Assignment No (8)

Name:		Date	
Topic:		Lecture No:	

Answer all of the following questions:

Solved problems

BINARY TREES

7.1 Suppose T is the binary tree stored in memory as in Fig. 7-43. Draw the diagram of T.

ROOT: 5

AVAIL: 9

	INFO	LEFT	RIGHT
1	20	0	0
2	30	1	13
3	40	0	0
4	50	0	0
5	60	2	6
6	70	0	8
7	80	0	0
8	90	7	14
9		10	
10		0	
11	35	0	12
12	45	3	4
13	55	11	0
14	95	0	0

Fig. 7-43

7.2 A binary tree T has 9 nodes. The inorder and preorder traversals of T yield the following sequences of nodes:

Inorder: E A C K F H D B G

Preorder: F A E K C D H G B

Draw the tree T.

7.3 Consider the algebraic expression $E = (2x + y)(5a - b)^3$.

(a) Draw the tree T which corresponds to the expression E.

(b) Find the *scope* of the exponential operator; i.e., find the subtree rooted at the exponential operator.

(c) Find the prefix Polish expression P which is equivalent to E, and find the preorder of T.

7.4 Suppose a binary tree T is in memory. Write a recursive procedure which finds the number NUM of nodes in T.

7.5 Suppose a binary tree T is in memory. Write a recursive procedure which finds the depth DEP of T.

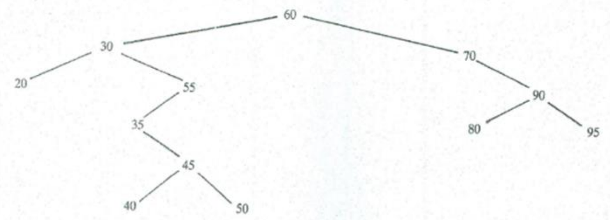
7.6 Draw all the possible nonsimilar trees T where:

(a) T is a binary tree with 3 nodes.

Solved problems

BINARY SEARCH TREES

7.7 Consider the binary search tree T in Fig. 7-44(*b*), which is stored in memory as in Fig. 7-43. Suppose ITEM = 33 is added to the tree T. (*a*) Find the new tree T. (*b*) Which changes occur in Fig. 7-43?



(*b*)

Fig. 7-44

ROOT: 5

AVAIL: 9

	INFO	LEFT	RIGHT
1	20	0	0
2	30	1	13
3	40	0	0
4	50	0	0
5	60	2	6
6	70	0	8
7	80	0	0
8	90	7	14
9		10	
10		0	
11	35	0	12
12	45	3	4
13	55	11	0
14	95	0	0

Fig. 7-43

7.8 Suppose the following list of letters is inserted in order into an empty binary search tree:

$$J, R, D, G, T, E, M, H, P, A, F, Q$$

(*a*) Find the final tree T and (*b*) find the inorder traversal of T.

7.10 Suppose n data items $A_1, A_2, \ldots, A_N$ are already sorted, i.e.,

$$A_1 < A_2 < \cdots < A_N$$

(*a*) Assuming the items are inserted in order into an empty binary search tree, describe the final tree T.

(*b*) What is the depth D of the tree T?

(*c*) Compare D with the average depth AD of a binary search tree with n nodes for (i) $n = 50$, (ii) $n = 100$ and (iii) $n = 500$.

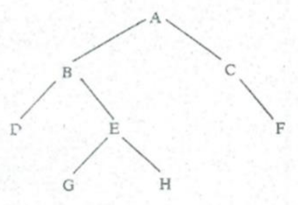
Supplementary Problems

BINARY TREES

7.19 Consider the tree T in Fig. 7-60(*a*).

(*a*) Fill in the values for ROOT, LEFT and RIGHT in Fig. 7-60(*b*) so that T will be stored in memory.

(*b*) Find (i) the depth D of T, (ii) the number of null subtrees and (iii) the descendants of node B.



(*a*)

ROOT: [ ]

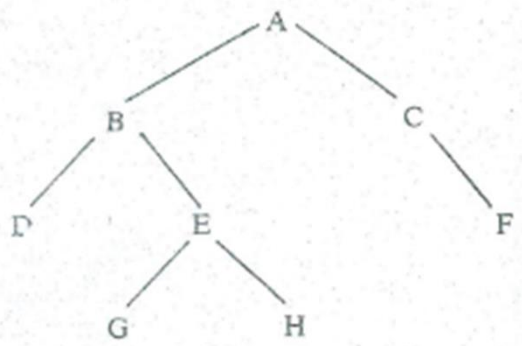
AVAIL: 5

	INFO	LEFT	RIGHT
1	A		
2	C		
3	D		
4	G		
5		6	
6		0	
7	H		
8	F		
9	E		
10	B		

(*b*)

