Agenda

1) Search for a word

2) How many strings with given prefix

Q.1 Given N string and a word, check if given word is present in N strings.

arr = [drink , draw , drone , cat , cattle , car , mango , man]

word	ans
drink	true
drawn	false
cate	false

i) Idea1 : add all strings of arr in Hashset and then check if word is present in HashSet or not.

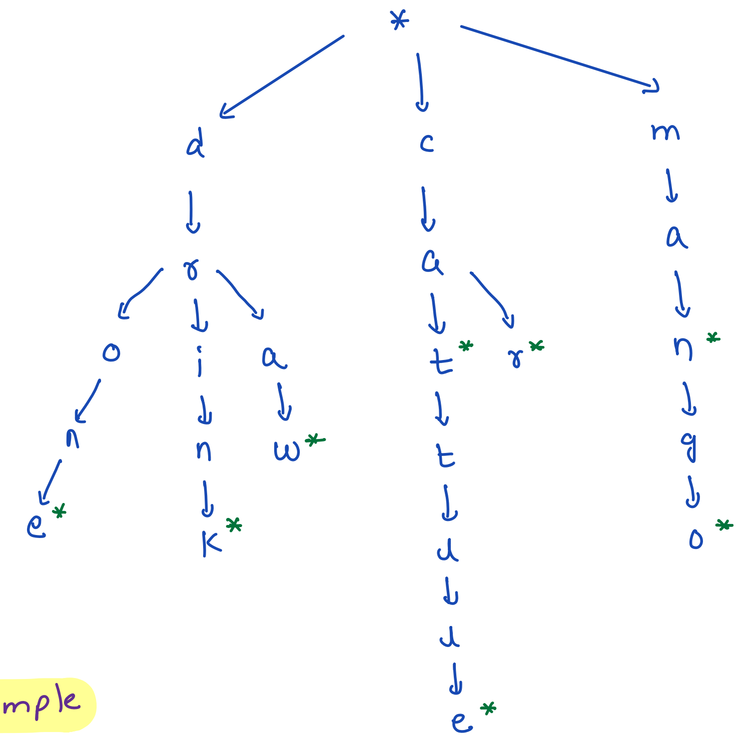
TC : O(nd)   SC : O(nd)

(avg length of string : d)

ii) Idea2 : Tries

Tries

arr = [drink , draw , drone , cat , cattle , car , mango , man]



word = car (true)

word = catt (~~true~~)

solution

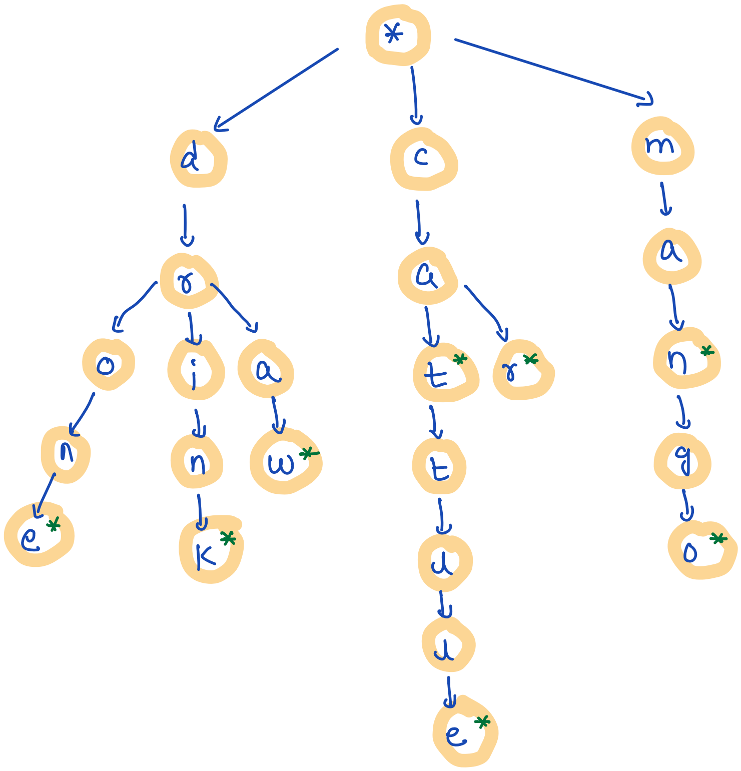
with every node keep a boolean value :

eow (end of word)

practical life example

i) phonebook

ii) spelling checker

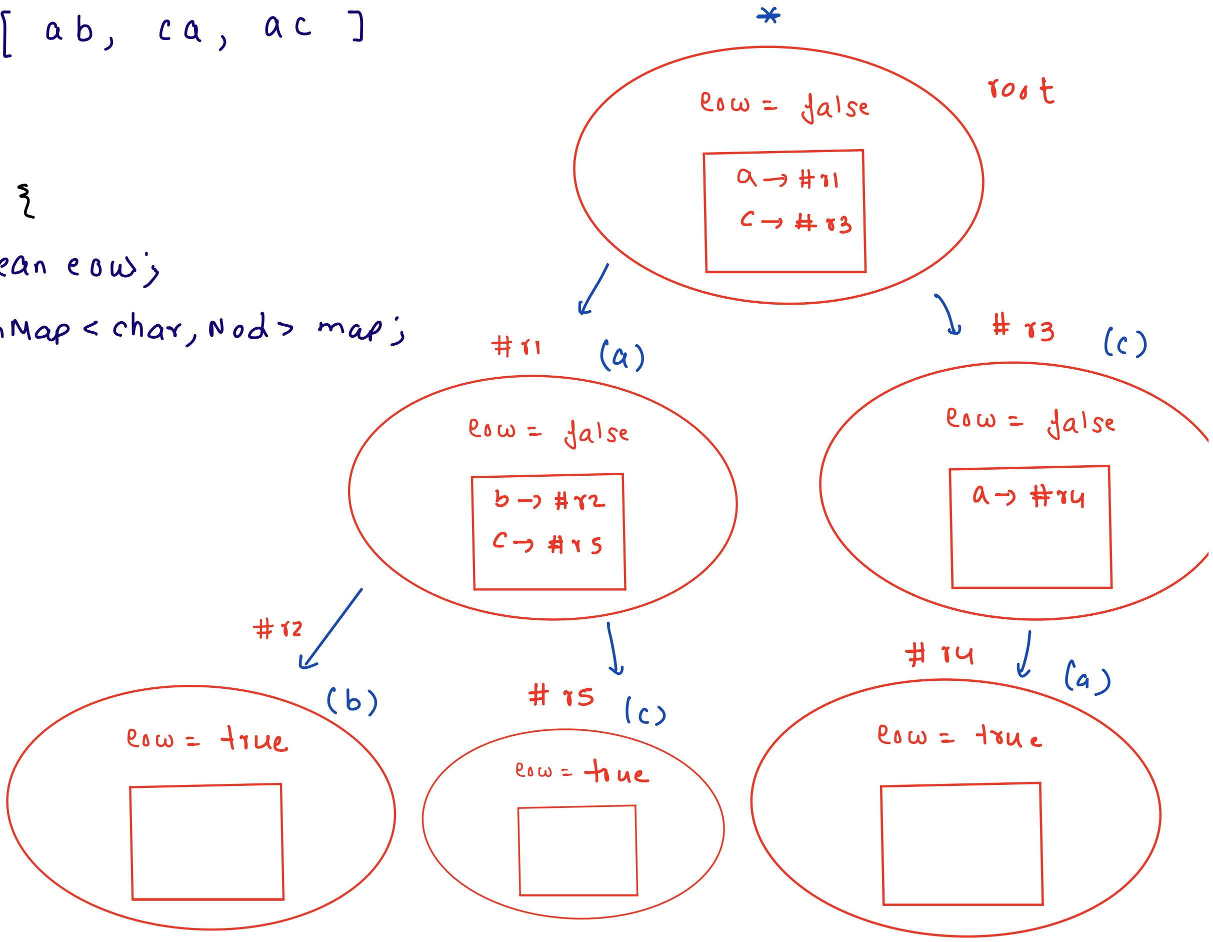


```
Node {
    boolean eow;
    HashMap<char, Nod> map;
}
```

