Name: ____Sample Solution____________________

Email address: ____________________________________

Quiz Section: __________

<u>CSE 332 Winter 2011: Midterm Exam</u>

(closed book, closed notes, no calculators)

Instructions: Read the directions for each question carefully before answering. We will give partial credit based on the work you **write down**, so show your work! Use only the data structures and algorithms we have discussed in class or that were mentioned in the book so far.

Note: For questions where you are drawing pictures, please circle your final answer for any credit.

Good Luck!

Total: 100 points. Time: 50 minutes.

Question	**Max Points**	**Score**
1	22	
2	10	
3	6	
4	8	
5	8	
6	8	
7	14	
8	14	
9	10	
Total	100	

1. (22 pts) Big-Oh

(2 pts each) For each of the functions *f(N)* given below, indicate the tightest bound possible (in other words, giving $O(2^N)$ as the answer to every question is not likely to result in many points). Unless otherwise specified, all logs are base 2. **You MUST choose your answer from the following** (not given in any particular order), each of which could be re-used (could be the answer for more than one of a) – k)):

$O(N^2)$, $O(N^{½})$,$O(N^3 \log N)$, $O(N \log N)$, $O(N \log \log N)$, $O(N)$, $O(N^2 \log N)$, $O(N^5)$, $O(\log N)$,$O(1)$,$O(\log \log N)$,$O(N^4)$,$O(N^{12})$,$O(N^N)$,$O(N^8)$,$O(\log^2 N)$, $O(N^{10})$, $O(2^N)$, $O(N^3)$

You do not need to explain your answer.

a) $f(N) = (1/2)(N \log N) + (\log N)^2$ $O(N \log N)$

b) $f(N) = \log_{10}(2^N)$ $O(N)$

c) $f(N) = N \log \log N + 2N \log N^2$ $O(N \log N)$

d) $f(N) = N^2 \cdot (N \log (N/2) + 2000)$ $O(N^3 \log N)$

e) $f(N) = N! + 2^N$ $O(N^N)$

1. (cont)

f) *Findmax in an AVL Tree containing N elements (worst case)* O(log N)

g) *Delete in a Binary Search Tree containing N elements (worst case)* O(N)

h) *Buildheap in a binary min heap containing N elements (worst case)* O(N)

i) *DecreaseKey(k, v) on a* ***binary min heap*** *containing N elements. Assume you have a reference to the key k. v is the amount that k should be decreased. (worst case)* O(log N)

j) T(N) = T(N/2) + 5 O(log N)

k) T(N) = T(N-1) + 5 O(N)

2. (10 pts) Big-O, Big Ω, Big Θ

(2 pts each) For parts (a) – (c) circle whether the statement is true or false. Part (d) has its own instructions. You do not need to show your work for parts (a) – (c).

TRUE / FALSE a) $5\ N^2 = \Omega(N)$

TRUE / FALSE b) $N \log N + N^2 = \Omega\ (N^2)$

TRUE / FALSE c) $3N^2 + N^3 = \Theta(N^3)$

d) (4 points) Demonstrate that: $45n^2 + 100$ is $O(n^2)$ by finding positive integers c and n_0 such that the definition of Big Oh is satisfied. You do not need to prove that these constants satisfy the definition, just specify the constants.

$$45n^2 + 100 \leq c \cdot n^2$$

for $n \geq n_0$, Many values for $c + n_0$ work.

Example Values: $\begin{cases} n_0 = 100 \\ c = 50 \end{cases}$

3. (6 pts) Recurrence Relationships -

Suppose that the running time of an algorithm satisfies the recurrence relationship

$T(1) = 6.$

and

$T(N) = 2 * T(N/2) + 5N$ for integers $N > 1$

Find the closed form for T(N) ***and show your work step by step***. In other words express T(N) as a function of N. Your answer should *not* be in Big-Oh notation – show the relevant *exact* constants in your answer (e.g. don't use "C" in your answer).