Fig. 7-60

7.20 List the nodes of the tree T in Fig. 7-60(*a*) in (*a*) preorder, (*b*) inorder and (*c*) postorder.



7.21 Draw the diagram of the tree T in Fig. 7-61.

ROOT: 14

AVAIL: 8

	INFO	LEFT	RIGHT
1	H	4	11
2	R	0	0
3		17	
4	P	0	0
5	B	18	7
6		3	
7	E	1	0
8		6	
9	C	0	10
10	F	15	16
11	Q	0	12
12	S	0	0
13		0	
14	A	5	9
15	K	2	0
16	L	0	0
17		13	
18	D	0	0

Fig. 7-61

7.22 Suppose the following sequences list the nodes of a binary tree T in preorder and inorder, respectively:

Preorder: G, B, Q, A, C, K, F, P, D, E, R, H

Inorder: Q, B, K, C, F, A, G, P, E, D, H, R

Draw the diagram of the tree.

7.23 Suppose a binary tree T is in memory and an ITEM of information is given.

(*a*) Write a procedure which finds the location LOC of ITEM in T (assuming the elements of T are distinct).

(*b*) Write a procedure which finds the location LOC of ITEM and the location PAR of the parent of ITEM in T.

(*c*) Write a procedure which finds the number NUM of times ITEM appears in T (assuming the elements of T are not necessarily distinct).

Remark: T is not necessarily a binary search tree.

7.24 Suppose a binary tree T is in memory. Write a nonrecursive procedure for each of the following:

(*a*) Finding the number of nodes in T.

(*b*) Finding the depth D of T.

(*c*) Finding the number of terminal nodes in T.

7.25 Suppose a binary tree T is in memory. Write a procedure which deletes all the terminal nodes in T.

7.26 Suppose ROOTA points to a binary tree T_1 in memory. Write a procedure which makes a copy T_2 of the tree T_1 using ROOTB as a pointer.

Supplementary Problems

BINARY SEARCH TREES

7.27 Suppose the following eight numbers are inserted in order into an empty binary search tree T:

50, 33, 44, 22, 77, 35, 60, 40

Draw the tree T.

7.28 Consider the binary search tree T in Fig. 7-62. Draw the tree T if each of the following operations is applied to the original tree T. (That is, the operations are applied independently, not successively.)

(*a*) Node 20 is added to T.
(*b*) Node 15 is added to T.
(*c*) Node 88 is added to T.
(*d*) Node 22 is deleted from T.
(*e*) Node 25 is deleted from T.
(*f*) Node 75 is deleted from T.

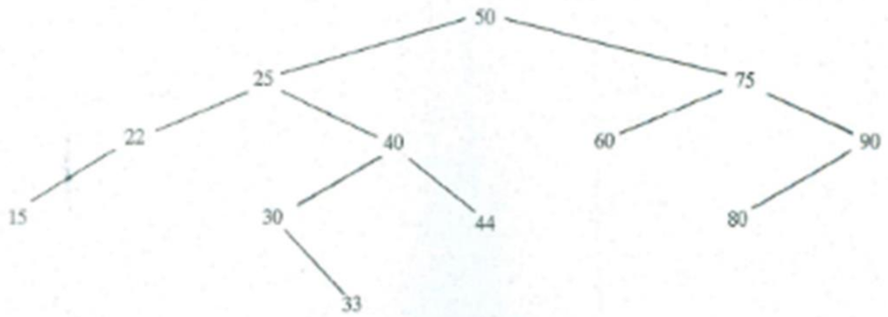


Fig. 7-62

7.29 Consider the binary search tree T in Fig. 7-62. Draw the final tree T if the six operations in Problem 7.28 are applied one after the other (not independently) to T.

(*a*) Node 20 is added to T.
(*b*) Node 15 is added to T.
(*c*) Node 88 is added to T.
(*d*) Node 22 is deleted from T.
(*e*) Node 25 is deleted from T.
(*f*) Node 75 is deleted from T.

