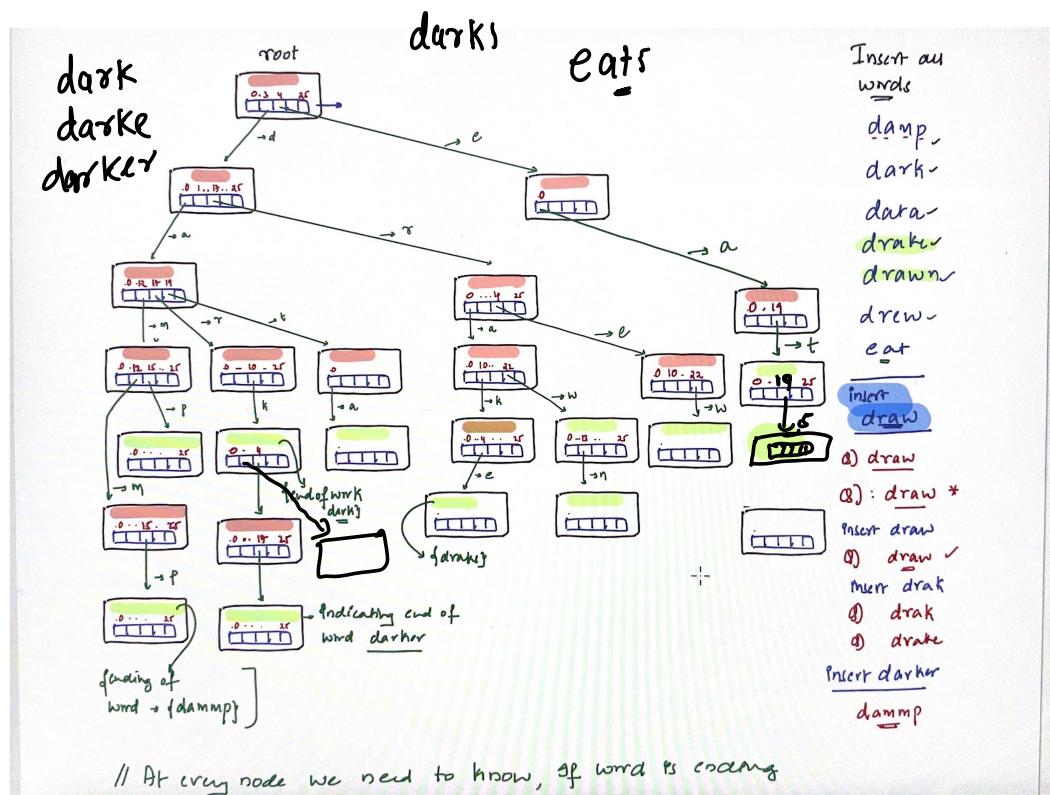


R darken

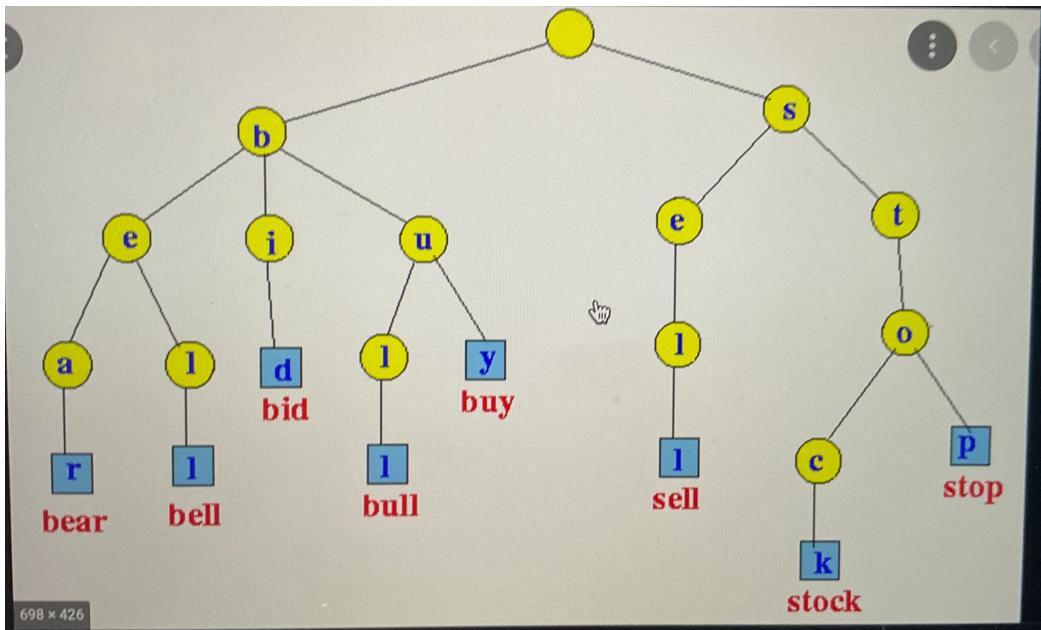


Trees 6 : Tries!

Trees will be done on Sunday



Boy tries to lift



$s_A: a a b c d$

$s_B: a a b c e$

length: λ

$\Rightarrow O(L)$

$\Theta \Rightarrow$ Given N strings & Q queries, for each query check if it is present in N strings

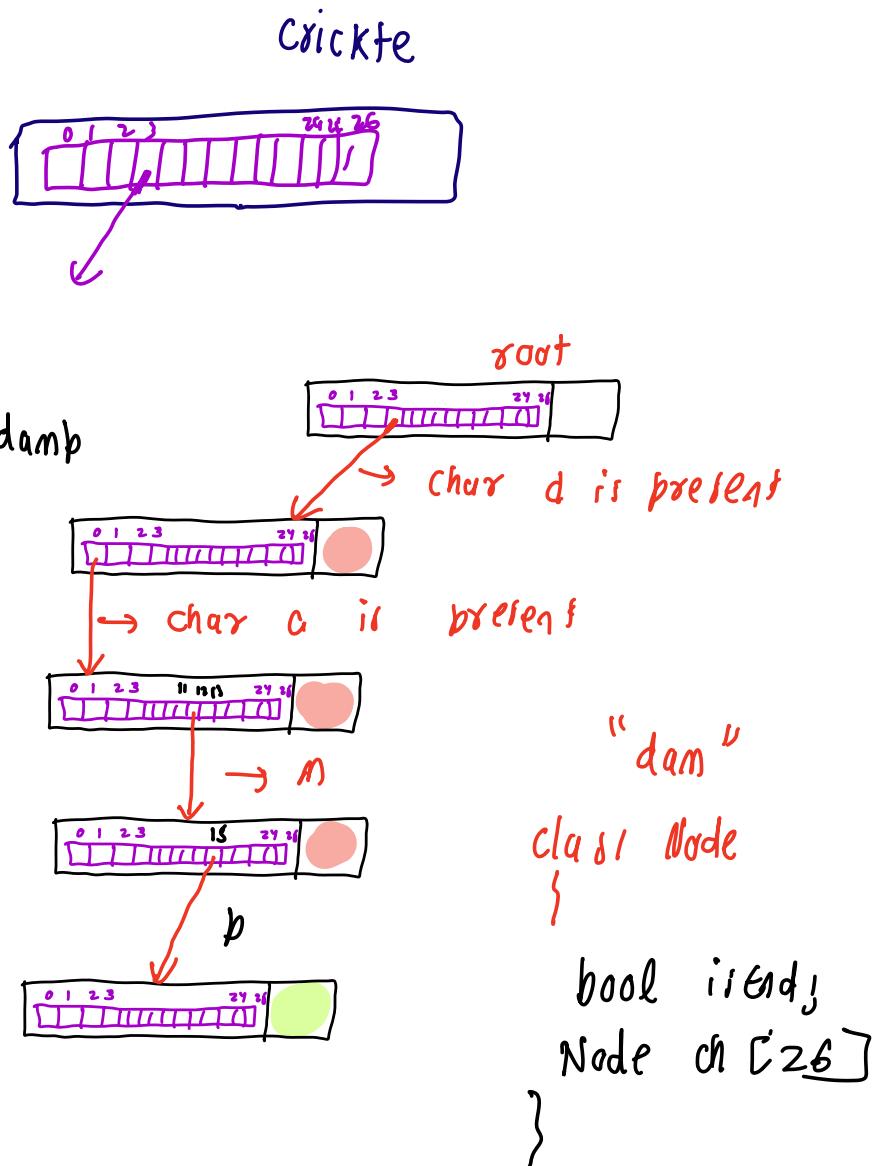
Constraints:

All chars are $['a' - 'z']$ & $1 \leq \text{length} \leq l$

words	queries	
damp	data ✓	Idea 1: For every query iterate over whole dict & compare
dark	draw ✓	
data	drew ✓	
drake	dumb ✗	$TC: O(N \cdot l \cdot Q)$
drawn	drawed ✗	
drew		Idea 2: Hashset
dried		
drunk		
draw		$TC: O(N \cdot l + Q \cdot l)$
tree		
tried		$SC: O(N \cdot l)$
trump		
tea		

Trie → hierarchical data structure
 → N-ary tree (children can be more than 2)
 → also called as prefix tree
 → Data is stored top to down

Q → Given a query, check if it belongs to correct word

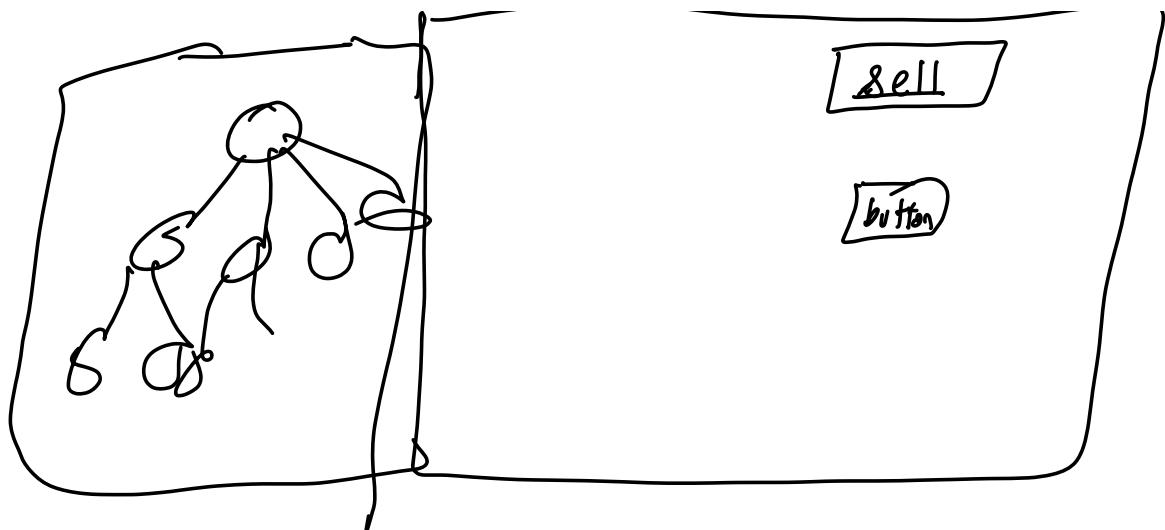