$$T(N) = 2 \cdot T\left(\frac{N}{2}\right) + 5N$$

$$= 2\left(2 \cdot T\left(\frac{N}{4}\right) + 5 \cdot \frac{N}{2}\right) + 5N$$

$$= 2\left(2 \cdot \left(2 \cdot T\left(\frac{N}{8}\right) + 5 \cdot \frac{N}{4}\right) + 5 \cdot \frac{N}{2}\right) + 5N$$

$$= 2^3 \cdot T\left(\frac{N}{2^3}\right) + 2^2 \cdot 5 \frac{N}{2^2} + 2^1 \cdot 5 \frac{N}{2^1} + 5N$$

$$= 2^3 \cdot T\left(\frac{N}{2^3}\right) + 5N + 5N + 5N$$

$$= 2^3 \cdot T\left(\frac{N}{2^3}\right) + 3 \cdot 5N$$

We want:

$$\frac{N}{2^K} = 1$$

$$2^K = N$$

$$K = \log_2 N$$

$$= 2^K \cdot T\left(\frac{N}{2^K}\right) + K \cdot 5N$$

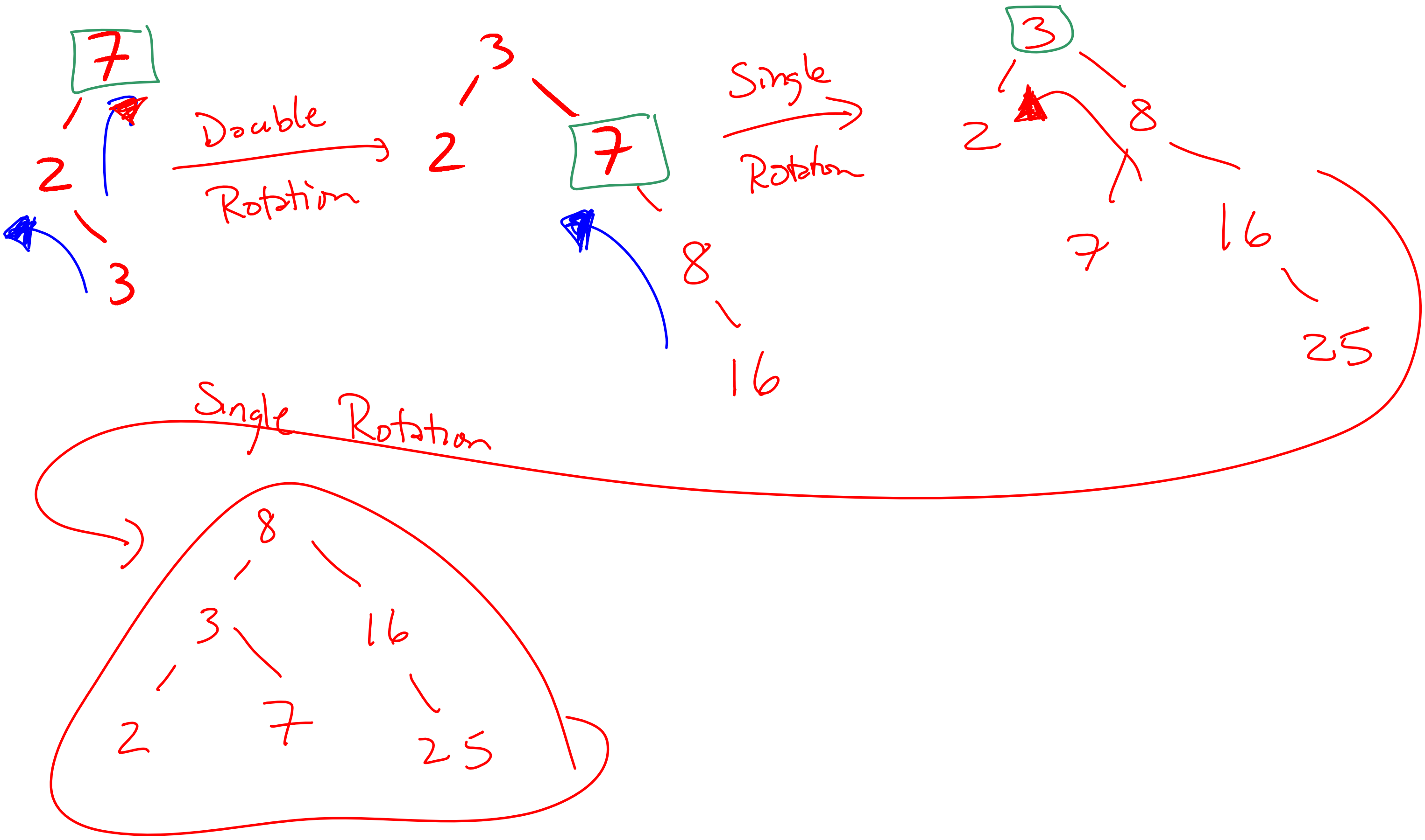
$$= 2^{\log_2 N} \cdot T(1) + \log N \cdot 5N$$

$$= N \cdot 6 + 5 N \log N$$

$$T(N) = 5 \cdot N \log N + 6N$$

4. (8 pts) AVL Trees

Draw the AVL tree that results from inserting the keys 7, 2, 3, 8, 16, 25, in that order into an initially empty AVL tree. You are only required to show the final tree, although if you draw intermediate trees, ***please circle your final result for ANY credit.***



5. (8 pts) AVL Trees

a) (4 pts) What is the minimum and maximum number of nodes in an **AVL tree of height 6?** (Hint: the height of a tree consisting of a single node is 0) ***Give an exact number*** for both of your answers – not a formula.

Minimum = 33 ← $S(h) = S(h-1) + S(h-2) + 1$

Maximum = 127 ← $2^{h+1} - 1 = 2^7 - 1 = 128 - 1$

b) (2 pts) You are given an **AVL tree** of height 6. The minimum and maximum number of *rotations* we might have to do when doing an insert is: (**Give an exact number, not a formula.** A single rotation = 1 rotation, a double rotation = 1 rotation)

Minimum = 0

Maximum = 1

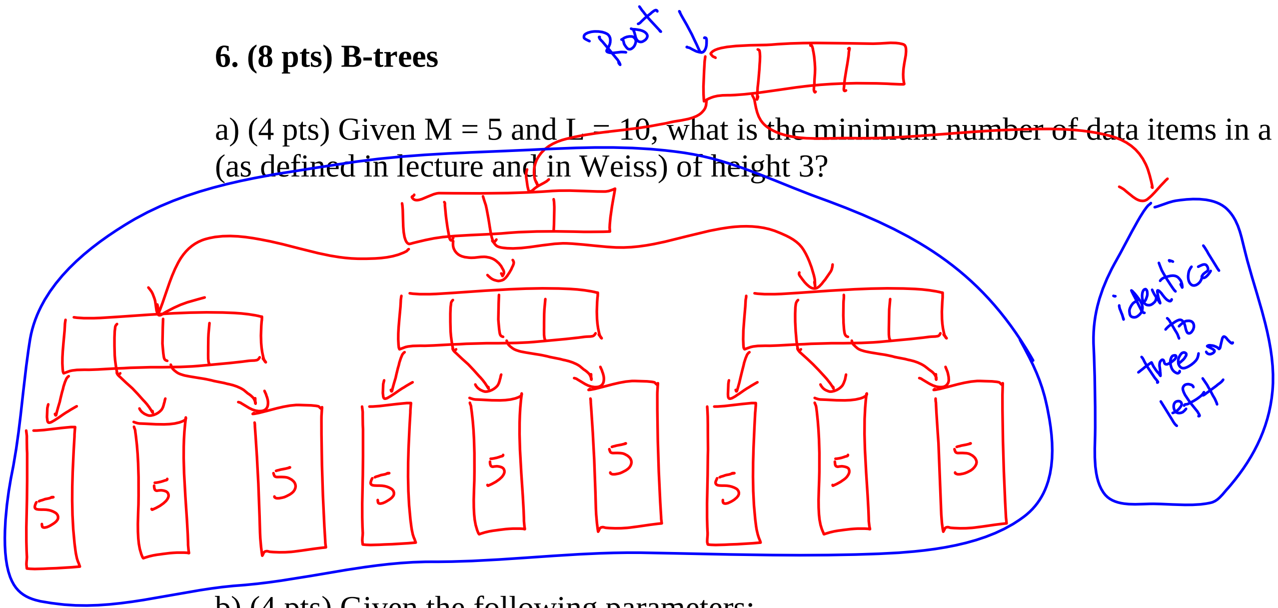
c) (2 pts) Give a ***Pre-order*** traversal of your final tree from the previous question (question 4 in the previous page) here:

8, 3, 2, 7, 16, 25

$n \geq 2 \cdot \lceil M/2 \rceil^{h-1} \cdot \lceil L/2 \rceil = 2 \cdot 3^2 \cdot 5 = 2 \cdot 9 \cdot 5 = 90$

6. (8 pts) B-trees

a) (4 pts) Given M = 5 and L = 10, what is the minimum number of data items in a B-tree (as defined in lecture and in Weiss) of height 3?



