CSE 332 Midterm Exam

Summer 2024 [sample]

Name ____

Net ID ______________________ (@uw.edu)

Academic Integrity: You may not use any resources on this exam except for writing instruments, your own brain, and the exam packet itself. This exam is closed notes, closed neighbor, closed electronic devices, etc.. The last two pages of this exam provide a list of potentially helpful identities as well as room for scratch work (respectively). No markings on these last two pages will be graded. Your answer for each question must fit in the answer box provided.

Instructions: Before you begin, **Put your name and UW Net ID at the top of this page. Your Net ID is what is used for your @uw.edu email address (not to be confused with your student ID, which is printed on your Husky card).** Make sure that your name and ID are LEGIBLE. Please ensure that all of your answers appear within the boxed area provided.

Section	**Max Points**
ADTs and Data Structures	6
Asymptotic Analysis	20
Priority Queues and Heaps	11
AVL Trees	11
Algorithms	12
Extra Credit	(+1.5)
Total	60

Section 1: ADTs and Data Structures

(3 pts)**Question 1: For each of the following data structures provided indicate which abstract data type it satisfies.**

1) Binary Heap:

2) AVL Tree:

3) Circular Array:

(3 pts)**Question 2: For this question we give a data structure and an English description of an algorithm. Give the name of the abstract data type operation that we described.**

1) Circular Array: set `x` to be the value at index `front`, then increment front and modulus by the array's length, finally return `x`.

2) Binary Min Heap: Move the item at index `size` − `1` to index 0. Percolate down from index 0.

3) AVL Tree: If the current node is Null, return Null. If the given key is equal to the current node's key, return the current node's value. If the current node's key is larger than the given key, recursively call this algorithm on the left child, otherwise recursively call this algorithm on the right child.

Section 2: Asymptotic Analysis

(4 pts)**Question 3: Give a simplified Θ bound on the best and worst case running times for the given code. (By simplified we mean it should contain no constant coefficients or non-dominant terms.) You should assume that no strings in the input list are null.**

```
int nonsense(List<String> stuff){
    int n = stuff.size();
    int result = 0;
    for(int i = n-1; i > 0; i--){
        if(stuff[i].equals("hello"){
            for(int j = i; j < n; j++){
                result++;
            }
            return result;
        }
        else{
            result += i;
        }
    }
    return result;
}
```

1) Best Case: $\Theta\left(\boxed{} \right)$

2) Worst Case: $\Theta\left(\boxed{} \right)$

(6 pts)**Question 4: For each of the expressions below, give a simplified Θ bound.**

1) $f(n) = 12n\log_2 n + 8n^2 - 16n$

$f(n) \in \Theta\left(\boxed{} \right).$

2) $f(n) = \log_4(2^{3n}) + \log_5(3^{2n})$

$f(n) \in \Theta\left(\boxed{} \right).$

3) $f(n) = 0.5^{2n} + 2^n$

$f(n) \in \Theta\left(\boxed{} \right).$

(6 pts)**Question 5: Show that $n^3 + 8n^2 + 100n$ belongs to $O(n^3)$.**

(4 pts)**Question 6: Suppose that the running time of an algorithm is expressed by the recurrence relation:**

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

$$T(1) = 5$$

For the following questions, use *the tree method* to solve the recurrence relation. We have broken up the process into subquestions to guide you through your answer. You may assume that n is always a power of 4.

1) Sketch the first three levels of the tree in space below. Include at least the first 3 levels of the tree (i.e. the root, its children, and its grandchildren), make clear what the input size is for each recursive call as well as the work per call.

2) Indicate exactly the total amount of work done at level i of the tree (define the root to be level 0). Include all constants and non-dominant terms.

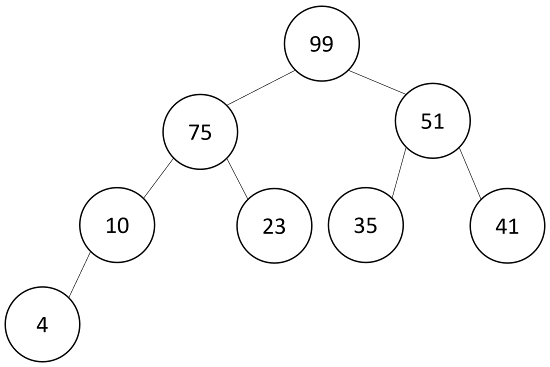
3) Indicate the level of the tree in which the base cases occur.

