

CSE 332 Midterm Exam

Autumn 2023

Name _____ Answer Key _____

Net ID _____

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.** 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	10
Asymptotic Analysis	16
Priority Queues and Heaps	14
AVL Trees	10
B Trees	10
Tries	5
Extra Credit	(+1.3)
Total	65

Section 1: ADTs and Data Structures

(4 pts) **Question 1:** For each of the following, indicate whether it is a Data Structure or an Abstract Data Type by writing DS or ADT (respectively) in the box provided.

(a) Trie: DS

(b) Heap: DS

(c) Dictionary: ADT

(d) List: ADT

(2 pts) **Question 2:** Alaska Airlines wants to build a system for managing calls to the customer support call center. The system allows users to leave a phone number, and the next available agent will later return the customers' calls in the order received. What ADT below should the system use to store the phone numbers? Indicate the letter of your choice in the box.

- A. Stack
- B. Priority Queue
- C. Queue
- D. Dictionary

C. Queue

(4 pts) **Question 3:** In 1 or 2 sentences, describe one reason why you might use a B Tree over an AVL Tree.

Your data structure will have a large number of entries, causing memory access to be a concern, since B-Trees make better use of cache.

Section 2: Asymptotic Analysis

(4 pts) **Question 4:** 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.

```
String spooky(List<String> tricksOrTreats){
    int n = tricksOrTreats.size();
    for(int i = 0; i < n; i += 2){
        if(tricksOrTreats[i].equals("trick")){
            for(int j = i+1; j < n; j++){
                tricksOrTreats[j] = "treat";
            }
        }
        else{
            tricksOrTreats[i] = "treat";
        }
    }
    return "BOO!";
}
```

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

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

(6 pts) **Question 5:** For each of the expressions below, give a simplified Θ bound.

(a) $T(n) = 2T(n/2) + 12$, given $T(1) = 1$

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

(b) $f(n) = 2(\log(n))^2 + 4 \log(n^2)$

$$f(n) \in \Theta\left(\boxed{(\log n)^2}\right).$$

(c) $g(n) = 7 \cdot 2^{3n} + 5 \cdot 3^{2n}$

$$g(n) \in \Theta\left(\boxed{9^n}\right).$$

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

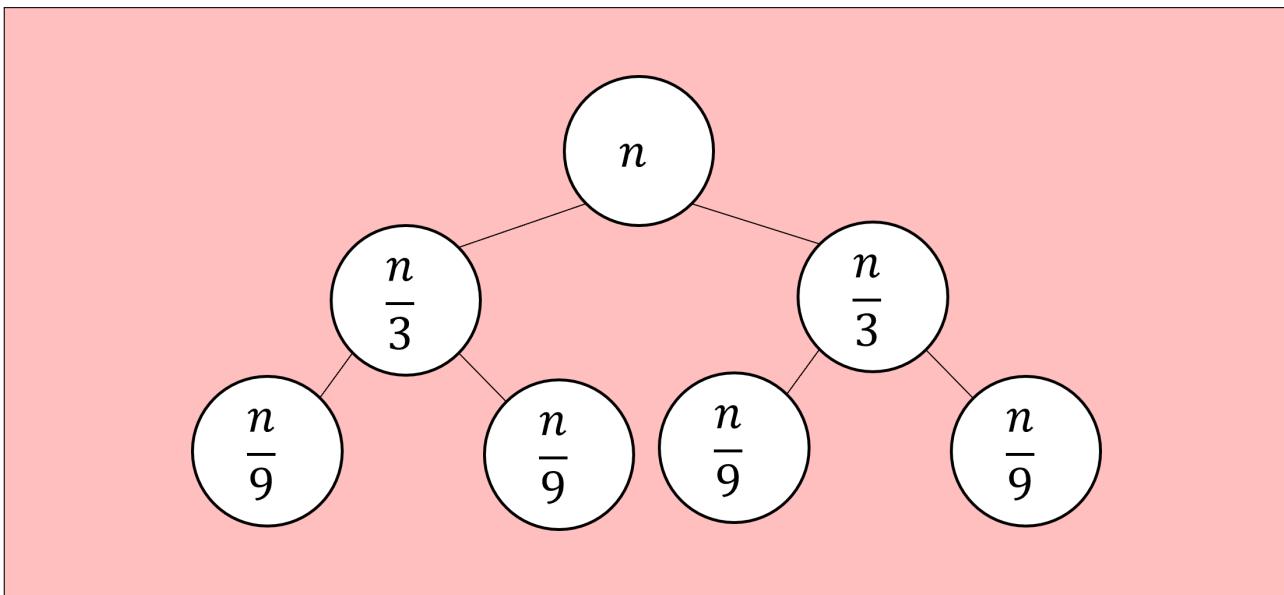
$$T(n) = 2 \cdot T\left(\frac{n}{3}\right) + c \cdot n$$

with the base case:

$$T(1) = c$$

For the following questions you must 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 3.

