CSE 332 Midterm Exam

Winter 2024

Name Answer Key

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	14
Priority Queues and Heaps	10
AVL Trees	11
B Trees	8
Tries	5
Algorithms	12
Extra Credit	(+1.5)
Total	66

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: Priority Queue

2) AVL Tree: Dictionary

3) Trie: Dictionary

(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`.

   Dequeue

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

   deleteMin

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.

   Find

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: $\Theta($ 1 $)$

2) Worst Case: $\Theta($ n $)$

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

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

$f(n) \in \Theta($ n^2 $)$.

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

$f(n) \in \Theta($ n $)$.

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

$f(n) \in \Theta($ 2^n $)$.

(4 pts)**Question 5: Suppose that the running time of an algorithm is expressed by the recurrence relation:**

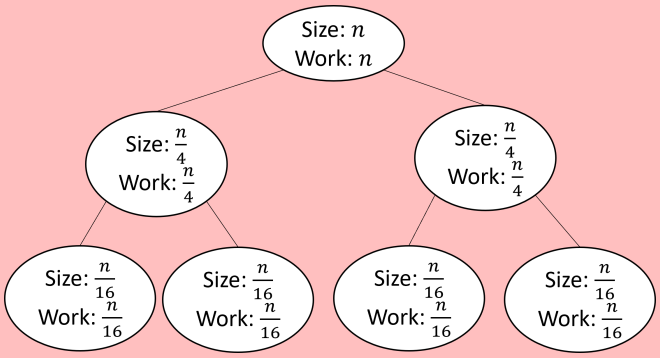
$$T(n) = 2 \cdot T\left(\frac{n}{4}\right) + 5$$
$$T(1) = 5$$

For the following questions, use *either the tree method or unrolling* 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. Each part may have different instructions depending on the method you select

1) **Tree Method:** 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.

   Unrolling: Show the result of the first 2 substitutions of the recurrence relation.

Tree:



Unrolling:
$T(n) = 4T(n/16) + 15$
$T(n) = 8T(n/64) + 35$

2) **Tree Method:** 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.

   Unrolling: Write the recurrence after i total substitutions

Tree: $5 \cdot 2^i$
Unrolling: $T(n) = 2^i T(\frac{n}{4^i}) + 5 \cdot \sum_{j=0}^{i} 2^j$ or $T(n) = 2^i T(\frac{n}{4^i}) + 5(2^{i+1} - 1)$

3) **Tree Method:** Indicate the level of the tree in which the base cases occur.

   Unrolling: Write the number of substitutions needed to reach the base case.

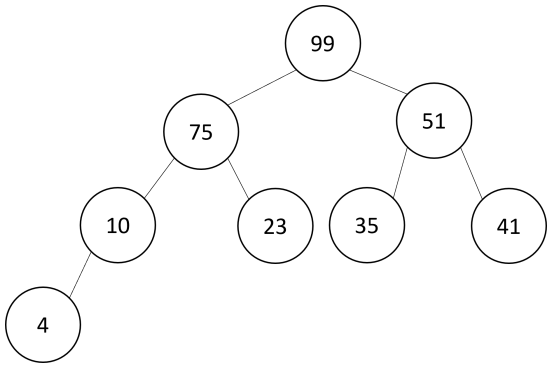
$\log_4(n)$

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

$\Theta\left(\sqrt{n}\right)$

Section 3: Priority Queues and Heaps

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



(2 pts)**Question 6: 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

B

(3 pts)**Question 7: 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
99	75	51	10	23	35	41	4							

(3 pts)**Question 8: 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
75	23	51	10	4	35	41								

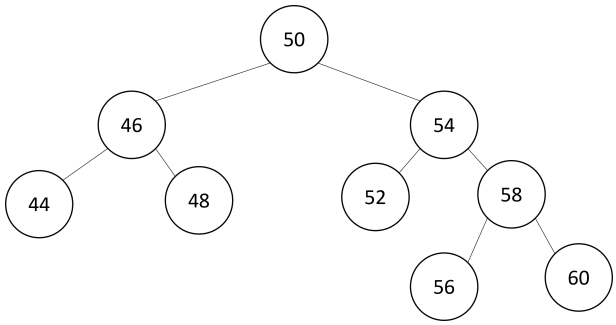
(2 pts)**Question 9: Answer "True" or "False":**

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.

False

Section 4: AVL Trees

(6 pts)**Question 10: 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.

