CSE102 - Data Structures and Algorithms
Mid Semester Exam

Full Marks: 60 Marks **Duration: 120 mins**

1. (2+2 Marks) Write if the following statements are **True** or **False**. Explain your answer in not more than two lines.
   (a) Quicksort is slower than mergesort for any input instance.
   (b) The worst case complexity of search in any binary tree with n nodes is $O(\log(n))$.

2. (3 Marks) What is the time complexity of the following function. Justify your answer.

```
void demo(int n){
  int a, b, c;
  for (a = 1; a <= n; a++)
    for (b = 1; b <= n; b = b*2)
      for (c = 1; c <= n; c = c+4)
        printf(''Done");
}
```

3. (3 Marks) Convert the following recursive program into an iterative program.

```
int foo(int val){
  if (val <= 0)
    return foo(val+1);
  if ((val % 2) != 0)
    return foo((val*2) + 1);
  if (val > 100000)
    return val + 1;
}
```

4. (5 Marks) Implement a data-structure PQueue that stores a collection of requests and their priorities. Every request is represented using an integer value called request id. PQueue supports two operations, insert and delete. Insert adds a new request with a given priority to PQueue. Delete always removes an element with the highest priority. Out of all the elements in the PQueue with the highest priority, delete removes an element that was inserted first. The return value of delete is the request_id of the deleted element. The priority is represented using an integer value between 1 to 64, where a large value indicates higher priority. Provide pseudo-code for insert and delete operations; and the initialization of the PQueue data-structure. The time complexity of each insert and delete operation should be O(1). The space complexity should not be more than O(n). PQueue should be able to store at least 1 million requests. The following example demonstrates the working of the PQueue. Let's say the following requests (request_id, priority) are inserted in the PQueue in the given order one by one:
   (100, 1), (101, 1), (102, 2), (103, 4), (104, 4), (105, 5), (106, 2), (107, 1), (109, 2)
   - The first delete operation removes (105, 5) from PQueue and returns 105.
   - The second delete operation removes (103, 4) from PQueue and returns 103.
   - The third delete operation removes (104, 4) from PQueue and returns 104.
   - The fourth delete operation removes (102, 2) from PQueue and returns 102.

5. (3 Marks) Write a pseudo code that reverses a given linked list.

6. (5 Marks) Give an example of an algorithm of some problem in which the average-case and the worst-case time complexities are different. Mention the complexities and how to arrive at them.

7. (7 Marks) Write an algorithm to delete the minimum element from a stack of integers. The input to your algorithm is the stack, and the output is the minimum value. After the deletion, the minimum value should not be present in the stack. You can directly use push, pop, and is_empty (returns true if the stack is empty) APIs. The additional space complexity of your algorithm should be O(1). You can use recursion. The space requirement for the recursive calls will not be counted in the space complexity. You can assume that the stack is not empty and all the elements in the stack have distinct values.

8. (3 Marks) Give the pre-order and post-order traversals of the tree in Figure 1.

node
left
right

left
right
node.

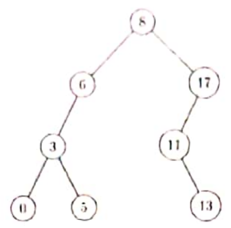

Figure 1: Tree for Traversal

9. (3 Marks) Consider the Binary Search tree in Figure 2. If the nodes 7 and 33 are deleted from the tree, what would be the final tree obtained? Show all intermediate steps.

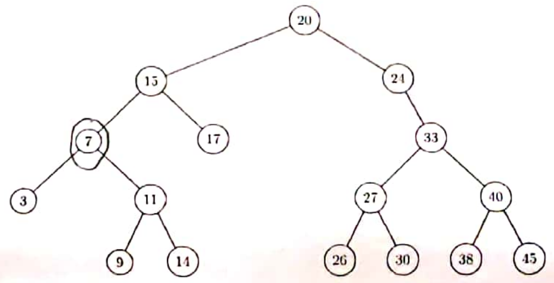

Figure 2: Tree for Deletion

10. (2+4 Marks) Write the pseudocode for the quicksort, including the partition procedure. Your algorithm should always select the leftmost element as the pivot. Write the contents of the entire array after each partition phase (not each step of the partition phase) when you run your algorithm for the following input.

4 9 8 10 2 16 1 3 11

11. (2+3 Marks) Write the pseudocode for the mergesort, including the merge procedure. Write the contents of the entire array after each merge phase (not each step of the merge phase) when you run your algorithm for the following input.

4 9 8 10 2 16 1 3 11 $\quad$ (8-4) : (4)

12. (4 Marks) Solve the following recurrence relation: $\quad$ 9-0+1 · (5)

$$T(n) = 5T(n-8) + 100; \quad T(i) = 1 \text{ for all } 0 \le i \le 8.$$

13. (4 Marks) Write an algorithm to find the median value in a linked list of integers. The additional space complexity of your algorithm should be O(1). You can assume that the linked list is not empty.

14. (5 Marks) Write an algorithm to find the minimum element in a binary tree of integers. The additional space complexity of your algorithm should be O(1). You can use recursion. The space requirement for the recursive calls will not be counted in the space complexity. You can assume that the tree is not empty.