answer procedure can be devised that will help a person appraise and develop his or her knowledge about a particular topic. It is intended to be an educational experience for a participant. The questions are only the *beginning* of the procedure. They are developed to help the participant think about the concepts and decide whether to pursue the matter further.

Author's Present Address: Gopal K. Gupta, Dept. of Computer Science, Monash University, Clayton, Victoria 3168, Australia; until end of year University of Illinois, 1304 West Springfield Ave., Urbana, IL 61801.

* Available from the ACM Order Department, P.O. Box 64145, Baltimore, MD 21264; for $7 to members and $15 to nonmembers.

The primary motivation of self-assessment is *not* for an individual to satisfy *others* about his or her knowledge; rather it is for a paticipant to appraise and develop his or her own knowledge. This means that there are several ways to use a self-assessment procedure. Some people will start with the questions. Others will read the answers and refer to the references first. These approaches and others devised by the participants are all acceptable if at the end of the procedure the participant can say, "Yes, this has been a worthwhile experience" or "I have learned something."

How to Use the Self-Assessment Procedure

We suggest the following way of using the procedure, but, as noted earlier, there are others. This is not a timed exercise; therefore plan to work with the procedure when you have an hour to spare, or you will be short-changing yourself on this educational experience. Go through the questions and mark the responses you think are most appropriate. Compare your responses with those suggested by the Committee. In those cases where you differ with the Committee, look up the references if the subject seems pertinent to you. In those cases in which you agree with the Committee, but you feel uncomfortable with the subject matter, *and* the subject is significant to you, look up the references.

Some ACM chapters may want to devote a session to discussing this self-assessment procedure or the concepts involved.

The Committee hopes some participants will send comments.

Acknowledgments The author is grateful to B. Srinivasan and two anonymous referees who very carefully read an earlier version of this test and made valuable comments.

PREVIOUS SELF-ASSESSMENT PROCEDURES

Self-Assessment Procedure I
Three concept categories within the programming skills and techniques area
May 1976

Self-Assessment Procedure II
System organization and control with information representation, handling, and manipulation
May 1977

Self-Assessment Procedure III
Internal sorting
September 1977

Self-Assessment Procedure IV
Program development tools and methods, data integrity, and file organization and processing
February 1978

Self-Assessment Procedure V
Database systems
Peter Scheuermann and C. Robert Carlson
August 1978

Self-Assessment Procedure VI
Queueing network models of computer systems
J.W. Wong and G. Scott Graham
August 1979

Self-Assessment Procedure VII
Software science
M.H. Halstead and Victor Schneider
August 1980

Self-Assessment Procedure VIII
The programming language Ada
Peter Wegner
October 1981

Self-Assessment Procedure IX
Ethics in computing
Edited by Eric A. Weiss, from a book by Donn B. Parker
March 1982

Self-Assessment Procedure X
Software project management
Roger S. Gourd
December 1982

Self-Assessment Procedure XI
One part of early computing history
Eric A. Weiss
July 1983

Self-Assessment Procedure XII
Computer architecture
Robert I. Winner and Edward M. Carter
January 1984

Self-Assessment Procedure XIII

This self-assessment procedure is not sanctioned as a test nor endorsed in any way by the Association for Computing Machinery. Any person using any of the questions in this procedure for the testing or certification of anyone other than himself or herself is violating the spirit of this self-assessment procedure and the copyright on this material.

Part I. Questions

1. A binary search tree is defined as
   a. a finite set of nodes which either is empty or consists of a root node with two disjoint binary trees.
   b. a binary tree used for searching.
   c. a binary tree such that for each node all keys in the left subtree of the node are less than the key in the node and those in the right subtree are greater than the key in the node.
   d. a binary tree whose nodes contain keys arranged in descending order along every path from the root to a leaf.

2. The internal path length of an extended binary tree is defined as
   a. the longest path length from the root of the tree to a leaf.
   b. the sum of the lengths of the path from the root to each of the internal nodes.
   c. the sum of the levels of all the internal nodes.
   d. the sum of the levels of all the leaves.

3. The external path length of an extended binary tree is defined as
   a. the sum of the lengths of the paths from the root to each of the leaves.
   b. the sum of the lengths of the paths from the root to each of the external nodes.
   c. the longest path length from the root to an external node.
   d. internal path length plus 1.

