

Experiment -5

DATE: Amol Saurav

23Bcs10067

Problem: WATTO and mechanism.

Approach:

for every string:

- (i) Iterate through every stored string 't' with the same length.
- (ii) compare 's' and 't' character by character
- (iii) Count mismatches
- (iv) If exactly one mismatch \rightarrow return Yes, else No

Brute force:

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
int main() {
```

```
    int n, m;
```

```
    cin >> n >> m;
```

IMPORTANT NOTES

```
vector<string> dict(n);
```

```
for(int i=0; i<n; i++) cin >> dict[i];
```



DATE:

while ($m--$) {

String s;

$cin >> s;$

bool isOK = false;

for (int i=0; i<n && !isOK; i++) {

if ((int)dict.size() != s.size()) continue;

int diff = 0;

for (int j=0; j < s.size(); j++) {

if (s[j] != dict[i][j]) diff++;

if (diff > 1) break;

}

if (diff == 1) isOK = true;

}

cout << (isOK ? "yes" : "no");

return 0;

}

IMPORTANT NOTES

Time Complexity $\rightarrow O(n \times m \times l)$



DATE:

Optimal:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
using ull = unsigned long long;
```

```
int main() {
```

```
const ull BASE = 911382323ull;
```

```
int n, m;
```

```
cin >> n >> m;
```

```
vector<unordered_set<ull>> S(600001);
```

```
vector<ull> pw(600001);
```

```
pw[0] = 1;
```

```
for (int i = 1; i < pw.size(); i++) {
```

```
pw[i] = pw[i - 1] * BASE;
```

```
auto val = [&](char c) → ull {
```

```
return (ull)(c - 'a' + 1);
```

```
};
```

```
auto getHash = [&](const string &s) → ull {
```

IMPORTANT NOTES

```
ull h = 0;
```

```
for (int i = 0; i < s.size(); i++) {
```

```
h += val(s[i]) * pw[i];
```

```
};  
return h;
```



5500

ELEGANT QST

5500

ELEGANT QST

5500



DATE:

for (int i=0; i<n; i++) {

String s;

cin >> s;

St[S.size()].insert(getHash(s));

}
while (m--) {

String s;

cin >> s;

int L = (int) S.size();

ull h = getHash(s);

bool isOK = false;

for (int i=0; i < L && !isOK; i++) {

ull cur = val(S[i]) * pw[i];

long Char c = 'a'; c <= 'c'; c++) {

if (c == S[i]) continue;

ull nh = h - cur + val(c) * pw[i];

if (S[L].find(nh) != S[L].end()) {

IMPORTANT NOTES

isOK = true;

y } } break;

cout << (isOK? "yes": "no");



Time Complexity: $O((n+m)^{LL})$

550D

ELEGANT QST

550D

ELEGANT QST

550D

