Name: ____________________

UWNetID: ____________________

CSE 332 Danny & Lucas' Midterm Prep Problems

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 so far.

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

Good Luck!

Total: 100 points. Time: 60 minutes.

Question	Midterm	Category
1	15 Winter	Big O
2	15 Winter	Big O
3	15 Winter	Big O
4	14 Spring	Recurrence
4	18 Autumn	Heaps
8	18 Winter	B-Trees
9	18 Winter	B-Trees

Name: ___________________________________

1. (18 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) *Pop in a* ***stack*** *containing N elements implemented using an array (worst case)* ________________

b) *Find in an* ***open addressing hash table*** *containing N elements where linear probing is used to resolve collisions (worst case). Tablesize =* N^2 ________________

c) *Merging two* ***binary min heaps*** *containing N elements each. (worst case)* ________________

d) *Determining what the 10 largest items are in an* ***open addressing hash table*** *containing N elements where quadratic probing is used to resolve collision (worst case). Tablesize = 2*N.* ________________

e) $T(N) = T(N/2) + 100$ ________________

f) $f(N) = \log \log (N + N) + \log^2 N$ ________________

g) *Insert in a* ***separate chaining hash table*** *containing N elements where each bucket points to a* sorted *linked list (worst case) Tablesize =* N^2 ________________

h) $f(N) = N \log^2 N + N^2 \log N$ ________________

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

Name: ________________________________

2. (8 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 happy (int n, int sum) {
         int k = 1;
         while (k < n) {
            for (int i = 0; i < k; i++) {
               sum++;
            }
            k++;
         }
         for (int j = n; j > 0; j--) {
               sum++;
         }
     }
```

```
II.  int smiley (int n) {
         if (n < 5)
            return n * n;
         else {
            for (int i = 0; i < 10,000; i++) {
              print i
            }
            return smiley (n / 2);
         }
     }
```

```
III. void sunny (int n, int sum) {
         for (int i = 1; i < n * n; i++) {
            if (sum > 10) {
               for (int j = 0; j < n; j++) {
                  sum++;
               }
            } else {
               sum++;
            }
         }
     }
```

```
IV.  void funny (int n, int sum) {
         for (int i = 0; i < n * n; i++) {
            if (i % 10 == 0) {
               for (int j = 0; j < i; j++) {
                  sum++
               }
            }
         }
     }
```

Name: ___________________________________

3. (10 pts) Big-O, Big Ω, Big Θ

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

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

$\Omega(N^2)$ $O(N^2)$ $\Theta(N^2)$ None of these

b) $N \log N + \log(\log N) + 300$ is:

$\Omega(\log N)$ $O(\log N)$ $\Theta(\log N)$ None of these

c) $N^2 \log N + N^4$ is:

$\Omega(N^3)$ $O(N^5)$ $\Theta(N^4)$ None of these

d) $N \log^2 N + N^2 \log N$ is:

$\Omega(N^2)$ $O(N^2)$ $\Theta(N^2 \log N)$ None of these

e) If f(n) is O(g(n)) and g(n) is O(h(n)), which of the following is correct? (**circle all that are true**)

i. f(n) * g(n) is O(f(n) * g(n))

ii. f(n) + g(n) is O(min(g(n), h(n)))

iii. f(n) is O(h(n))

iv. h(n) is O(f(n))

v. none of the above

4. (6 pts) Recurrence Relationships -

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

$T(1) = 6.$

and

$T(N) = T(N-1) + N$ 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).

4. (8 pts) 3 Heaps

Given a 3-heap of height h, what are the minimum and maximum number of nodes in the **middle sub-tree** of the root? Give your answer in closed form (there should not be any summation symbols).

Min nodes in middle sub-tree:

Max nodes in middle sub-tree:

8. (6 pts) B-Trees

Given M=5 and L=15, what is the minimum and maximum number of **<u>pointers</u>** in a B-Tree (as defined in lecture and in Weiss) of height 5? By pointers, we mean pointers that point to an interior or leaf node. Do not count the pointer that points to the root node. **Give a single number or a single number with an exponent for your answers, not a formula. <u>For any credit, explain briefly how you got your answers.</u>**

Minimum number of pointers: _______________

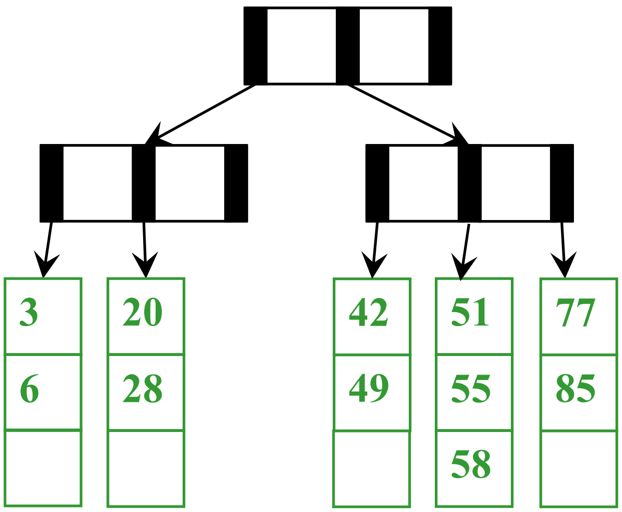
Maximum number of pointers: _______________

Explanation:

9. (7 pts) B-trees

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

ORIGINAL:



After inserting 60:

b) (3 pts) Starting with the **ORIGINAL** B-tree shown above, in the box above, draw the tree resulting after inserting the value 60 (*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 28 (*including values for interior nodes*). Use the method for deletion described in lecture and in the book.

After deleting 28: