

Indian Institute of Science Education and Research Thiruvananthapuram
DSC 314 - Data Structures
Midsem Examination (February 20, 2025)

Time: 2 hr

Max. marks: 50

Answer all questions. Each question carries 1 marks.

1. Write the recurrence relation that can arise in relation to the time complexity of the *binary search* algorithm.

2. Consider the following postfix expression with single digit operands:

6 2 3 * / 4 2 * + 6 8 * -

The top two elements of the stack after the second * is evaluated, are:

- i) 6, 3 ii) 8, 1 iii) 8, 2 iv) 6, 2

3. Assume that the algorithms considered here sort the input sequences in ascending order. If the input is already in ascending order, which of the following is/are TRUE?

I Quicksort runs in $\Theta(n^2)$ time.

II Bubblesort runs in $\Theta(n^2)$ time

III Merge sort runs in $\Theta(n)$ time

IV Insertion sort runs in $\Theta(n)$ time

- (a) I and II only (b) I and III only (c) II and IV only (d) I and IV only

4. The height of a binary tree is the maximum number of nodes in any root-to-leaf path. The maximum number of nodes in a binary tree of height h is:

5. Let T be a perfect binary tree with n leaves. Then how many nodes in T have degree 2?

Answer all questions. Each question carries 2.5 marks.

1. Write the recurrence relation that can arise in relation to the time complexity of the following C function. Solve that recurrence relation and compute its time complexity.

```
int recursive(int n)
{
    if (n==2)
        return(1)
    else
        return(recursive(n/2)+recursive(n/2))
}
```

2. Given an array of n numbers. You are asked to pick a number which is **not** the second largest. Can you propose an $\mathcal{O}(1)$ algorithm? If so, write an algorithm; if not, state why?

3. What type of list implementation (for example, singly linked list, doubly linked list, circular linked list etc.) should be used to perform the concatenation of two lists in $\mathcal{O}(1)$ time? Justify your answer.
4. Given two sorted arrays A and B having numbers of elements m and n, respectively. Write a pseudo-code of linear time complexity ($\theta(\max(m, n))$) to merge A and B into a third array C, such that C is also sorted and having $m + n$ number of elements.

Answer all questions. Each question carries 5 marks.

1. Solve the following recurrence relations:

i $T(n) = T(n - 1) + n$
ii $T(n) = 2T(\sqrt{n}) + \log n$ and $T(1) = 1$

2. A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node (for example front, rear etc.) should p point such that both the operations enQueue and deQueue can be performed in constant time? Justify your answer.

3. We have studied "Binary Search" algorithm in class. Now consider a 'Ternary Search' algorithm in which two mid-indices are maintained; $m_1 = n/3$ and $m_2 = 2n/3$. This divides the array into three parts; left to m_1 , m_1 to m_2 , and m_2 to right. Search is conducted recursively in these three parts. Write pseudo code for the ternary search algorithm and compute its time complexity.

4. Convert the following infix expressions into postfix expressions using the operator stack.

i $A + B * C + D/E/F + P$
ii $A/B * C + D * E * F + P * Q$
iii $P * Q * R + D/E/F + A - B$
iv $P - Q/R + D * E/F$
v $A + M - L * S + (N * M) * P/Q/R * O + B$

5. Write an algorithm for level-order traversal of a binary tree. Analyze the time complexity of the algorithm.

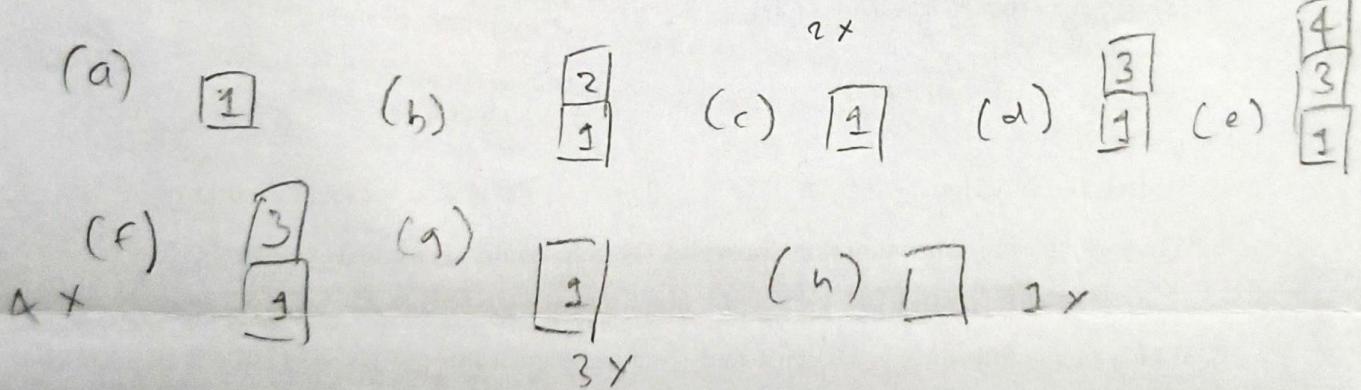
Answer all questions. Each question carries 10 marks.

1. Mr. Ram is a chemist who receives ten medicine boxes with batch numbers 35, 33, 42, 10, 14, 19, 27, 44, 26, 31 printed on them. He always arranges the boxes manually and gets frustrated every time. He thought he would have a lot of problems in the future arranging the boxes if the number of boxes of medicine was large. He wants to make this task easier. As you are a good programmer, Mr. Ram is asking for your help. Write an optimal **quick sort** algorithm to arrange the boxes in increasing order of batch numbers and show each step of the algorithm in detail using the above sequence of batch numbers. Also, write the best case and worst case time complexity of the algorithm?

- a) $S = \text{push}(S, 1);$
- b) $S = \text{push}(S, 2);$
- c) $S = \text{pop}(S);$
- d) $S = \text{push}(S, 3);$
- e) $S = \text{push}(S, 4);$
- f) $S = \text{pop}(S);$
- g) $S = \text{pop}(S);$
- h) $S = \text{pop}(S);$

Which of the following is the correct order in which elements are popped?

- a) 1, 2, 3, 4 b) 2, 1, 3, 4 c) 2, 3, 4, 1 d) 2, 4, 3, 1



$2x \rightarrow 4x \rightarrow 3x \rightarrow 1x$