KATHMANDU UNIVERSITY
End Semester Examination [C]
April/May, 2023

30 APR 2023

Level : B.E./B.Sc.
Year : II
Time : 2 hrs. 30 mins.

Course : COMP 202
Semester : I
F.M. : 40

SECTION "B"

[6 Q. × 4 = 24 marks]

Attempt *ANY SIX* questions.

1. Define stack ADT. Write pseudocode for major operations on stack implemented using a linked list. [1+3]

2. Which data structure would you use in the following scenario? Justify your choice. [2+2]
   i. You need to store application settings, such as language, theme etc.
   ii. You need to store information about movies and users who watched those movies.

3. Define tree traversal. Discuss different types of traversal in a binary tree. [1+ 3]

4. What is a priority queue? Using heap data structure, construct an ascending priority queue from the following data.
   2-A, 5-B, 8-X, 6-W, 1-C, 4-D, 7-E, 3-F
   Here, the first number is the priority (lower value indicates higher priority) and the letter is the data. [1+3]

5. Explain the divide-and-conquer strategy in algorithm design. Give an example of a problem which can be solved using this strategy. [2+2]

6. The best case input for a sorting algorithm is not always an array that is already sorted. Justify this statement with appropriate examples. [4]

7. Write short notes on *ANY TWO*. [2+2]
   i. Collision resolution techniques
   ii. Binary search tree
   iii. Array data structure

SECTION "C"

[2 Q. × 8 = 16 marks]

Attempt *ANY TWO* questions.

8. Define minimum spanning tree. Compare and contrast Prim's and Kruskal's algorithms. Do we always obtain the same minimum spanning tree from these two algorithms? [2 + 4 + 2]

9. Write an algorithm to convert a decimal number to binary using circular queue or doubly linked list. Compute the time complexity of your algorithm. What are the benefits of a circular queue over a simple queue? [3 + 3 + 2]

10. Assume that we represent a binary search tree by a linked data structure in which each node contains a key, and three pointers, left, right, and parent, that point to the nodes corresponding to its left child, its right child and its parent, respectively. Now consider the following algorithm, where x, a node in the BST, is an input to this algorithm, and minimum is an algorithm that returns the minimum key in the tree with the given root:

```
1: if x.right then
2:     return minimum(x.right)
3: end if
4: y = x.parent
5: while y and x == y.right do
6:     x = y
7:     y = y.parent
8: end while
9: return y
```

i. List out the properties of binary search tree. [2]

ii. Explain, with an example, what this algorithm returns. [4]

iii. Compute the time complexity of this algorithm if the time complexity of the algorithm minimum is O(h), where h is the height of the tree. [2]