4. In an extended binary tree of n internal nodes, the number of external nodes is given by
   a. $n + 1$
   b. n
   c. $2n$
   d. $2n - 1$

5. Let I be the internal path length and E the external path length of an extended binary tree with n internal nodes. Which of the following is correct?
   a. $E = I + n$
   b. $E = I + 2n$
   c. $E = I + n + 3$
   d. $E = I + n + 1$
   e. None of the above.

6. The maximum number of nodes in a binary tree of height h is
   a. 2^{h+1}
   b. $2^{h+1} + 1$
   c. $2^{h+1} - 1$
   d. $2^h + 1$

7. A complete binary tree of height 7 must at least have the following number of nodes.
   a. 64
   b. 65
   c. 129
   d. 33
   e. None of the above.

8. Which one of the following is correct?
   a. In a binary tree of height h, the number of leaves is 2^h.
   b. In a complete binary tree the number of leaves is one more than the number of non-leaf internal nodes.
   c. In a full binary tree, the number of leaves is always one more than the number of non-leaf internal nodes.
   d. A complete binary tree with height h always has $2^{h+1} - 1$ nodes.

9. Assuming that all keys in a tree are searched for with equal likelihood, minimum average search time in a binary tree is achieved if the binary tree is
   a. full.
   b. complete.
   c. height-balanced.
   d. weight-balanced.

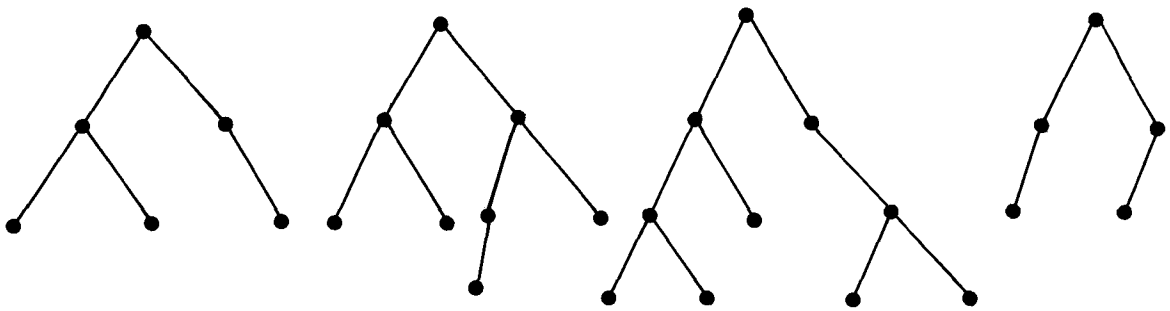


Figure for Question 11

10. An AVL tree is defined as

a. a tree which has two subtrees whose heights differ by at most 1.

b. a binary tree whose leaves appear on at most two adjacent levels.

c. a binary tree whose longest paths in the left subtree and the right subtree differ by at most 1.

d. a tree such that it and each of its subtrees have the property that the heights of the left and right subtrees of the root differ by at most 1.

11. For the trees at the top of the page, which of the following is correct?

a. None of the above trees is AVL.

b. None of the above trees is a full tree.

c. All of the above trees are bounded-balance BB[2/5].

d. All of the above trees are complete.

12. Which of the following is correct?

a. A heap is always a binary search tree.

b. A binary search tree is always a heap.

c. A heap is always a complete binary tree.

d. A complete binary tree is always a heap.

13. An AVL tree of height 4 must have at least the following number of nodes.

a. 21

b. 7

c. 13

d. 12

e. None of the above.

14. For the trees at the foot of the page, which of the following is *not* correct?

a. The tree D below is obtained from tree A by applying a double rotation at the node 10.