90

b) (4 pts) Given the following parameters:

Disk access time = 1 milli-sec per byte

1 Page on disk = 1024 bytes

Key = 16 bytes

Pointer = 8 bytes

Data = 100 bytes per record (includes key)

Assuming you can place things where you want in memory (in other words this is not a question about Java implementation of B-trees), what are the best values in a B-tree for:

M = 43

and

L = $\frac{1024}{100} = \lfloor 10.24 \rfloor = 10$

$1024 \geq (M-1) \cdot 16 + M \cdot 8$

$= 16M - 16 + 8M = 24M - 16$

$1040 = 24M$

$M = \lfloor \frac{1040}{24} \rfloor = 43$

7. (14 pts) Hash Tables

For each of the following versions of hash tables, insert the following elements in this order: 55, 86, 16, 25, 6, 7. For each table, TableSize = 10, and you should use the primary hash function h(k) = h%10. For each table the first column holds the indices and the second can hold values or pointers, depending on the hash-table type. If an item cannot be inserted into the table, please indicate this and continue inserting the remaining values.

a) Separate chaining hash table – use a linked list for each bucket, insert at front

0		
1		
2		
3		
4		
5		→ 25 → 55
6		→ 6 → 16 → 86
7		→ 7
8		
9		

b) Linear probing hash table:

0	7_3	
1		
2		
3		
4		
5	55_0	25_0
6	86_0	16_0 25_1 6_0
7	16_1	25_2 6_1 7_0
8	25_3	6_2 7_1
9	6_3	7_2

