Chennai Mathematical Institute

Advanced Programming

Final Examination, Spring Semester, 2021–2022

Date : 20 May, 2022 (9:30am-12:30pm) Marks : 60
Duration : Three hours Weightage : 30%

1. Assume we have a Python class `List` in which each node has the basic structure indicated by the definition to the right. The value is never `None` except for an empty list. The last element in the list has `next` equal to `None`.

```
class List:
  def __init__(self):
    self.value = None
    self.next = None
    return
```

Add a Python function to the class that takes as argument another `List` and merges the second list with the original one assuming both the lists are sorted in *non-increasing* order. In other words the call `lst1.merge(lst2)` should update `lst1` with the merge of the two sorted lists `lst1,lst2`. You should stick as closely to Python syntax as possible in attempting this question.

(8 marks)

2. Let `BST` be a binary search tree class with fields `value`, `left` and `right`. Two binary search trees are said to be isomorphic if they have the same structure and the same `value` at each position.

Write a Python method `isIsomorphic(self,T)` in the class `BST` to check if its two arguments are isomorphic. We assume that the `value` is never `None` unless the tree is empty. What is the running time of your algorithm in terms of the number of values in the two trees?

(8 marks)

3. Given two sorted lists of lengths m, n respectively, show how to find the k-th smallest element in their union in time $O(\log(mn))$. You can assume that every element in their union is distinct.

(10 marks)

4. Let $\mathcal{H}$ be a set of functions from a finite set A to a finite set B. We say that $\mathcal{H}$ is divine if it satisfies:

$$Pr_{h \in_R \mathcal{H}}[h(a_1) = b_1 \wedge h(a_2) = b_2] = \frac{1}{|B|^2},$$

for every a_1, a_2, b_1, b_2 where $a_1, a_2 \in A$ are distinct and $b_1, b_2 \in B$.

We say that $\mathcal{H}$ is universal if it satisfies:

$$Pr_{h \in_R \mathcal{H}}[h(a_1) = h(a_2)] \leq \frac{1}{|B|}$$

Let $h_0 : [n] \to [2n]$ be defined by $h_0(i) = i$. Let $\mathcal{H}_0 = \{h_0\}$.

Answer the following with substantiation:

(a) Is $\mathcal{H}_0$ divine?

(b) Is $\mathcal{H}_0$ universal?

(c) Is every divine $\mathcal{H}$ universal?

(6 marks)

5. A central vertex s in a directed graph $G = (V, E)$ is a vertex such that all vertices of G are reachable from s through directed paths. Given a vertex v show how to determine if it is central in time $O(|V| + |E|)$. Show how to determine if G has a central vertex in time $O(|V| + |E|)$. You only need to provide a description of your algorithm – no Python code is necessary. *(10 marks)*

6. Suppose you are given a square grid with m rows and n columns. Each edge of the grid is (arbitrarily) coloured either red or blue. A path from the north-west corner $(1, 1)$ to the south-east corner (m, n) is a sequence of edges each of which is either eastward or southward. A path is valid if its consecutive edges are coloured differently. Show how you will compute the number of valid paths from $(1, 1)$ to (m, n). What is the time complexity of your algorithm? Having solved the previous parts and knowing that there is a valid path how much more time do you require to find such a path? Describe your algorithm to find such a path. *(10 marks)*

7. Assume that we have defined a class `Node` to build a linked list, and a function `insert(l,n)` that inserts a new value `n` at the head of the list `l`, as given below. The function `new(Node())` allocates a fresh object of type `Node` and returns a reference (pointer) to the location of this object.

```
class Node {
  int value;
  Node next;
  ...
}
```

```
boolean insert(Node head, int newvalue){
  Node newnode;
  newnode = new(Node());
  newnode.next = head.next;
  newnode.value = head.value;
  head.next = newnode;
  head.value = newvalue;
  return(True);
}
```

Let S and H be the memory stack and heap, respectively, before a call to `insert(..)`. Using diagrams, explain how the stack and heap are transformed with respect to the original S and H when `insert(..)` is running, and after `insert(..)` returns.

(8 marks)