

$abcd a$
 00001
 $s[0] \leftarrow d^{m-1} + s[1] \leftarrow d^{m-2}$

Q. Test cases

$$\alpha(p + ABC)$$

ABC

Input: 5

A

$$O/P \rightarrow A$$

ABA

ABA

A

A

Q. • Watto and Mechanism →

* Test Case, Input, 2 3

aaaaa

~~acacac~~ acacaca

- aabaa


- ccaacc

caaac

0/0 → YES

20

No

Approach: Brute force +  For each query string:

i) Compare it with every string of same length in the database

ii) Count how many chars differ.

iii) If difference = 1, print "YES".

ii) If no such string found \rightarrow "No"

Time $\rightarrow O(m \times n \times L)$

database strings \swarrow queries length

length of string

database strings → queries length

Optimal →

① Instead of comparing each query with all n strings → Store all database strings in a hash set.

→ For each query →

→ Try changing exactly one character.

→ Since alphabet = $\{a, b, c\}$, only 2 other options per position.

→ Check if modified string exists in set.

Time → $O(m \times n \times L) \rightarrow O(m \times L \times 3)$

⇒ Code →

```
int main() {
    int n, m;
    cin >> n >> m;
    unordered_set<string> st;
    for (int i = 0; i < n; i++) {
        string s;
        cin >> s;
        st.insert(s);
    }

    while (m--) {
        string query;
        cin >> query;
        bool found = false;
        for (int i = 0; i < query.size() && !found; i++) {
            char original = query[i];
            for (char ch = 'a'; ch <= 'c'; ch++) {
                if (ch == original) continue;
                query[i] = ch;
                if (st.count(query)) {
                    found = true;
                    break;
                }
            }
            query[i] = original;
        }
    }
}
```

Time
→ $O(m \times n \times 3)$

cout << (found) ? "YES" : "NO"