c) Quadratic probing hash table

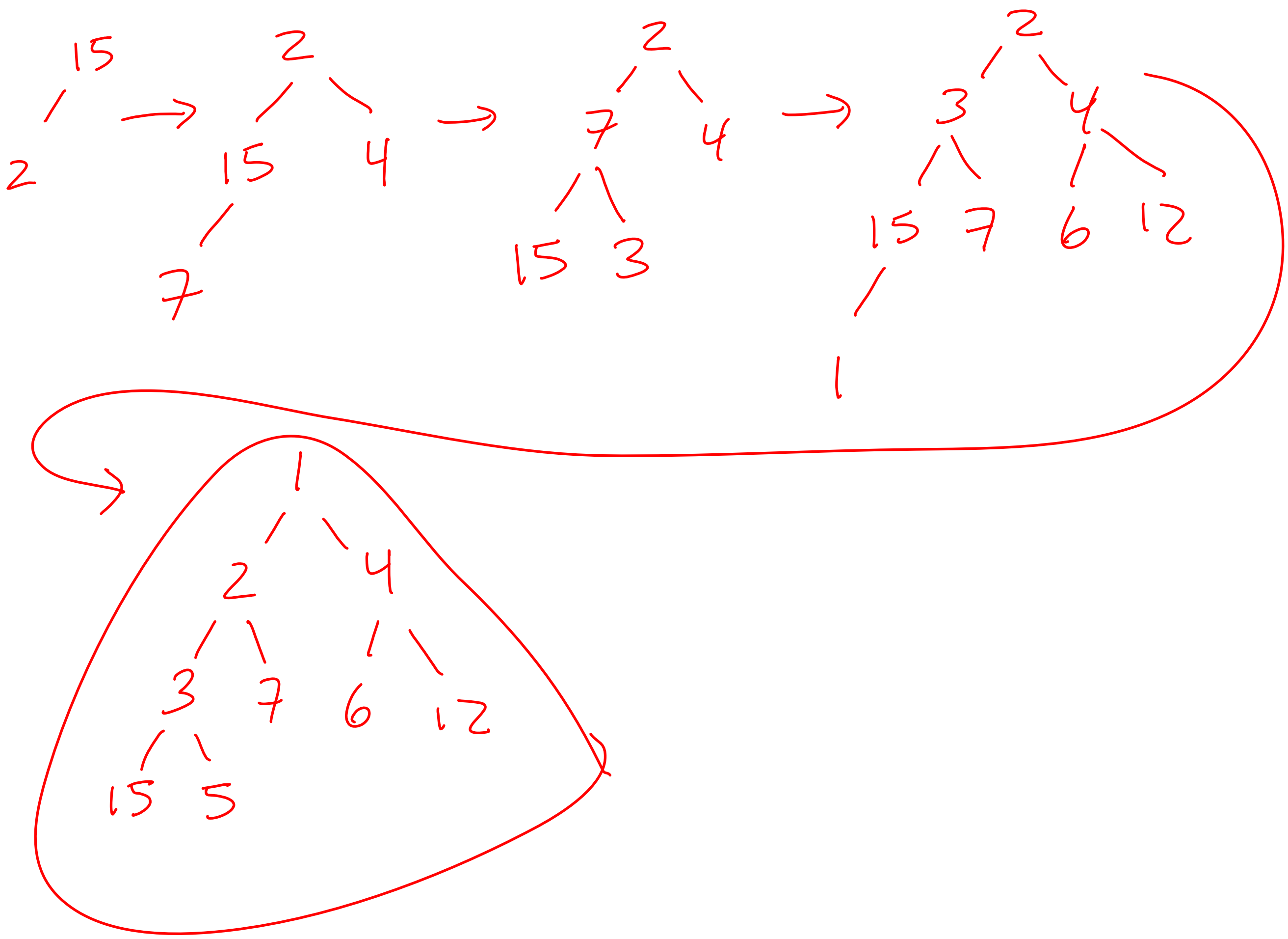
0	6_2	
1		
2		
3		
4		
5	55_0	25_0
6	86_0	16_0 25_1 6_0
7	16_1	6_1 7_0
8	7_1	
9	25_2	