- (a) Begin by sketching the tree in space below. For full credit, you must include at least the first 3 levels of the tree (i.e. the root, its children, and its grandchildren), and it should be clear what the input size is for each recursive call.



- (b) 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.

$$c \left(\frac{2}{3}\right)^i n$$

- (c) Indicate the level of the tree in which the base cases occur.

$$\log_3(n)$$

- (d) Give the total work across the base case level (i.e. the leaves of the tree). Include all constants and non-dominant terms.

$$c \cdot 2^{\log_3 n}$$

- (e) Give a summation to express the total work. Include all constants and non-dominant terms.

$$c \cdot n \cdot \sum_{i=0}^{\log_3 n} \left(\frac{2}{3}\right)^i$$

- (f) Finally, give a simplified Θ bound on the solution to the sum. (You do not need to show your work.)

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

Section 3: Priority Queues and Heaps

(6 pts) **Question 7:** Answer the following questions about the given binary heap. Note that we have not identified the priority of the item at index 4 and it instead has x as a placeholder. Indices 12, 13, 14, and 15 are not yet occupied.

The empty space below is available if you would like to draw the heap as a binary tree. This drawing is optional and it will not be graded.

7	12	9	15	x	11	22	16	18	35	14	31				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(a) What is the smallest integer value for x which would cause the heap property to hold?

12

(b) Give an integer priority that, when added to the heap, would result in exactly 2 swaps during the percolate up procedure.

7 or 8

(c) Suppose the priority x was changed to be 0 in the original tree provided, what would be found at index 4 after the heap property was restored using percolate up/down?

12

(2 pts) **Question 8:** Given a minHeap containing n items, what is the worst case running time of the most efficient algorithm for finding (without removing) the minimum item, asymptotically?

$$\Theta\left(\boxed{1}\right)$$

(2 pts) **Question 9:** Given a minHeap containing n items, what is the worst case running time of the most efficient algorithm for finding (without removing) the maximum item, asymptotically?

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

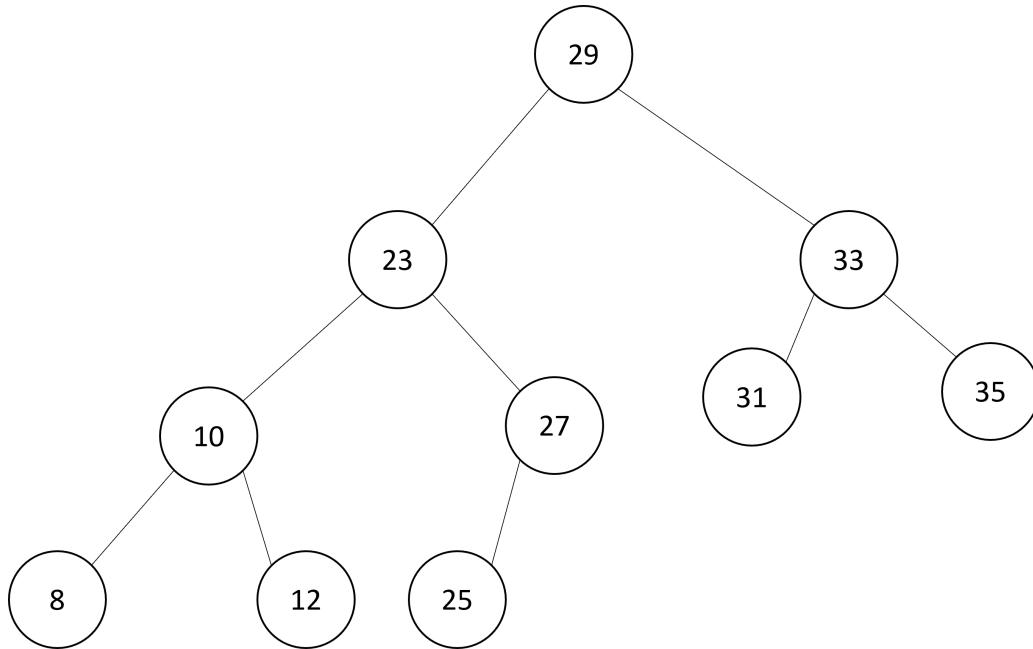
(4 pts) **Question 10:** Describe (in 1 or 2 sentences) a $\Theta(1)$ algorithm for finding (without removing) the second smallest item in a minHeap. You may assume the heap contains at least 2 elements.

return the minimum of the left and right children of the root.

(The remainder of this page intentionally left blank)

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).



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

26

- (b) Insert the key 6 into the original tree drawn above, what node will become the root of the tree?

23

- (c) Insert the key 11 into the original tree drawn above, identify the parent of the node with key 27.