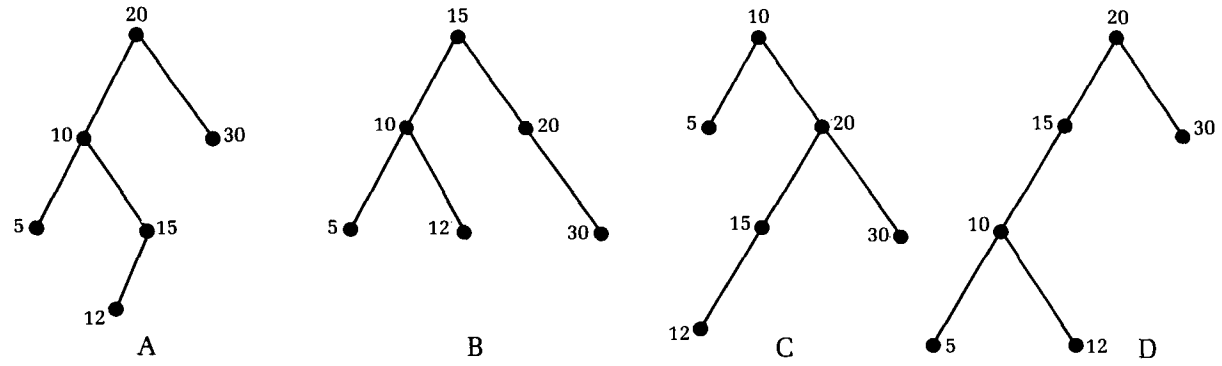
b. The tree C below is obtained from tree A by applying a single rotation at the root.

c. The tree B below is obtained from tree A by applying a double rotation at the root.

d. The tree B below is obtained from tree A by applying two single rotations.

e. The tree B below is obtained from tree D by applying a single rotation at the root.

Figure for Question 14



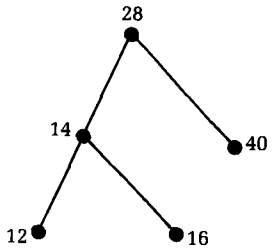


Figure for Question 15

15. For the tree above, which of the following is *not* correct?

a. If a key with value 42 or 29 or 27 is inserted in the tree, it does not lose the AVL property.

b. After the insertion of 20, a single rotation is necessary to restore the AVL property of the tree.

c. A double rotation is necessary to restore the AVL property after the insertion of 20. This makes 16 the new root of the tree.

d. A single rotation is needed to restore the AVL property if any node with value less than 14 is added to the present tree.

16. For the tree below, which of the following is correct?

a. If the node 3 is deleted, one double rotation is needed to keep the tree height-balanced.

b. If the node 3 is deleted, two single rotations will restore the AVL property of the tree.

c. If the node 10 is deleted, 20 becomes the new root.

d. If the node 17 is deleted, a double rotation must be applied to restore the AVL property of the tree.

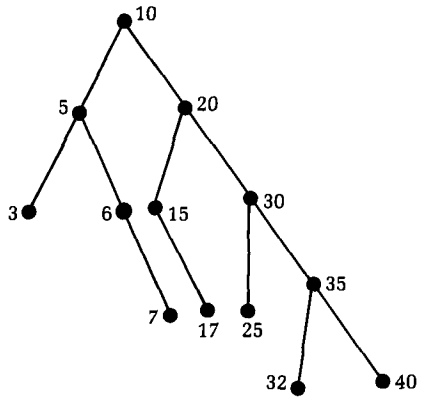


Figure for Questions 16 and 17

17. For the AVL tree at the foot of the page, which of the following is *not* correct?

a. If the node 20 is deleted, 25 is moved to replace it making the tree lose the AVL property at the node 30.

b. If the node 25 is deleted, a single rotation is needed to restore the AVL property.

c. If the node 10 is deleted, it is replaced by 15 and a single rotation is needed at the node 20 to restore the AVL property of the tree.

d. On the average the number of rotations necessary to restore the AVL property are higher for each deletion than for each insertion.

18. Which of the following is *not* correct?

a. AVL trees are often used for storing dynamic files on secondary storage.

b. AVL trees are mainly used for maintaining dynamic files in primary storage.

c. Given a distribution of frequencies of search for the keys, AVL trees may also be used for building almost optimal trees.

d. A B-tree is a good scheme for storing and maintaining large dynamic files on secondary storage.

19. Which of the following is correct?

a. If insertions and deletions are made randomly, the AVL tree will on the average be about 38.6 percent less efficient in searching than the complete binary tree with the same number of nodes.

b. If insertions and deletions are made randomly, the B-tree on the average utilizes 50 percent of the storage.

c. Searching an AVL tree is guaranteed to take at most about 44 percent more comparisons than searching a complete binary tree with the same number of nodes.

d. If insertions and deletions are made in an AVL search tree at random, then no search path length will be longer than about 1.386 times the path length for a complete binary tree with the same number of nodes.