55 or 57

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

58

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

0

(3 pts)**Question 11: 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	50	60
In-Order	44	60
Post-Order	44	50

(2 pts)**Question 12: Answer "True" or "False":**

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

False

Section 5: B Trees

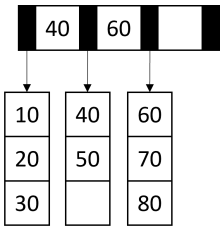
(2 pts)**Question 13: Suppose we had a B Tree with $L = 4$ and $M = 5$ that has height 3. What is the *maximum* number of key-value pairs in the tree? (Recall that we define height to be the maximum number of edges/"hops" between the root and a leaf)**

500

(2 pts)**Question 14: Suppose we had a B Tree with $L = 4$ and $M = 5$ that has height 3. What is the *minimum* number of key-value pairs in the tree?**

36

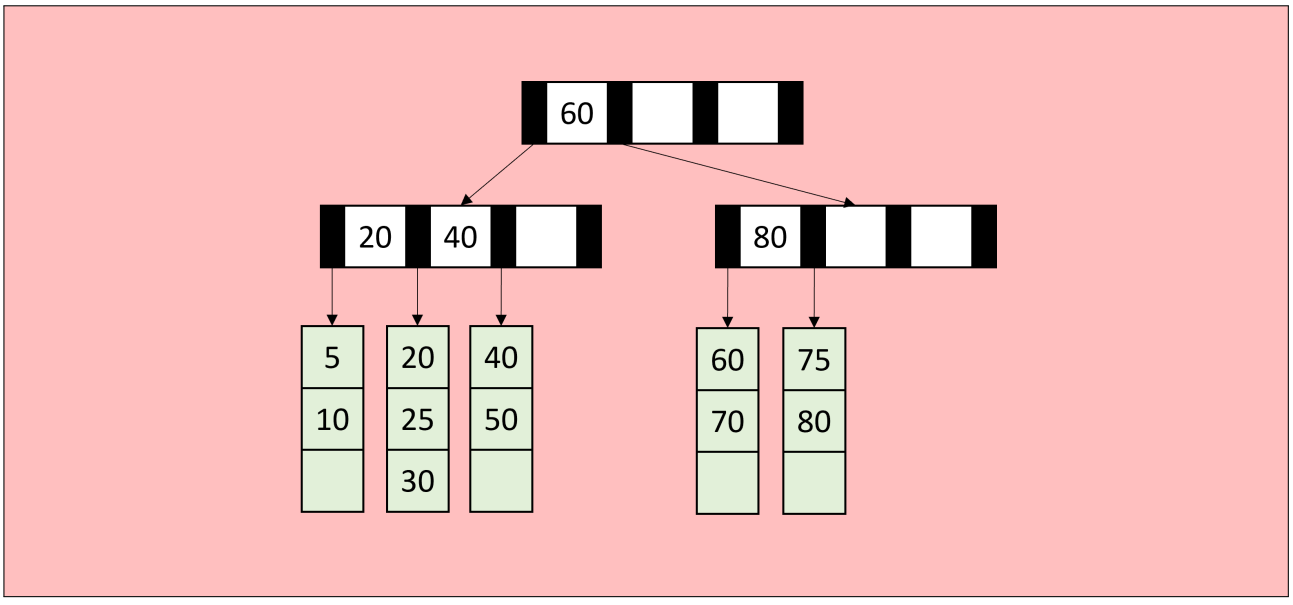
(4 pts)**Question 15: Answer the following questions about the B tree below.**



1) What is L? 3

2) What is M? 4

3) Insert the keys $5, 25, 75$ in that order, then draw the resulting B tree below



Section 6: Tries

(5 pts)**Question 16: Perform the operations below starting with an empty trie, then sketch the resulting trie. You should include the key and value associated with each node in the trie.**

First, insert the following key-value pairs:

– key="nate", value=1

– key="name", value=2

– key="nat", value=3

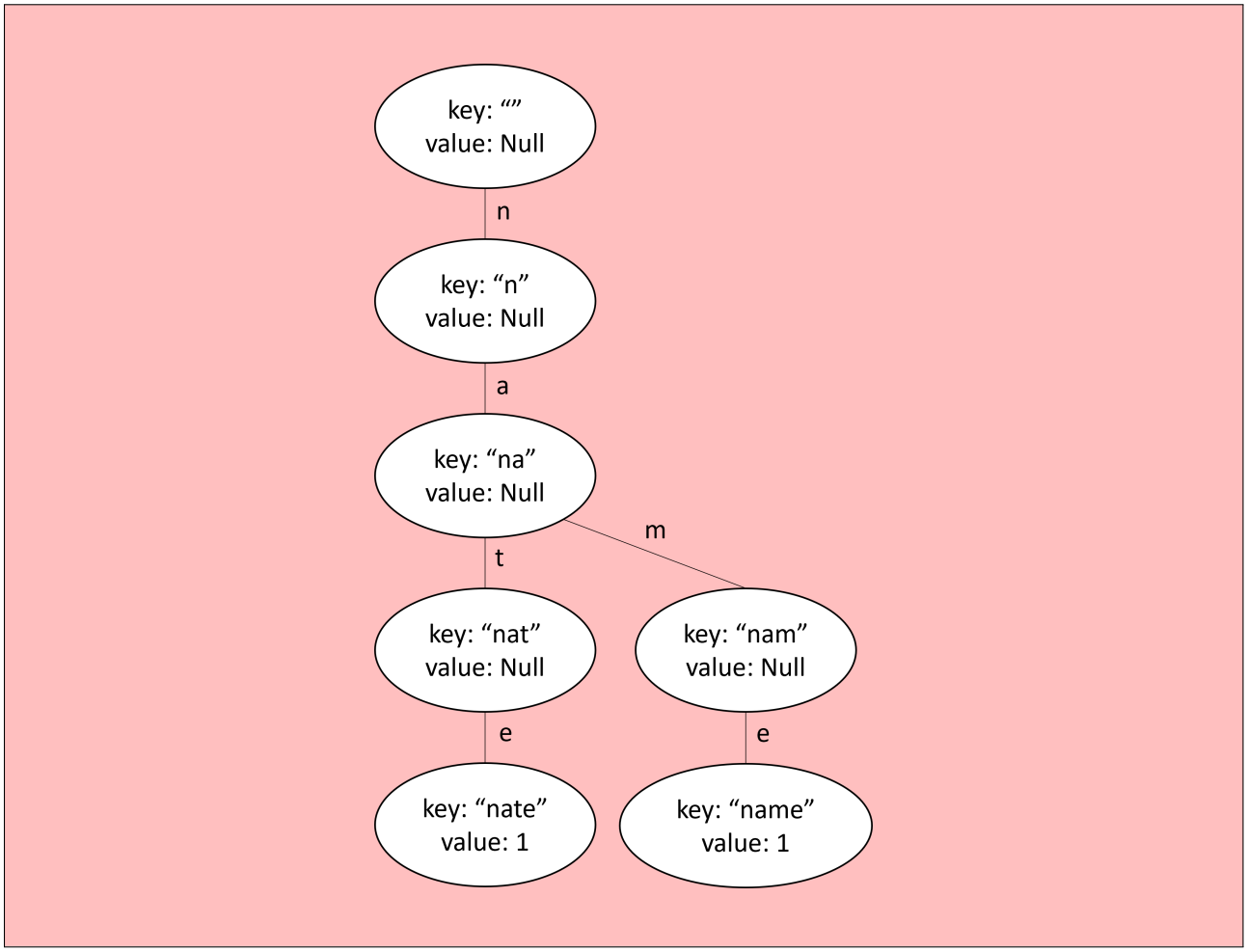
– key="name", value=1

Next delete the following keys:

– "nat"

– "e"

Now sketch the final trie.



Section 7: Algorithms

(12 pts)**Question 17: 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 second smallest priority value for priority queues. 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:

Return the smallest child of the root.

Asymptotic Running Time: $\Theta\left(\boxed{1}\right)$.

2) Binary Max Heap:

Scan through the entire array while recording the smallest two values. Return the second smallest.
(Asymptotically-matching alternative solutions: The second smallest will be a leaf or parent of a leaf, so it's sufficient to check just the last $\lceil \frac{3n}{4} \rceil$ items. Or you could find the smallest 2 leaves, then return the smaller of the second smallest leaf and the parent of the smallest leaf. You could also copy everything over to a new array, then call buildheap on that new array to convert into a minheap, then use the algorithm above.)

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

problem continues onto the next page.

3) B Tree with $M = 5$ and $L = 20$:

navigate to the left-most leaf node by following the left-most child at each internal node, then return the item at index 1.

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

4) AVL Tree:

Navigate to the leftmost node in the tree. If it's not a leaf, return the leftmost item in the right subtree. If it is a leaf, return the leftmost item in its parent's right subtree (which is the parent itself if it has no right child).

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

Section 8: 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.