Q: Given N input strings & Q queries. for each query check if given query is prefix of any given input string.

String starts at index=0 (Complete string is also a prefix)

Note :- 1 =< length of every string <= l

Input strings (N)	Queries (Q)	Ans
anaconda	anaco	✓
dress	fry	✗
eaten	roade	✓
friends	algor	✓
roades	sour	✗
anaco	dress	✓
algorithms		
sound		

Idea:

1. Insert all the N given words in Trie.
2. for every query string, iterate over the trie from root & check if the query string is prefix or not.

TC: $N \times l \times O(1) + Q \times l \times O(1)$

SC: $N \times l$

Note: Using Trie DS, searching prefix is optimal

Trie ≡ Prefix Tree

Q: Given a binary matrix mat[N][M], find the
✱ no. of distinct rows.

[2 4 3 2 4 5]

mat[7][5] $N \times M$

	0	1	2	3	4	
0	1	0	0	1	0	X
1	1	1	0	1	1	X
2	0	1	0	1	0	✓
3	1	1	0	1	1	✓
4	1	1	0	0	1	✓
5	1	0	0	1	0	✓
6	0	0	1	1	0	✓

⇒ 5

Idea 1:

For every row, compare it with all the rows below it. if freq == 0 ⇒ Count++

TC: (# of row comparisons) * (TC for each row comparison)

$$N^2 \times M$$

$$\Rightarrow O(N^2 M)$$

SC: O(1)

Idea 2 :

Convert each row into String & insert into HashSet.

$$TC: \underbrace{N * M}_{\text{Converting each row into String}} + \underbrace{N \times M}_{\text{To insert N String in HS.}}$$

$$\therefore O(NM)$$

$$SC: O(NM)$$

Idea 3 : Binary to Decimal

mat[7][5]

	2^4	2^3	2^2	2^1	2^0	
	0	1	2	3	4	
0	1	0	0	1	0	⇒ 18
1	1	1	0	1	1	⇒ 27
2	0	1	0	1	0	⇒ 10
3	1	1	0	1	1	⇒ 27
4	1	1	0	0	1	⇒ 25
5	1	0	0	1	0	⇒ 18
6	0	0	1	1	0	⇒ 6

S

(I) For every row, convert it into decimal.

(II) Insert all the decimal no's in HashSet.

$$TC: \underbrace{O(N \cdot M)}_{\text{Convert every row into decimal}} + \underbrace{O(N)}_{\text{Insert N integers in HashSet.}}$$

$$: O(NM)$$

$$SC: O(N)$$

M bits

$M < 31 \Rightarrow$ int
$M < 64 \Rightarrow$ long
$M = 100 \Rightarrow$ X

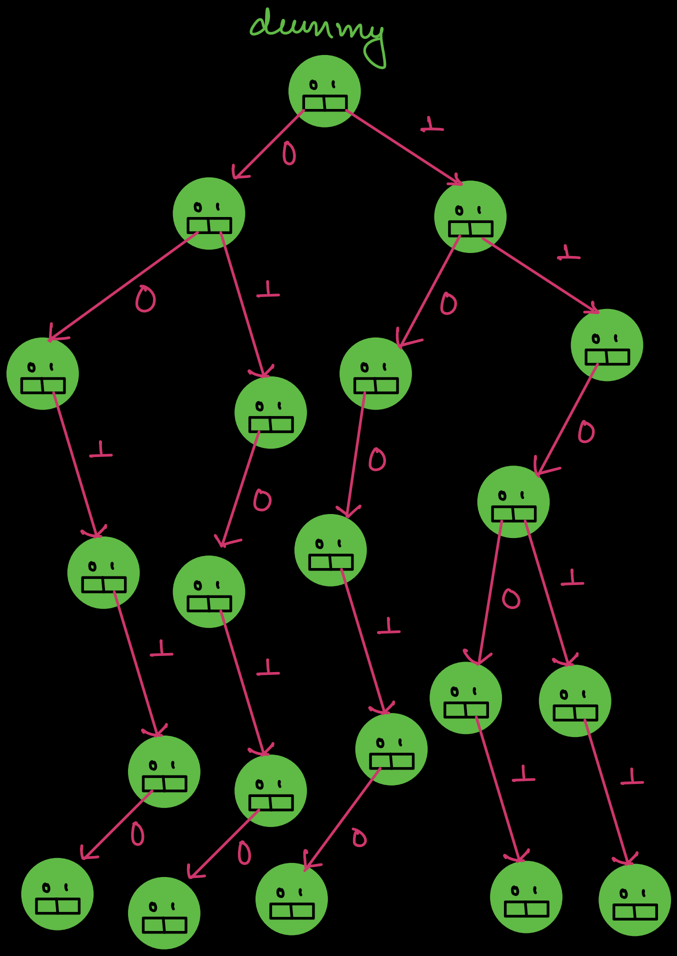
Idea 4: Trie

⇒ Insert each row in Trie

```
Class Node {
    Node left;
    Node right;
}
```

✓

```
Class Node {
    Node c[2];
    Node() {
        c[0] = Null;
        c[1] = Null;
    }
}
```



⇒ When inserting a row in Trie, if we are not creating a single new node, it means entire row is already present in the trie.

```
Node root = new Node();
```

```
int uniqueRows( int mat[][] , N, M){
      count =0
      for(i=0; i< N; i++){
            if (insert (root, mat[i], M) ){
                  count ++
            }
      }
      return count;
}
```

will return true if even a single node was created while inserting the row in Trie.

```
bool insert (root, arr[] , M){
      bool flag = false;
      for( i=0; i< M; i++){
            // Insert arr[i]
            e = arr[i];
            if (root . c[e] == NULL){
                  // Create a new node
                  root. c[e] = new Node();
                  flag = true
                  root = root. c[e];
            }
            else{
                  root = root. c[e];
            }
      }
      return flag;
}
```

TC : O(NM)

SC : O(NM) { less than N×M }

Q. Given an Array of N elements, find the max XOR value of any pair.

A[i] ^ A[j] ⇒ Max

i != j

A :	0	1	2	3
	4	3	2	7

A[0] ^ A[1] = 4^3 = 7 ✓

A[0] ^ A[2] = 4^2 = 6

A[0] ^ A[3] = 4^7 = 3

A[1] ^ A[2] = 3^2 = 1

A[1] ^ A[3] = 3^7 = 4

A[2] ^ A[3] = 2^7 = 5