4) Give a simplified Θ bound on the solution. (You may find the log rules on page 12 helpful.)

$\Theta\left(\boxed{}\right)$

Section 3: Priority Queues and Heaps

The next three questions relate to the given binary max heap.



(2 pts)**Question 7: Suppose that we add a node with priority x as the right child of the node with priority 10. Which of the following best describes which values of x result in the given heap being valid? Write the letter of your answer in the box provided.**

A. $x \geq 10$

B. $x \leq 10$

C. $x > 10$

D. $x < 10$

E. $10 \leq x \leq 75$

F. $4 \leq x \leq 10$

G. None of the above

(2 pts)**Question 8: Give the array representation of the original heap given above. Place the root at index 0 of the array.**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

(5 pts)**Question 9: Perform deleteMax on the heap and give the array representation of the resulting heap. Place the root at index 0 of the array.**

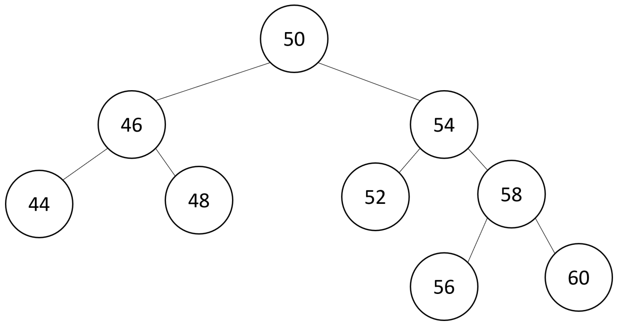
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

(2 pts)**Question 10: Answer "True" or "False":**

If the index of item x is smaller than the index of item y in a min heap, then the priority of item x must be less than or equal to the priority of item y.

Section 4: AVL Trees

(6 pts)**Question 11: Answer the following questions about the AVL Tree below. Each question should be considered completely independently (i.e. "reset" to the image between questions).**



1) Give a integer key which, when inserted into the given AVL tree, would cause a double rotation.

2) Insert the key 61 into the original tree, what key will become the parent of the node with key 54?

3) Insert the key 60 into the original tree drawn above, identify the number of rotations that will occur.

(3 pts)**Question 12: For each traversal below, performed on the original tree above, indicate the keys of first and last nodes processed.**

Traversal	First Node's Key	Last Node's Key
Pre-Order		
In-Order		
Post-Order		

(2 pts)**Question 13: Answer "True" or "False":**

The smallest item in an AVL tree is always a leaf.

Section 5: Algorithms

(12 pts)**Question 14: For each data structure below, describe an algorithm which returns its second smallest item, then give the asymptotic running time of your algorithm. By second smallest item we mean either the item with the second smallest key for dictionaries or the second smallest priority value for priority queues (e.g. Min heaps keep the smallest priority values near the root). Make the algorithm as efficient as you can (asymptotically). You may assume the size of the data structure is greater than 2.**

1) Binary Min Heap:

Asymptotic Running Time: $\Theta\left(\boxed{\quad}\right)$.

2) Binary Max Heap:

Asymptotic Running Time: $\Theta\left(\boxed{\quad}\right)$.

problem continues onto the next page.

3) AVL Tree:

Asymptotic Running Time: $\Theta\left(\boxed{}\right)$.

Section 6: Extra Credit

(+0.75 pts)**Question Extra 1: In the space below, draw a picture of how you were feeling coming into this exam.**

(+0.75 pts)**Question Extra 2: Now draw a picture of how you are feeling coming out of it.**

Identities

Nothing written on this page will be graded.

Summations

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \text{ for } |x| < 1$$

$$\sum_{i=0}^{n-1} = \sum_{n}^{i=1} = n$$

$$\sum_{i=0}^{n} i = 0 + \sum_{n}^{i=1} i = \frac{n(n+1)}{2}$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}$$

$$\sum_{i=0}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2 = \frac{n^4}{4} + \frac{n^3}{2} + \frac{n^2}{4}$$

$$\sum_{i=0}^{n-1} x^i = \frac{1-x^n}{1-x}$$

$$\sum_{i=0}^{n-1} \frac{1}{2^i} = 2 - \frac{1}{2^{n-1}}$$

Logs

$$x^{\log_x(n)} = n$$

$$a^{\log_b(c)} = c^{log_b(a)}$$

$$\log_b(a) = \frac{\log_d(a)}{\log_d(b)}$$

Scratch Work

Nothing written on this page will be graded.