20. Which of the following is *not* correct?

a. While an insertion in an AVL tree requires at most one rotation, deletion of a node may require up to log(n) rotations.

b. A random insertion in an AVL tree requires rebalancing 83 percent of the time.

c. The average cost of searching an AVL tree is almost the same as that of a complete

binary tree with the same number of nodes.

d. For sufficiently large AVL trees, the average number of rotations needed per insertion to restore the AVL property is independent of the size of the tree.

21. Which of the following is *not* correct?

a. Fibonacci trees are bounded balance (BB) trees with $\alpha = \frac{1}{3}$.

b. All AVL trees are bounded balance trees with $\alpha = \frac{1}{3}$.

c. For any non-zero α, AVL trees can be constructed which are not BB[α].

d. BB[α] trees (α not equal to $\frac{1}{2}$) in general do not satisfy the criterion of AVL trees.

22. Which of the following is correct?

a. An AVL tree is always a complete binary tree.

b. A bounded-balance tree is always an AVL tree.

c. A complete binary tree is always an AVL tree.

d. A full binary tree is always an AVL tree.

23. In a binary search tree of n nodes, the average cost (i.e. the number of comparisons) of successful search (C_n) is

a. given by

$$C_n = \frac{(C'_n + 1)n}{(n + 1)}$$

where C'_n is the cost of unsuccessful search.

b. given by

$$C_n = (1 + 1/n)C'_n - 1$$

c. on the average the same as the cost of deleting a node.

d. on the average the same as the cost of inserting a node in the tree.

24. Given n keys K_i, $i = 1, 2, \ldots, n$ and associated frequency counts of successful search a_i, $i = 1, 2, \ldots, n$ and unsuccessful search b_i, $i = 0, 1, \ldots, n$, we can build an optimal tree bottom-up based on the following main principle.

a. Optimal trees have all their subtrees such that the root is the node of the highest weight in the subtree.

b. Optimal trees have all their subtrees such that the root balances the weights in the subtree.

c. All subtrees of an optimal tree are optimal.

d. The weighted path length of the optimal tree is the sum of the weighted path lengths of right and left subtrees plus the weight.

25. A weighted binary search tree T has as its root the key K_r with a weight of 10 and subtrees T_L and T_R. Internal weighted path lengths of T_L and T_R are 79 and 95 respectively and their weights are 50 and 60 respectively. The internal weighted path length of the tree is

a. 174.

b. 284.

c. 294.

d. None of the above.

26. The optimum tree of 7 keys (say K_i, $i = 1, 2, \ldots, 7$) is being built. Let p_{ij} be the weighted path length of the optimum binary tree with keys K_{i+1} to K_j. In the middle of the algorithm, values of the weighted path lengths p_{03}, p_{14}, p_{25}, and p_{36} computed are 36, 58, 48, and 46 respectively. The roots of the subtrees corresponding to the above path lengths were K_3, K_3, K_4, and K_4 respectively. Given this information, we can say that the root of the optimum tree must be

a. K_3

b. K_4

c. K_3 or K_4. There is not enough information to decide which one.

d. K_3 and K_4 both since two optimum trees with same weighted path lengths exist.

27. Which of the following is *not* correct?

a. The optimum node size for disk resident B-trees should be selected based on rotational delay time, seek time, transfer rate, and key size.

b. A file organization based on B-trees is guaranteed to have at least about 50 percent storage utilization.

c. A file organization based on B-trees on the average has storage utilization of about 69 percent.

d. A B*-tree guarantees storage utilization of at least about 75.0 percent.

28. Some B-tree structures store all the records in the leaves only. We will call such trees B^+-trees. Which of the following statements is *not* correct?

a. In a B^+-tree the records and the index are separate.

b. Sequential processing of the records is facilitated in the B^+-trees.

c. If the order of the tree is large, say 100, storing all the records in the leaves does involve substantially more storage than used by the conventional B-trees.

d. Since the index needs to store only the keys and not the records, the order of the tree can be higher and therefore searching can be faster in B^+-trees.

29. The tree at the foot of the page only shows the keys present in each node. It does not show the size of the nodes. All the nodes are of equal size. Which of the following statements is correct?

a. The tree is not a B-tree since the storage utilization is less than 50 percent.

b. The B-tree must be of order 3.

c. The tree must be of order 3 or 4.

d. The tree could be of order 2, 3, or 4.

e. The tree could be of order 3, 4, or 5.

30. For the B-tree at the foot of the page, which of the following statements is *not* correct?

a. The B-tree has the minimum number of nodes possible in a B-tree of order 5 and height 2.

b. The B-tree has the maximum number of nodes possible in a B-tree of order 3 and height 2.

c. If the order of the B-tree is 5, the height of the tree may not increase even if 100 keys are inserted in the tree.

d. If the order of the B-tree is 5, insertion of another 40 keys may increase the height of the tree.

31. Which of the following statements is *not* correct?

a. The height of a B-tree grows only when the root is split.

b. In the worst case, only 4 accesses are necessary if a file of about 2 million records is stored as a B-tree of order 200.

c. In the best case about 16 million records could be stored in a B-tree of order 200 and height 2.

d. In the worst case, a B-tree of height 2 may contain only half the number of records that may be stored in a B-tree of the same height in which storage utilization is 100%.

32. A B-tree of order 5 is built-up by inserting the keys 10, 20, 30, 25, 15, 5, 17, 27, 37, 35, 32 and 22 in that order to a null tree. Which of the following statements is correct about the resulting tree?

a. The tree is of height 2.

b. The root node has keys 10, 20, 25 and 30.

c. The root node has keys 20 and 30.

d. Altogether there are 5 nodes in the tree.

33. A B-tree is built-up by inserting the keys in the last question in the reverse order (that is, by first inserting 22 to a null tree, then inserting 32, 35, 37, and so on). If the resulting tree is compared with the tree obtained in the last question, which of the following statements is correct?

a. The two B-trees are identical.

b. The number of nodes in the two trees is the same.

c. The storage utilization in the two trees is the same.

d. The height of the two trees is the same.

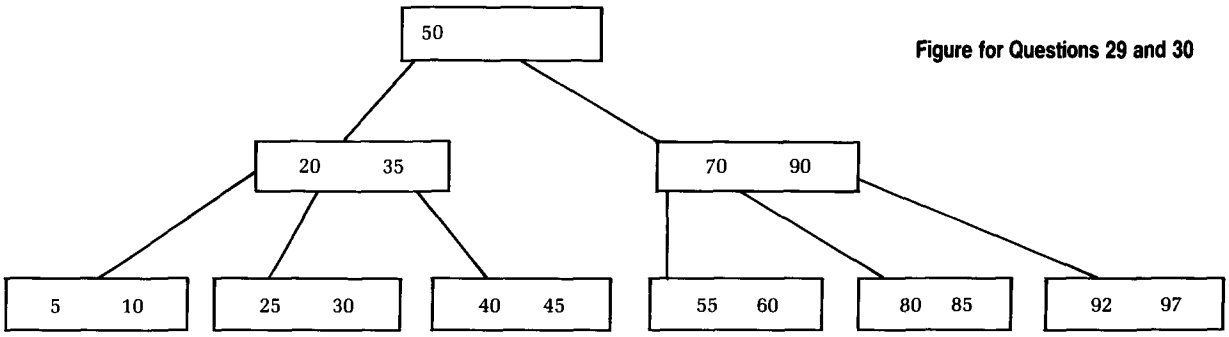


Figure for Questions 29 and 30

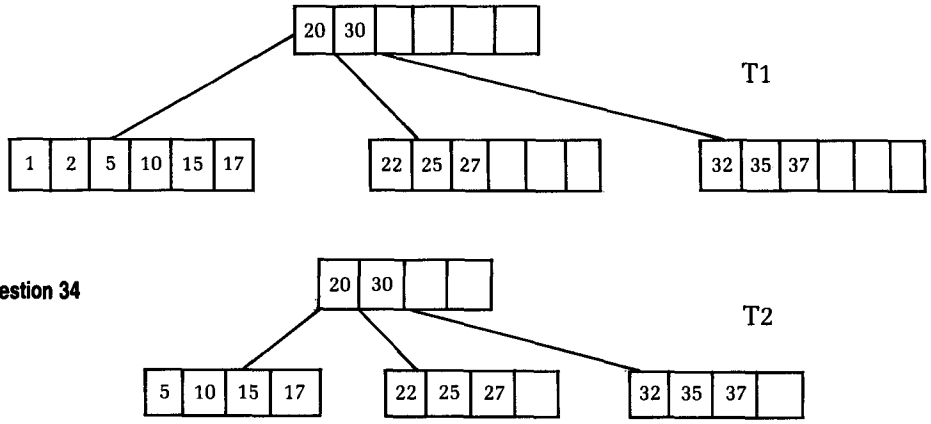


Figure for Question 34

34. In the figure above, the tree T1 is a B-tree of order 7 and tree T2 is of order 5. Which of the following is *not* correct?

a. If key 30 is deleted from the tree T2, either 22 or 32 is moved to the root node.

b. If key 32 is deleted from the tree T1, two rightmost nodes at level 1 and key 30 from the root node are combined.

c. If key value 2 is inserted in the tree T2, the node is split and the key 10 is moved to the root.

d. If key value 22 is deleted from the tree T1, 20 is moved to the leaf node which underflows and 17 is moved to the root.