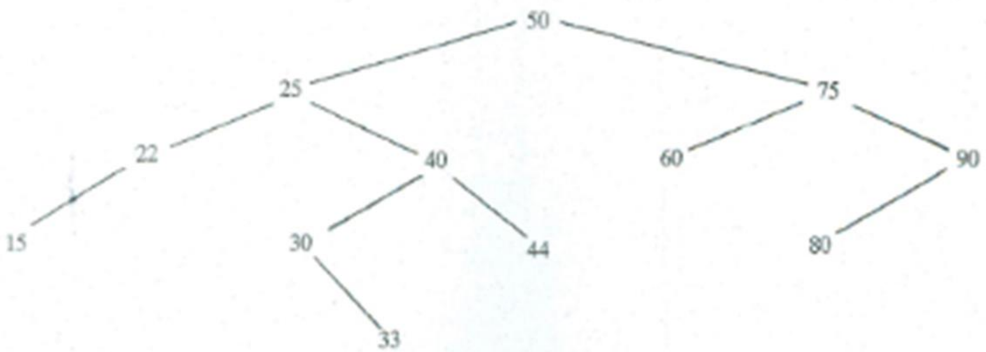


Fig. 7-62

7.30 Draw the binary search tree T in Fig. 7-63.

ROOT = 4

AVAIL = 3

	INFO	LEFT	RIGHT
1	Jones	7	0
2	Fox	11	1
3		8	
4	Murphy	2	15
5		13	
6	Thomas	0	0
7	Green	0	0
8		9	
9		10	
10		5	
11	Conroy	0	0
12	Parker	0	0
13		14	
14		0	
15	Rosen	12	6

Fig. 7-63

7.31 Consider the binary search tree T in Fig. 7-63. Describe the changes in INFO, LEFT, RIGHT, ROOT and AVAIL if each of the following operations is applied independently (not successively) to T.

(*a*) Davis is added to T.
(*b*) Harris is added to T.
(*c*) Smith is added to T.
(*d*) Parker is deleted from T.
(*e*) Fox is deleted from T.
(*f*) Murphy is deleted from T.

ROOT = 4

AVAIL = 3

	INFO	LEFT	RIGHT
1	Jones	7	0
2	Fox	11	1
3		8	
4	Murphy	2	15
5		13	
6	Thomas	0	0
7	Green	0	0
8		9	
9		10	
10		5	
11	Conroy	0	0
12	Parker	0	0
13		14	
14		0	
15	Rosen	12	6

Fig. 7-63

7.32 Consider the binary search tree T in Fig. 7-63. Describe the final changes in INFO, LEFT, RIGHT, ROOT and AVAIL if the six operations in Problem 7.31 are applied one after the other (not independently) to T.

(*a*) Davis is added to T.
(*b*) Harris is added to T.
(*c*) Smith is added to T.
(*d*) Parker is deleted from T.
(*e*) Fox is deleted from T.
(*f*) Murphy is deleted from T.

ROOT: 4

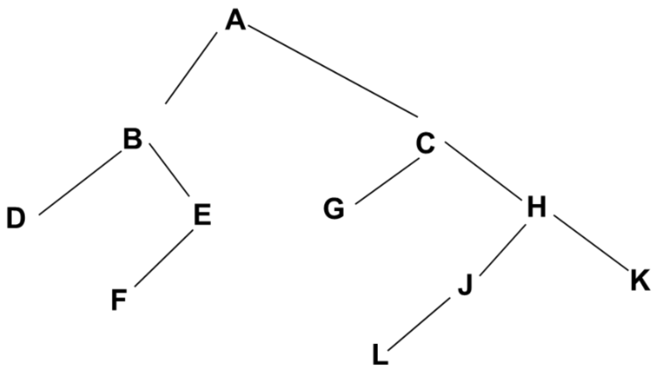
AVAIL: 3

	INFO	LEFT	RIGHT
1	Jones	7	0
2	Fox	11	1
3		8	
4	Murphy	2	15
5		13	
6	Thomas	0	0
7	Green	0	0
8		9	
9		10	
10		5	
11	Conroy	0	0
12	Parker	0	0
13		14	
14		0	
15	Rosen	12	6

Fig. 7-63

Programming problems

Problems 7.43 to 7.45 refer to the tree T in Fig. 7-1, which is stored in memory as in Fig. 7-68.



ROOTA: 5

AVAIL: 8

	INFO	LEFT	RIGHT
1	K	0	0
2	C	3	6
3	G	0	0
4		14	
5	A	10	2
6	H	17	1
7	L	0	0
8		9	
9		4	
10	B	18	13
11		19	
12	F	0	0
13	E	12	0
14		15	
15		16	
16		11	
17	J	7	0
18	D	0	0
19		20	
20		21	
21		22	
22		23	
23		24	
24		0	

Fig. 7-68

7.43 Write a program which prints the nodes of T in (*a*) preorder, (*b*) inorder and (*c*) postorder.

7.44 Write a program which prints the terminal nodes of T in (*a*) preorder (*b*) inorder and (*c*) postorder. (*Note*: All three lists should be the same.)

7.45 Write a program which makes a copy T' of T using ROOTB as a pointer. Test the program by printing the nodes of T' in preorder and inorder and comparing the lists with those obtained in Prob. 7.43.