word → cb

arr= [ab, ca, ac]

```
Node {
    boolean eow;
    HashMap<char, Nod> map;
}
```

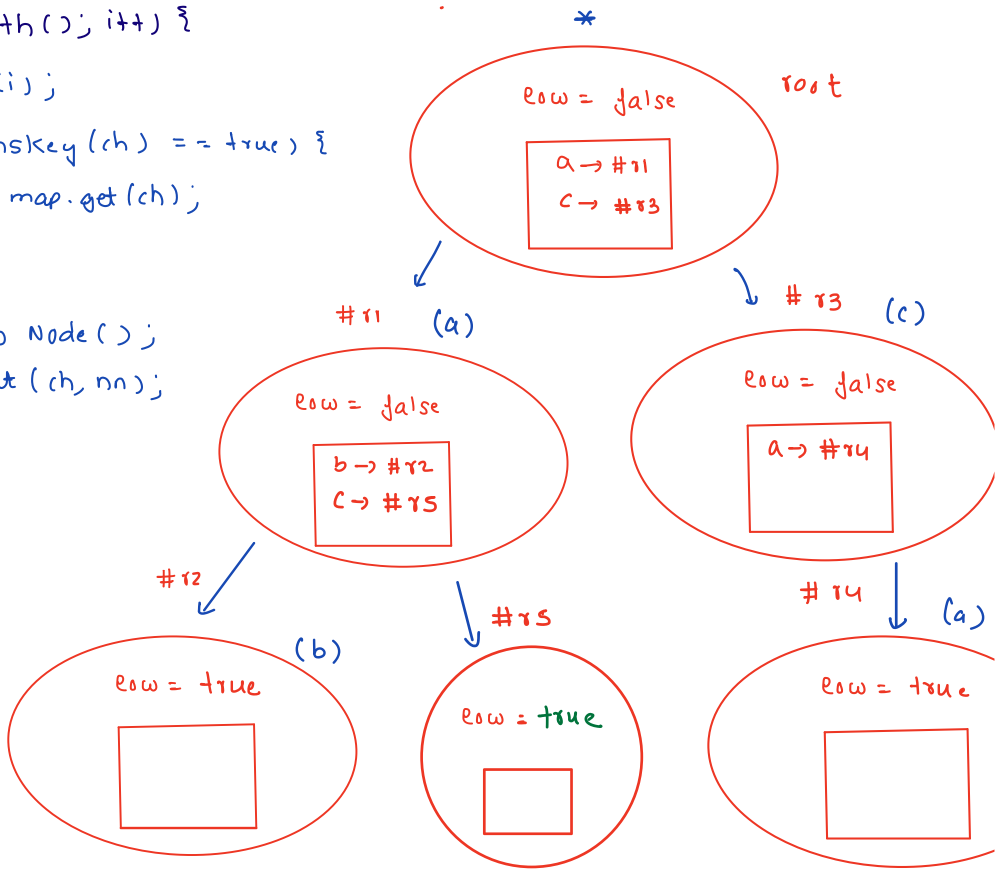


Insert a word into Trie

val = ac ↓

arr = [ab, ca, ac]

```
for(int i=0; i<val.length(); i++) {
    char ch= val.charAt(i);
    if( temp.map.containsKey(ch) == true) {
            temp= temp.map.get(ch);
    }
    else {
            Node nn= new Node();
            temp.map.put(ch, nn);
            temp=nn;
    }
}
temp.eow = true;
```



```
void  insert ( Node root, string val ) {
      Node temp=root;
      for(int i=0; i< val.length(); i++) {
            char ch= val.charAt(i);
            if( temp.map.containsKey(ch) == true) {
                  temp= temp.map.get(ch);
            }
            else {
                  Node nn= new Node();
                  temp.map.put(ch, nn);
                  temp=nn;
            }
      }
      temp.eow = true;
}
```

TC: O(nd)

SC: better than O(nd)

```
boolean  search ( Node root, string word)
      Node temp=root;
      for (int i= 0; i< word.length(); i++) {
            char ch= word.charAt(i);
            if( temp.map.containsKey(ch) ==true) {
                  temp= temp.map.get(ch);
            }
            else {
                  return false;
            }
      }
      return temp.eow;
}
```

Q.2 Given N string and a word, check how many words have prefix equals to given word.

arr = [drink , draw , drone , cat , cattle, car, mango, man, drinking]

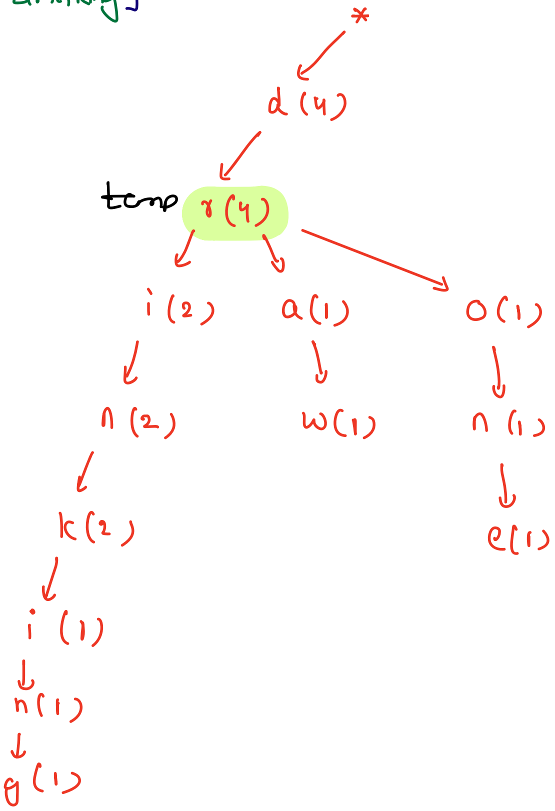
word	ans
dr	3
cat	2
mang	1
c	3

```
Node {
    int pc = 0;
    HashMap<char, Node> map;
}
```

arr = [drink , draw , drone , drinking]

how many strings startswith

word = "dr"



TC : O(nd)

SC : better than O(nd)

code snippets

https://www.interviewbit.com/snippet/85e71f9d5a689560d73e/ (search for word)

https://www.interviewbit.com/snippet/555c96e2f862ba960d36/ (how many strings starts with a given word)