29

(4 pts) **Question 12:** Describe (in 1 or two sentences) a $\Theta(n)$ running time algorithm which prints all elements of a given non-empty AVL Tree in descending order (item with the largest key printed first, smallest key last).

An in-order traversal that first processes the right child.

Section 5: B Trees

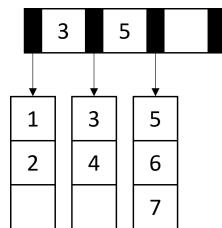
(2 pts) **Question 13:** Suppose we had a B Tree with $L = 6$ and $M = 10$ that contains 60 items. What is the minimum height of this tree? (Recall that we define height to be the maximum number of edges between the root and a leaf)

1

(2 pts) **Question 14:** Suppose we had a B Tree with $L = 6$ and $M = 10$ that contains 60 items. What is the maximum height of this tree?

2

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



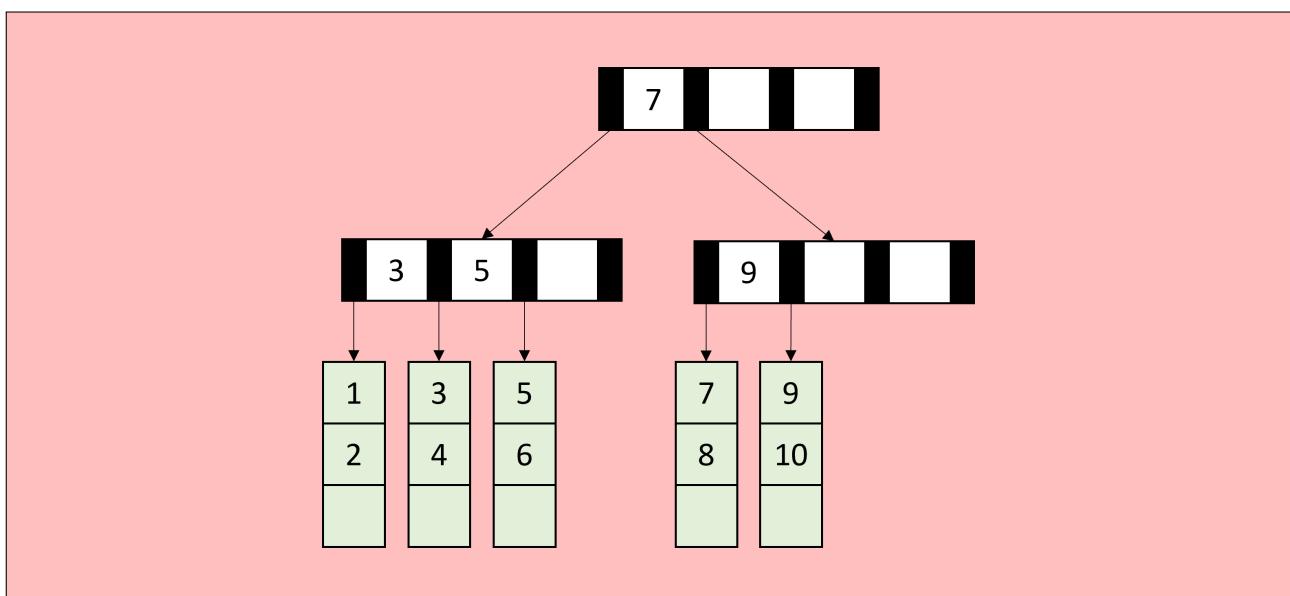
(a) What is L ?

3

(b) What is M ?

4

(c) Insert the keys 8, 9, 10 in that order, then draw the resulting 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.

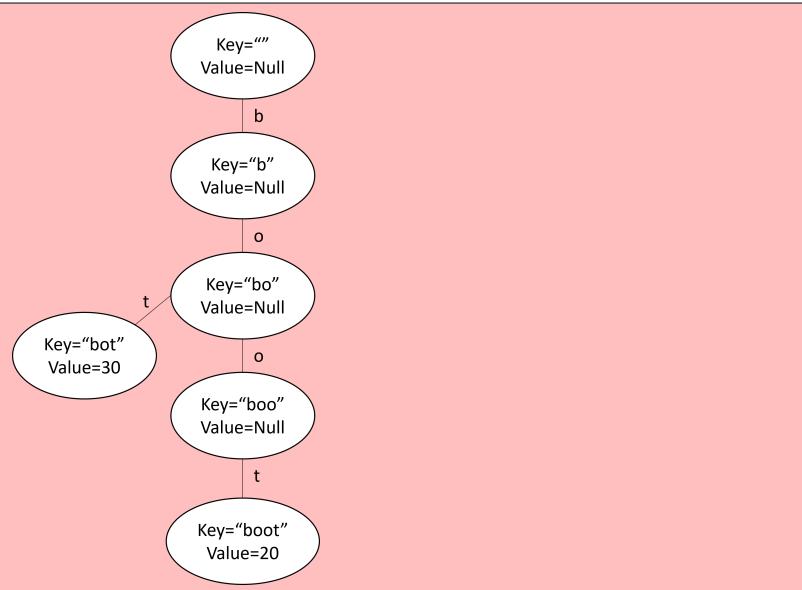
First, insert the following key-value pairs:

- key="boo", value=10
- key="boot", value=20
- key="bot", value=30

Next delete the following keys:

- "boo"
- "b"

Now sketch the final trie.

