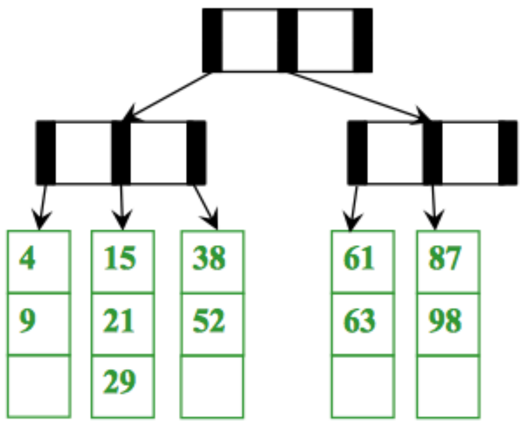
9. (8 pts) B-trees

a) (1 pt) In the B-Tree shown below, **write in the values for the interior nodes**.

ORIGINAL:



After inserting 30:

b) (4 pts) Starting with the **ORIGINAL** B-tree shown above, in the box above, draw the tree resulting after inserting the value 30 (*including values for interior nodes*). Use the method for insertion described in lecture and in the book.

c) (3 pts) Starting with the **ORIGINAL** B-tree shown above, below, draw the tree resulting after deleting the value 87 (*including values for interior nodes*). Use the method for deletion described in lecture and in the book.

After deleting 87:

6. (6 pts) Trees

a) (2 pts) What is the **minimum number of nodes** in an **AVL tree of height 5?** (Hint: the height of a tree consisting of a single node is 0) ***Give an exact number*** not a formula.

b) (4 pts) Given the following parameters:

1 Page on disk = 1000 bytes

Disk access time = 2 milli-secs per byte

Key = 4 bytes

Pointer = 16 bytes

Data = 30 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 =

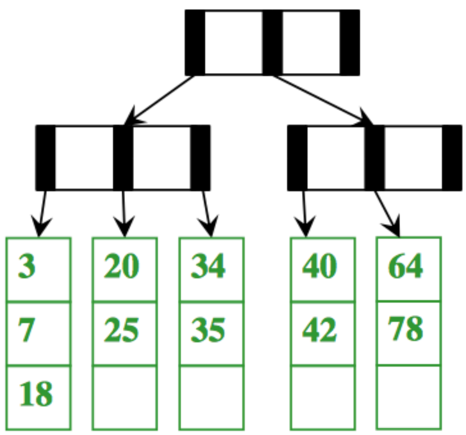
and

L =

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

a) (2pts) In the B-Tree shown below, please write in the appropriate values for the interior nodes.

b) (4 pts) Starting with the B-tree shown below, insert **9**. Draw and circle the resulting tree (*including values for interior nodes*) below. Use the method for insertion described in lecture.



c) (4 pts) Starting with the ***original*** B-tree shown above on the left (***before* inserting 9**), delete **78**. Draw and circle the resulting tree (*including values for interior nodes)* below. Use the method for deletion described in lecture.

4. **(6 pts) Recurrence Relationships -**
Suppose that the running time of an algorithm satisfies the recurrence relationship

$$T(1) = 7.$$

and

$$T(N) = T(N-2) + 4 \quad \text{for odd integers } N > 1$$

Find the closed form for T(N) ***<u>and show your work step by step</u>***. 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).

From 11Wi midterm:

3. (6 pts) Recurrence Relationships -

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

$$T(1) = 6.$$

and

$$T(N) = 2 * T(N/2) + 5N \qquad \text{for integers } N > 1$$

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).

1. (20 pts) Big-Oh

(2 pts each) For each of the operations/functions 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. **Your answer should be as "tight" and "simple" as possible.** For questions that ask about running time of operations, assume that the most efficient implementation is used. For array-based structures, assume that the underlying array is large enough. For questions about hash tables, assume that no values have been deleted (lazily or otherwise).

You do not need to explain your answer.

a) $T(N) = T(N - 1) + 50$ ____________________

b) *Printing all elements in a* ***binary search tree*** *containing N elements from largest to smallest. (worst case).* ____________________

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

d) Finding the maximum value in a ***binary min heap*** *containing N elements. (worst case)* ____________________

e) remove*(k) on a* ***binary min heap*** *containing N elements. Assume you have a reference to the key k that should be removed. (worst case)* ____________________

f) *Creating a* ***binary min heap*** *from the elements in a* ***binary search tree*** *containing N elements (worst case).* ____________________

g) *Dequeue in a* ***FIFOqueue*** *containing N elements implemented using linked list nodes (worst case)* ____________________

h) *Finding the minimum value in an* ***AVL tree*** *containing N elements (worst case)* ____________________

i) $T(N) = 2\, T(N/2) + N$ ____________________

j) $f(N) = \log_{10} (2^N)$ ____________________

2. **(12 pts) Big-Oh and Run Time Analysis:** Describe the worst case running time of the following pseudocode functions in Big-Oh notation in terms of the variable `n`. Your answer should be as "tight" and "simple" as possible. ***Showing your work is not required***

Runtime:

I.
```
void haunted(int n, int sum) {
     for (int i = 1; i < n * n; i++) {
        for (int j = 0; j < n; j += 3) {
           sum++;
        }
     }
     for (int k = 0; k < n * n; k++) {
        sum++;
     }
}
```

II.
```
int sweet(int n, int sum) {
     if (n < 5) {
        return sum;
     } else {
        for (int i = 0; i < n; i++) {
           sum++;
        }
     }
     return sweet(n-2, sum);
}
```

III.
```
int treat(int n) {
     if (n < 10) {
        return n;
     } else if (n < 100) {
        return n + 1;
     }
     return n * treat(n / 2) + treat(n / 2);
}
```

IV.
```
void spooky(int n, int sum) {
     int k = n;
     while (k > 0) {
        for (int i = 0; i < n * n; i++) {
           if (i % 2 == 0) {
              for (int j = 0; j < i; j++) {
                 sum++
              }
           }
        }
        k--;
     }
}
```

3. (15 pts) Big-O, Big Ω, Big Θ

(2 pts each) For parts (a) – (e) circle **ALL** of the items (if any) that are TRUE. You do not need to show any work or give an explanation.

a) $N \log (N^2)$ is:

Ω (N log N) Θ (N $\log^2$N) O(N log N) Θ (N^2)

b) N log (log N) is:

Ω (N log N) O (N log N) O (N^2) Ω (N $\log^2$N)

c) $5^N \log N + N^4$ is:

Ω (N^5) O (N^5) Θ (5^N) Θ (N^N)

d) $N \log N + N \log^3 N$ is:

Ω (N log N^3) O (N^2) O (N log N) Θ (N log N^3)

e) $N^{1/2} + \log N$ is:

Ω (log N^3) O ($N^{1/2}$) O (log N) Ω ($N^{1/3}$)

From 13Wi midterm:

5. (8 pts) Draw the AVL tree that results from inserting the keys 1, 4, 6, 8, 9, 5, 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.***

From 11Wi midterm:

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 =

Maximum =

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 =

Maximum =