

abcda
o o o 1
 $s[0]^m \cdot d^{m-1} + s[1]^m \cdot d^{m-2}$
Data Page

437

Q. Test Cases

Input : 1 O/P : ABC
ABC

Input : 5
A O/P → A
ABA
ABA
A
A

Q. * Watto and Mechanism →

* Test Cases, Input : 2 3

aaaaa']
acaaacacaca]
- aabaa]
- ccacaccc]
caaac]

O/P : YES

NO

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".

iv) If no such string found → "NO".

Time → $O(m \times n \times l)$ → length of string
database queries
strings 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"

}