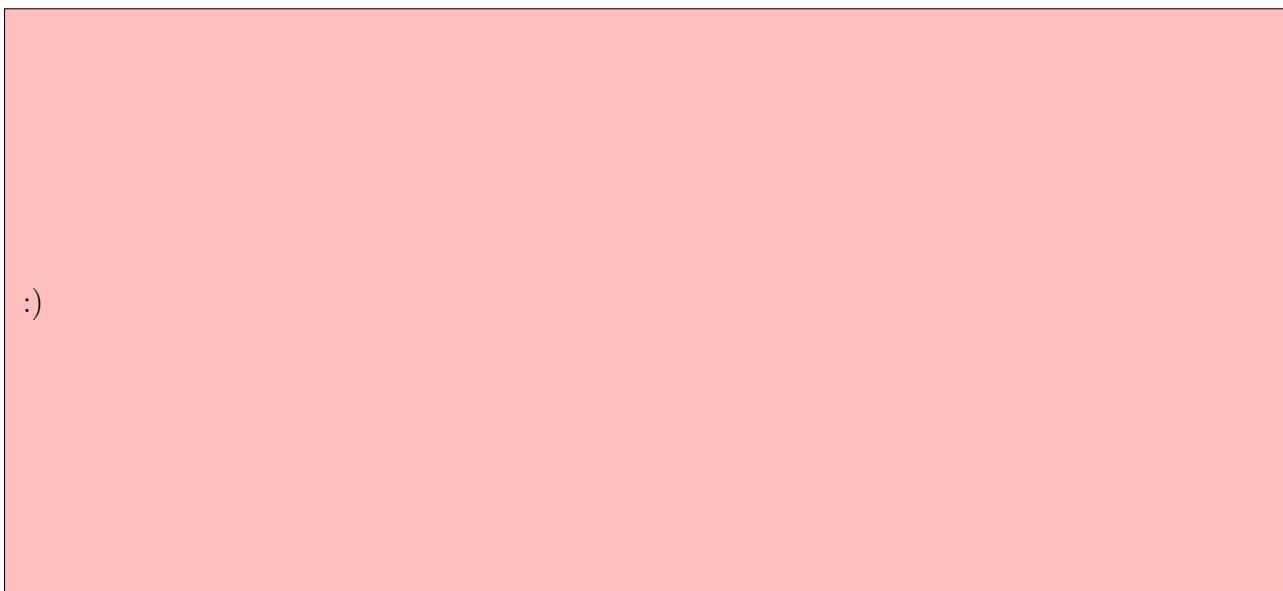


Section 7: Extra Credit

(+0.65 pts) **Question Extra 1:** In the space below, draw a picture of how you were feeling coming into this exam.

A large light red rectangular area for drawing, with a small black dollar sign (\$) symbol in the bottom-left corner.

(+0.65 pts) **Question Extra 2:** Now draw a picture of how you are feeling coming out of it.

A large light red rectangular area for drawing, with a small black smiley face (:) symbol in the bottom-left corner.

Identities

Nothing written on this page will be graded.

Summations

$$\begin{aligned} \sum_{i=0}^{\infty} x^i &= \frac{1}{1-x} \text{ for } |x| < 1 \\ \sum_{i=0}^{n-1} &= \sum_{i=1}^n = n \\ \sum_{i=0}^n i &= 0 + \sum_{i=1}^n i = \frac{n(n+1)}{2} \\ \sum_{i=1}^n i^2 &= \frac{n(n+1)(2n+1)}{6} = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} \\ \sum_{i=0}^n i^3 &= \left(\frac{n(n+1)}{2} \right)^2 = \frac{n^4}{4} + \frac{n^3}{2} + \frac{n^2}{4} \\ \sum_{i=0}^{n-1} x^i &= \frac{1-x^n}{1-x} \\ \sum_{i=0}^{n-1} \frac{1}{2^i} &= 2 - \frac{1}{2^{n-1}} \end{aligned}$$

Logs

$$\begin{aligned} x^{\log_x(n)} &= n \\ a^{\log_b(c)} &= c^{\log_b(a)} \\ \log_b(a) &= \frac{\log_d(a)}{\log_d(b)} \end{aligned}$$

Scratch Work

Nothing written on this page will be graded.