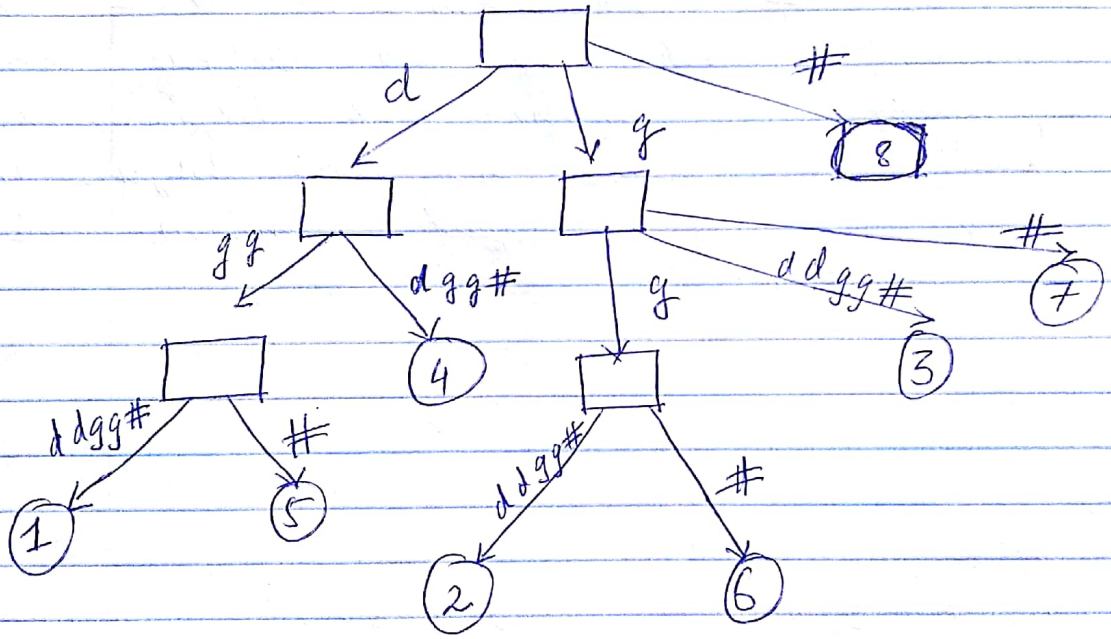
ADS Assignment

Last Name : Gupta
First Name : Siddhant
UFID : 2421-2658

Question 1

Suffix tree for d gg d d gg #



Q-1

(b) Assume 2 strings → S, Q

① set $ as the new character which is not contained in both S and Q.

② Put/set u = S $ Q #

Now the problem is modified to → finding the longest string that appears both

before and after '$'

(3) Construct a suffix tree ~~for~~ for the resulting new string 'u'

(4) Label all the branch nodes with no of elements present in that subtree

(5) Find the branch node with largest character # and has at least one element node in its subtree that represents a suffix, that begins in S, and at least one that begins in Q. This is the longest common substring in S $ Q.

(Q-2)

(a)

(1) insert (6, 6)

[0, 32]

(6, 6)

(2) Insert (9, 17)

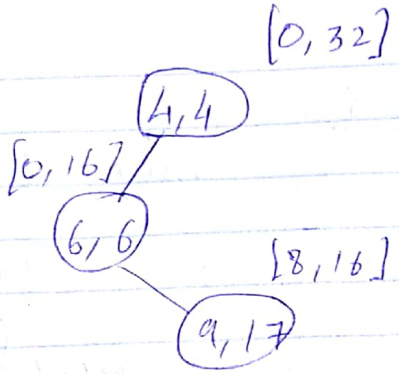
[0, 32]

(6, 6)

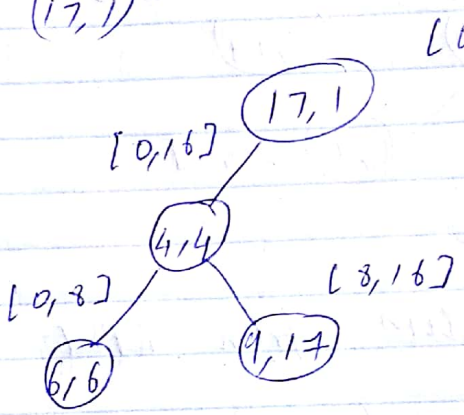
[0, 16]

(9, 17)

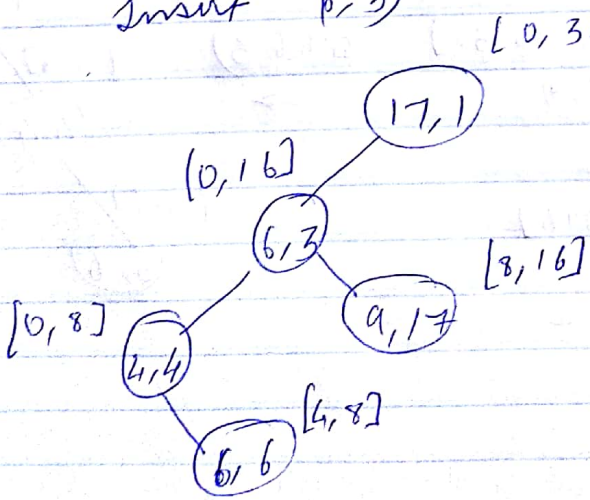
(3) Insert (4, 4)



(4) Insert (17, 1)