$$TC: (NL + q, l)$$

$$SC: N * L * 26$$

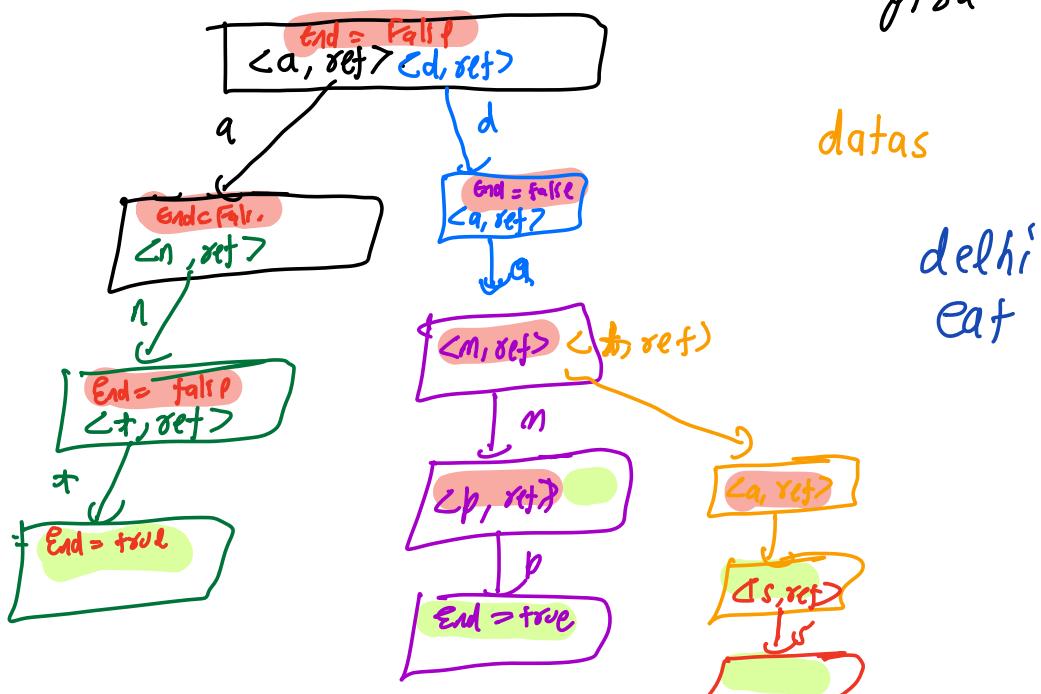
wastage

0xford dict



Has many leaf : good
girl

ant
dumb
data
dam



```
class Node
{
    bool isNode;
    HM<char, Node> hm;
    Node()
    {
    }
```

```
    isEnd = False;  
}  
  
{ damp, dam, ant, anti }
```

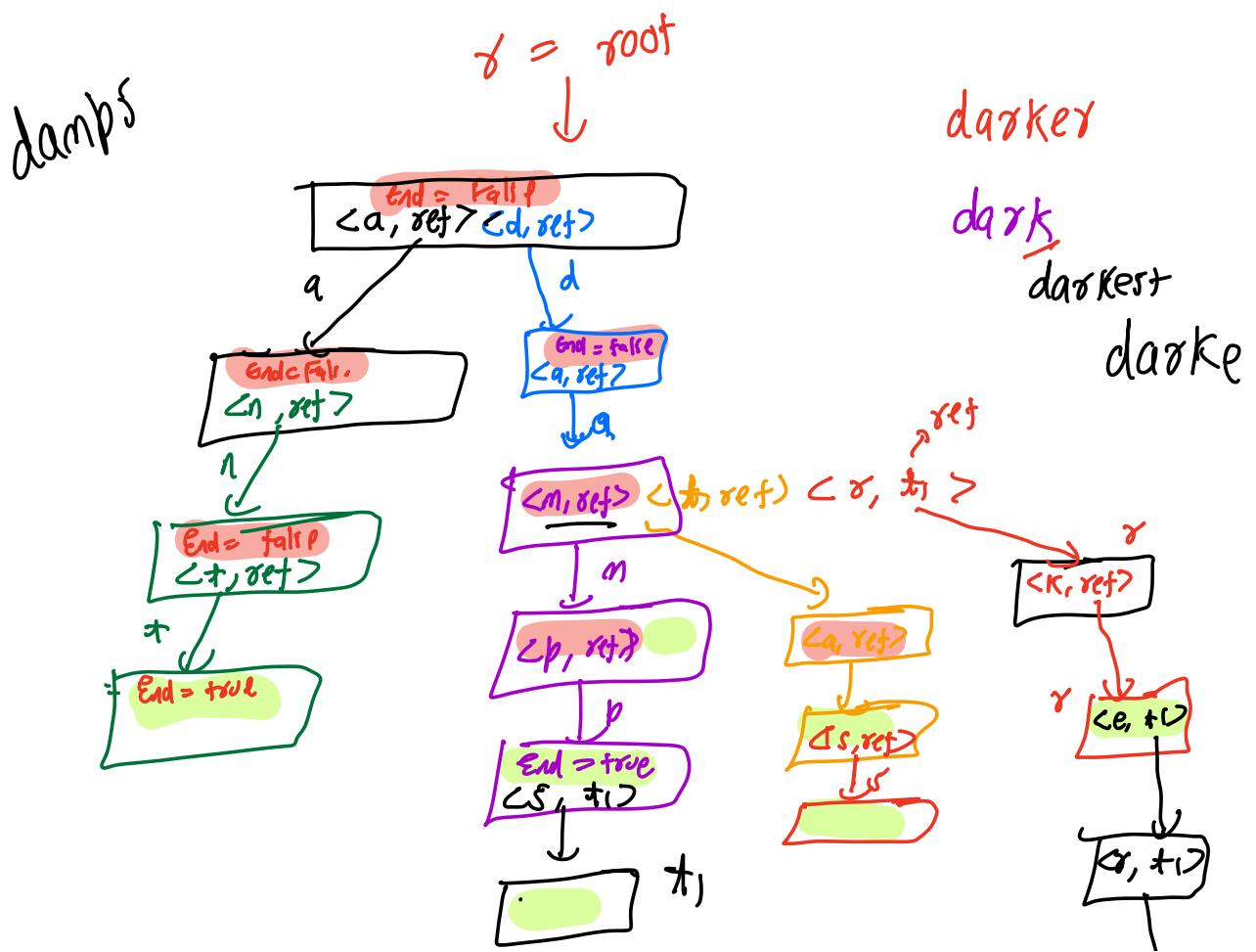
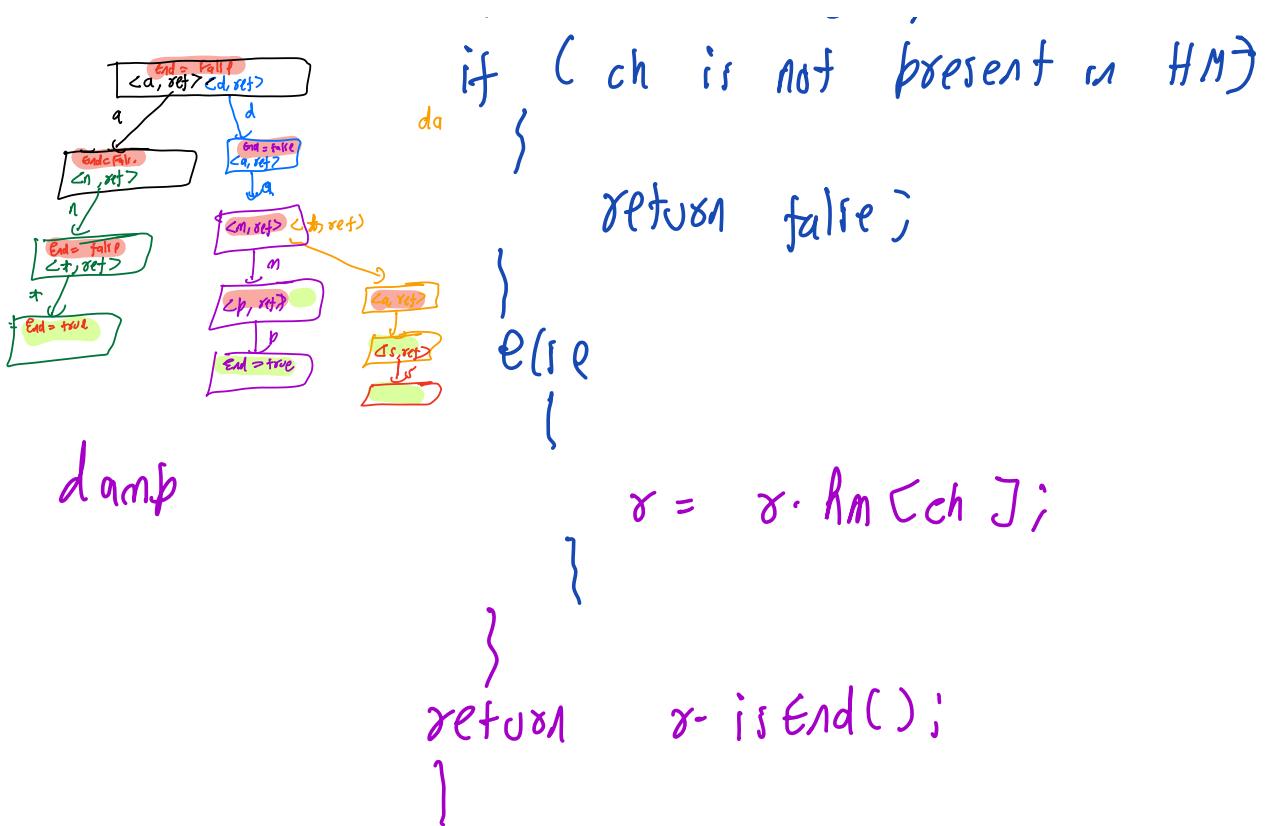
```
Read N;  
for (i=1; i <= N; i++)
```

```
    }  
    Read word;  
    add (word, root)
```

$Q \Rightarrow \{ \text{ant}, \text{dam} \}$

```
Read Q;  
for (i=1; i <= Q; i++)  
    if ( find (word, root) )  
        print "      "  
    else  
        print "      "
```

```
bool { find (String str, Node x)  
}  
int n = str.length();  
for (i=0; i < n; i++)  
    }  
    char ch = str[i];
```



```

void add (string str, Node *r)
{
    int n = str.length();
    for (int i=0; i<n; i++)
    {
        char ch = str[i];
        if (ch is not present in r-hm)
        {
            Node *t1 = new Node();
            r-hm.insert(ch, t1);
            r = r-hm[ch];
        }
        else
        {
            r = r-hm[ch];
        }
    }
    r-isEnd = true;
}

```