d) What is the load factor in Table a)? $6/10$

e) What is the load factor in Table b)? $6/10$

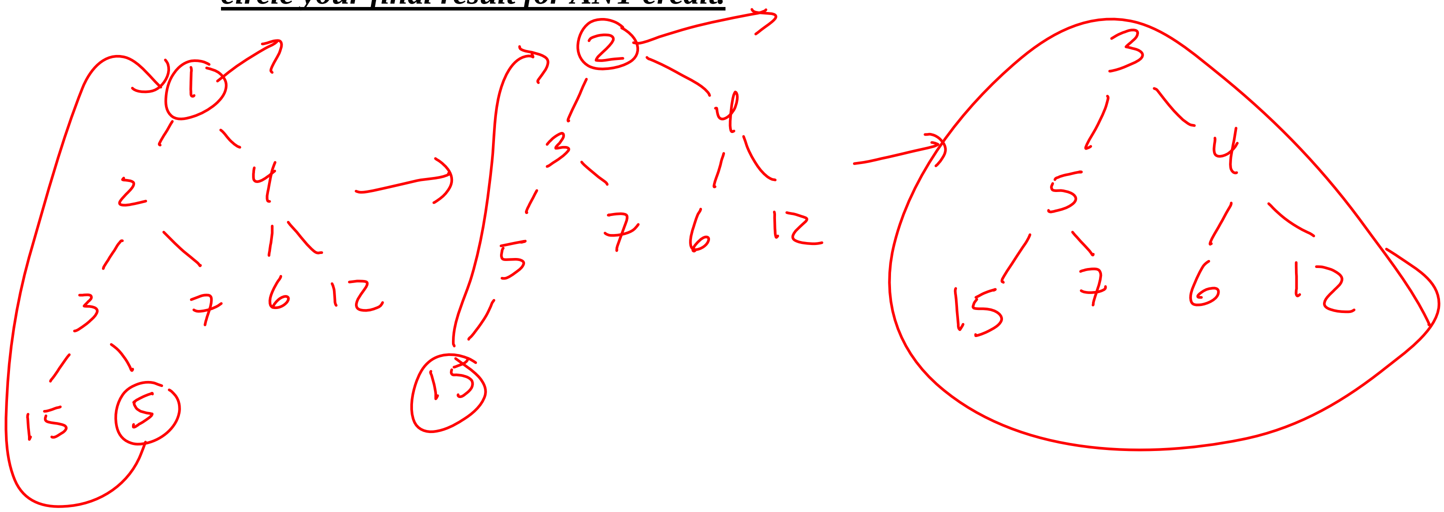
8. (14 pts) Binary Min Heaps

a) (8 pts) Draw the binary min heap that results from inserting 15, 2, 4, 7, 3, 6, 12, 1, 5 in that order into an **initially empty binary heap**. You do not need to show the array representation of the heap. You are only required to show the final tree, although if you draw intermediate trees, ***please circle your final result for ANY credit.***



8. Binary Min Heaps (continued)

b) (4 pts) Draw the result of doing 2 deletemins on the heap you created in part a. You are only required to show the final tree, although if you draw intermediate trees, ***please circle your final result for ANY credit.***



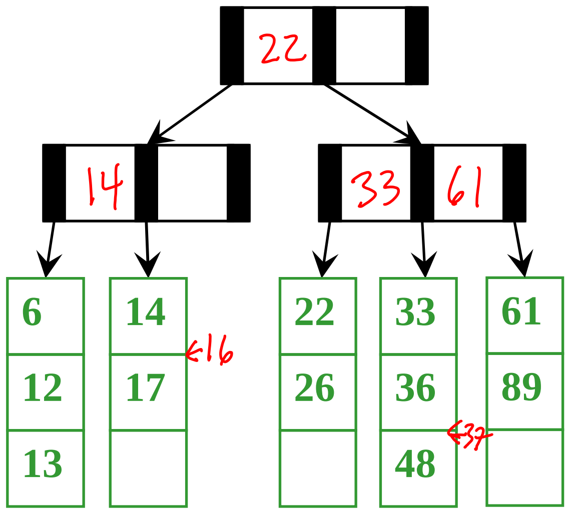
c) (2 pts) Draw the array representation of the tree you drew at the end of step b above (after the two deletemins). You only need to fill in the parts of the array that hold valid data values.

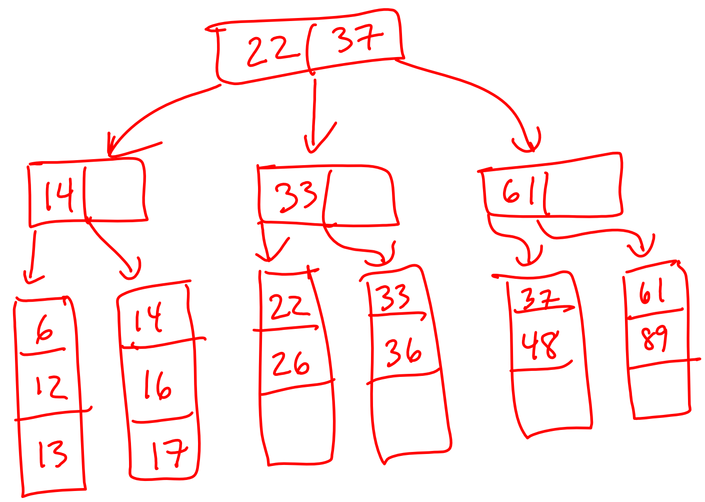
Index	0	1	2	3	4	5	6	7	8	9
Value		3	5	4	15	7	6	12		

9. (10 pts) B-tree Insertion and Deletion

a) (2pts) In the B-Tree shown below with M=3 and L=3, please write in the appropriate values for the interior nodes.

b) (4 pts) Starting with the B-tree shown below, insert 16 and 37 in that order. Draw and circle the resulting tree (*including values for interior nodes*) below. Use the method for insertion described in lecture and in homework 3. (split)





c) (4 pts) Starting with the ***original*** B-tree shown above on the left (***before* inserting 16 and 37**), delete 14 and 89 in that order. Draw and circle the resulting tree (*including values for interior nodes)* below. Use the method for deletion described in lecture. (adopt from sibling, else merge)