2053. Kth Distinct String in an Array

```
/*
    n: size of given array of strings
    m: size of each string of in input array
    Time complexity: O(26*n*m)=O(nm)
    Space complexity: O(26*total number of letters of all strings)
*/

class Trie{
    private:
        class TrieNode{
            public:
                TrieNode* children[26]={nullptr};
                int count=0;
        };

        TrieNode* root;
    public:
        Trie(){
            root=new TrieNode();
        }

        /*
            m: size of s
            Time complexity: O(26m)=O(m)
            space complexity: O(26m)
        */
        void insert(std::string& s){
            TrieNode* current=root;
            for(auto& c: s){
                int i=c-'a';
                TrieNode* node=current->children[i];
                if(!node) {
                    node=new TrieNode();
                    current->children[i]=node;
                }
                current=node;
            }
            current->count++;
        }
```

```
        /*
            m: size of s
            Time complexity: O(26m)=o(m)
            space complexity: O(1)
        */
        bool is_distinct(std::string& s){
            TrieNode* current=root;
            for(auto& c: s){
                int i=c-'a';
                TrieNode* node=current->children[i];
                if(!node) return false;
                current=node;
            }
            return current->count==1;
        }
};

class Solution {
public:
    string kthDistinct(vector<string>& arr, int k) {
        Trie trie=Trie();
        for(auto& s: arr) trie.insert(s);
        for(auto& s: arr){
            if(trie.is_distinct(s)) k--;
            if(k==0) return s;
        }
        return "";
    }
};
```