23B1085

CS213/293 Data Structure and Algorithms 2024 IITB India
Midterm
Prof. Ashutosh Gupta

Duration : 2 hours Total: 50 marks

Please write pseudocode otherwise there will be no marks. Some supporting definitions are given below.

Section A (26 marks)

1. (6 marks) Mark the following statements as True / False. Also, justify.
   (a) KMP is $O(n+m)$ for text of size n and pattern of size m.
   (b) For a fixed array of size 2^k for integer k, the binary search always takes the same amount of time in the case of an unsuccessful search.
   (c) Each black node of a red-black tree must have a red child.
   (d) Each node in a trie represents a unique word.
   (e) A hash function must be one-to-one.
   (f) Code `A& x = new A();` will give compilation error. Assume class A is defined and has constructor A().
2. (4 marks) Prove/disprove: if there are no red nodes in a red-black tree, then the tree is a complete tree.
3. (4 marks) Given a suffix tree for a text T, give an efficient algorithm to find the longest string that repeats in T.
   text T
   string repeats
4. (6 marks) Prove/Disprove: $O(f) + O(g) \subseteq O(f+g)$.
5. (8 marks) Let us consider a binary tree T.
   a. Give an efficient algorithm to find a farthest node from a node n in T.
   b. Give an efficient algorithm for calculating the diameter of T.
   Assume that you have access to the parent pointer. Please write pseudocode otherwise there will be no marks.

Supporting definitions

Definition 1.1 A binary tree is complete if the height of the root is h and every level $i \leq h$ has 2^i nodes.

Definition 1.2 Let f and g be functions $\mathbb{N} \to \mathbb{N}$. We say $f(n) \in O(g(n))$ if there are c and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

Definition 1.3 Let A and B be subsets of $\mathfrak{p}(\mathbb{N} \to \mathbb{N})$. $A + B = \{f + g | f \in A \wedge g \in B\}$.

Definition 1.4 The diameter of a tree is the length of the longest path between any two nodes of the tree.

Definition 1.5 A suffix tree of a text T is a trie built for suffixes of T.

Section B (24 marks)

6. (4+8 marks) A leftist heap is a heap without the structural property. Instead, it satisfies leftist property npl(left(n)) $\geq$ npl(right(n)) for each node n, where

$$\text{npl}(n) = \begin{cases} -1 & \text{if n is null} \\ \min(\ \text{npl}(\text{left}(n)),\ \text{npl}(\text{right}(n))\ ) + 1 & \text{otherwise.} \end{cases}$$

We define merge operation as follows.

Algorithm 1.1: merge(LeftistHeap a, LeftistHeap b)

```
if a == Null then return b;
if b == Null then return a;
if (value(b) < value(a)) then return merge( b, a);
right(a) := merge(right(a),b);
if npl(left(a)) < npl(right(a)) then SWAP(left(a),right(a));
return a
```

Algorithm 1.2: insert(Node a, LeftistHeap b)

```
left(a)=right(a):= Null; Return merge(a,b)
```

Algorithm 1.3: deleteMin(LeftistHeap a)

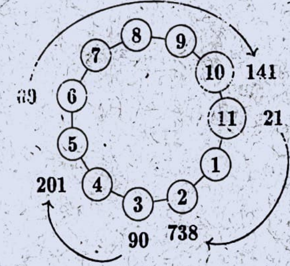
```
if a == Null then return; a
Return merge(left(a),right(a))
```

a. Prove that insert and deleteMin return leftist heaps

b. Prove that insert and deleteMin are O(log n) operations

7. (10 marks) Let us suppose we want to store n keys on m servers. The servers have IDs. We may use a hash function h to map the keys to servers. Both keys and IDs of the servers are hashed. The computed hash values are placed on a ring, like a clock. Some of the points on the ring are the servers and some are keys. A key is stored on the closest server in the clockwise direction. Each time a server is added/removed the keys are moved according to the above rule.

Example: Let $h(x) = 1 + x\%11$. We want to store keys $21, 60$, and 90 on three server with IDs $141, 201$, and 738. The hash values of the keys are $11, 6, 3$ and the hash values of the servers are $10, 4, 2$. The keys $21, 60$, and 90 will be stored in servers 738, 141, and 201 respectively.



a. Give an efficient algorithm for adding/removing keys on the servers.

b. Give an efficient algorithm for implementing add/remove of servers.

You may use the algorithms taught in the course as sub-procedures for this question. Please write pseudocode otherwise there will be no marks.