35. Given the B-tree of order 5 at the foot of the page, which of the following sequence of operations is correct when the keys 25, 26 and 24 are deleted?

a. The key 25 is deleted from the tree, the root disappears and the nodes B and C are combined. Deletion of 26 results in borrowing of key 29 from the node C and moving of 32 to the node C. Deletion of 24 similarly involves moving of 19 down to the node F and of 18 up to the node B.

b. The key 25 is deleted and a key from the leaves is moved to the root. If 26 is moved to the root, the key 29 moves down to the node G while 32 moves up to the node C. Deletion of 26 then involves moving 27 to the root and concatenation of the nodes G and H and key 32. Deletion of 24 then causes 19 to be moved to the node F and 18 to be moved to the node B.

c. When the key 25 is deleted, 26 is moved to the root. The node G underflows and this results in key 29 moving down to the node G while 32 moves up to the node C. Deletion of 26 then involves moving 27 to the root and concatenation of the nodes G and H and key 32. This makes the node C to underflow and this results in concatenation of the node C, node B and the root to form the new root. The height of the tree is therefore reduced. Deletion of 24 now involves moving 19 to the node F and 18 to the node B.

36. Which of the following is correct?

a. A 2-3 tree is a B-tree of order 3.

b. A 2-3 tree is a B-tree in which the number of keys in each node may be 2 or 3.

c. A 2-3 tree is a multiway tree in which each node except the leaves has 2 keys and 3 descendents.

d. Storage utilization in the 2-3 tree on the average is about 79 percent.

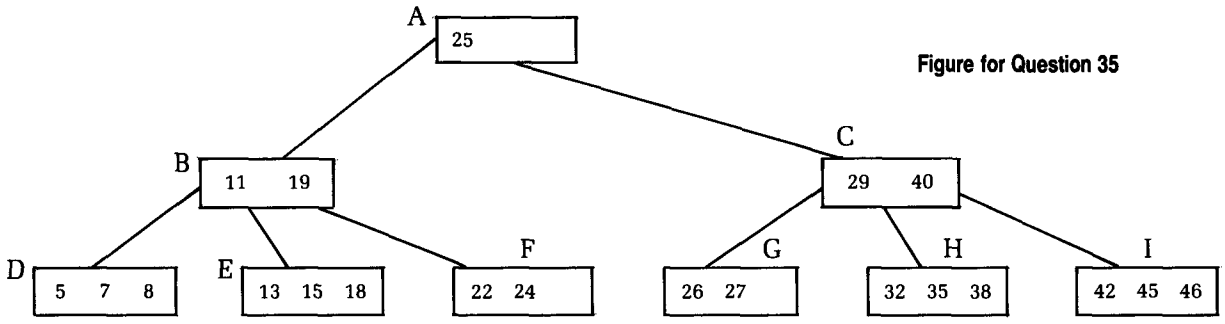


Figure for Question 35

Part II. Suggested Responses

1. *c.* **2.** *b* or *c.* **3.** *b.* **4.** *a.* **5.** *b.* **6.** *c.* **7.** *e.* **8.** *c.* **9.** *b.* **10.** *d.* **11.** *b.* **12.** *c.* **13.** *d.* **14.** *a.* **15.** *a, b.* **16.** *b.* **17.** *d.* **18.** *a, c.* **19.** *c.* **20.** *b.* **21.** *b.* **22.** *c.* **23.** *b.* **24.** *c.* **25.** *b.* **26.** *c.* **27.** *d.* **28.** *c.* **29.** *e.* **30.** *b.* **31.** *d.* **32.** *c.* **33.** *d.* **34.** *d.* **35.** *c.* **36.** *a.*

Part III. Suggested References

(References are fully cited in Reference Titles below.)