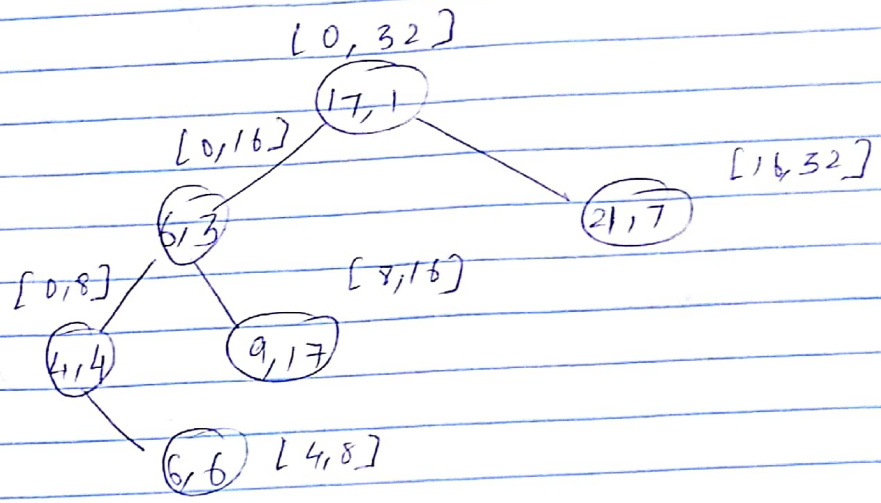


(5) Insert (6, 3)



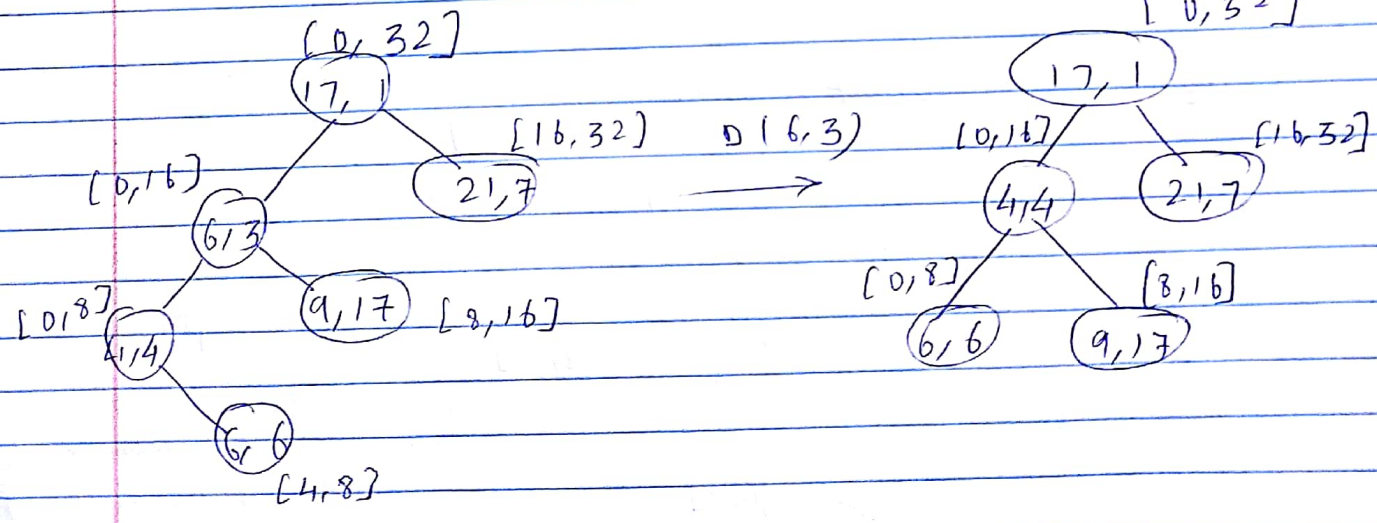
Step 6

Insert (21, 7)



Q-2 (b) Delete from RPST

Delete (6, 3)



(Q 3)

(a) Bloom Filters:

No of hash functions (h) to use =

$$(\ln 2) * (m/u)$$

$$\Rightarrow h = 0.693 * (m/u)$$

$$\Rightarrow h = 0.693 \times \frac{5000}{1000}$$

$$= 3.465$$

$$h = 3 \text{ or } 4$$

(b) Prob of false error [P(u)]

$$= e^{-u/n}\left(1 - e^{uh/m}\right)$$

For $h = 3$

$$\frac{u \times h}{m} = \frac{1000 \times 3}{5000} = 3.5$$

$$P(u) = e^{-1000/1,00,000}\left(1 - e^{-3.5}\right)^3$$

$$= 0.0909$$

For $h = 4$

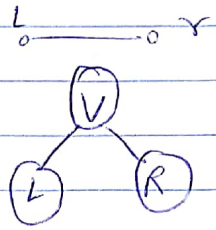
$$\frac{u \times h}{m} = \frac{1000 \times 4}{5000} = 4.5$$

$$P(u) = e^{-1/1000}\left(1 - e^{-4.5}\right)^4$$

$$= 0.0910$$

(Q-4)

Query interval $I[l, r]$



3 cases

(1) If $v.key \in [l, r]$

→ All intervals in root i.e v overlaps the query interval

→ Search both L & R trees recursively to search for overlapping intervals.

(2) If $v.key < l_i$ and $l_i > v.k$

for any interval (l_i, r_i) check for all intervals in V_i if $r_i \geq l$ return it as overlapping interval.

- Search R for any additional overlapping interval

- No need to search in L tree

(3) If $r_i < v.key$ ——

check for all intervals in V if for any interval (l_i, r_i) if $l_i \leq r$ return it as overlapping interval.

- Search L tree for additional overlapping interval

- No need to search in R tree