1. Wirth (1976), p. 200.
2. Knuth (1973a), p. 399 and Standish (1980), p. 101.
3. Same as for question 2.
4. Knuth (1973a), p. 399.
5. Knuth (1973a), p. 400 and Standish (1980), p. 102.
6. Standish (1980), p. 109.
7. Knuth (1973a), pp. 400–401.
8. Standish (1980), pp. 53–54.
9. Standish (1980), p. 103 and Knuth (1973a), p. 400.
10. Knuth (1973b), p. 452.
11. Standish (1980), p. 51 and Gotlieb and Gotlieb (1978), pp. 221–222.
12. Knuth (1973b), p. 145 and Standish (1980), p. 84.
13. Standish (1980), pp. 109–110.
14. Knuth (1973b), p. 454.
15. Knuth (1973b), p. 454.
16. Knuth (1973b), pp. 454, 465–466 and Gotlieb and Gotlieb (1978), pp. 219–221.
17. Same as for question 16.
18. Standish (1980), pp. 108, 118 and Knuth (1973b), p. 452.
19. Standish (1980), p. 110, Knuth (1973b), p. 453 and Wirth (1976), p. 214–215.
20. Standish (1980), pp. 110–111 and 117 and Knuth (1973b), pp. 465–466.
21. Gotlieb and Gotlieb (1978), pp. 221–222 and Nievergelt (1974), p. 205.
22. Standish (1980), pp. 84, 107–108 and Gotlieb and Gotlieb (1978), p. 221.
23. Knuth (1973b), p. 410 and Standish (1980), p. 104.
24. Knuth (1973b), p. 435.
25. Knuth (1973b), p. 435.
26. Knuth (1973b), p. 436.
27. Knuth (1973b), pp. 478–479, Comer (1979), pp. 129, 133 and Gotlieb and Gotlieb (1978), pp. 340–341.
28. Comer (1979), pp. 129–130.
29. Knuth (1973b), pp. 473–474.
30. Knuth (1973b), pp. 473–476.
31. Knuth (1973b), p. 476.
32. Knuth (1973b), pp. 475–476.
33. Same as for question 32.
34. Same as for question 32.
35. Same as for question 32.
36. Gotlieb and Gotlieb (1978), p. 225, including footnote 20.

Reference Titles

1. Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1983), *Data Structures and Algorithms*, Addison-Wesley, Reading, Mass.
2. Baer, J.-L. (1975), "Weight-Balanced Trees," *Proceedings of AFIPS 1975 NCC*, Vol. 44, AFIPS Press, Reston, Va., pp. 467–472.
3. Comer, D. (1979), "The Ubiquitous B-Tree," *ACM Computing Surveys*, Vol. 11, pp. 121–137.
4. Gotlieb, C.C. and Gotlieb, L.R. (1978), *Data Types and Structures*, Prentice-Hall, Englewood Cliffs, N.J.
5. Horowitz, E. and Sahni, S. (1976), *Fundamentals of Data Structures*, Computer Science Press, Rockville, Md.
6. Knuth, D.E. (1973a), *The Art of Computer Programming Volume 1: Fundamental Algorithms*, Addison-Wesley, Reading, Mass.
7. Knuth, D.E. (1973b), *The Art of Computer Programming Volume 3: Searching and Sorting*, Addison-Wesley, Reading, Mass.
8. Nievergelt, J. and Reingold, E.M. (1973), "Binary Search Trees of Bounded Balance," *SIAM Journal of Computing*, Vol. 2, pp. 33–43.
9. Nievergelt, J. (1974), "Binary Search Trees and File Organization," *ACM Computing Surveys*, Vol. 6, pp. 195–207.
10. Standish, T. (1980), *Data Structure Techniques*, Addison-Wesley, Reading, Mass.
11. Tremblay, J.P. and Sorensen, P.G., Second edition (1984), *An Introduction to Data Structures with Applications*, McGraw-Hill, New York.
12. Wirth, N. (1976), *Algorithms + Data Structures = Programs*, Prentice-Hall, Englewood Cliffs, N.J.

Epilogue

Now that you have reviewed this self-assessment procedure and have compared your responses to those suggested, you should ask yourself whether this has been a successful educational experience. The Committee suggests that you conclude that it has only if you have:
—discovered some concepts that you did not previously know about or understand, or
—increased your understanding of those concepts which were relevant to your work or valuable to you.