


Practice

Compete

Jobs

Rank

Leaderboard

 bor0s

Dashboard > Algorithms > Implementation > The Grid Search

Badge Progress  
☆☆☆

Points: 1126.20 Rank: 21928

# The Grid Search

by PRASHANTB1984

Problem	Submissions	Leaderboard	Discussions	Editorial
---------	-------------	-------------	-------------	-----------

Given a 2D array of digits, try to find the occurrence of a given 2D pattern of digits. For example, consider the following 2D matrix:

```
1234567890
0987654321
1111111111
1111111111
2222222222
```

Assume we need to look for the following 2D pattern:

```
876543
111111
111111
```

If we scan through the original array, we observe that the 2D pattern begins at the second row and the third column of the larger grid (the **8** in the second row and third column of the larger grid is the top-left corner of the pattern we are searching for).

So, a 2D pattern of ***P*** digits is said to be present in a larger grid ***G***, if the latter contains a contiguous, rectangular 2D grid of digits matching with the pattern ***P***, similar to the example shown above.

## Input Format

The first line contains an integer, ***T***, which is the number of test cases. ***T*** test cases follow, each having a structure as described below:  
The first line contains two space-separated integers, ***R*** and ***C***, indicating the number of rows and columns in the grid ***G***, respectively.  
This is followed by ***R*** lines, each with a string of ***C*** digits, which represent the grid ***G***.  
The following line contains two space-separated integers, ***r*** and ***c***, indicating the number of rows and columns in the pattern grid ***P***.  
This is followed by ***r*** lines, each with a string of ***c*** digits, which represent the pattern ***P***.

## Constraints

$1 \leq T \leq 5$   
 $1 \leq R, r, C, c \leq 1000$   
 $1 \leq r \leq R$   
 $1 \leq c \leq C$

## Output Format

Display 'YES' or 'NO', depending on whether (or not) you find that the larger grid ***G*** contains the rectangular pattern ***P***. The evaluation will be case sensitive.

## Sample Input

```
2
10 10
7283455864
6731158619
8988242643
3830589324
2229505813
5633845374
6473530293
7053106601
0834282956
4607924137
3 4
9505
3845
3530
15 15
400453592126560
114213133098692
474386082879648
522356951189169
887109450487496
252802633388782
502771484966748
075975207693780
511799789562806
```

```
404007454272504
549043809916080
962410809534811
445893523733475
768705303214174
650629270887160
2 2
99
99
```

#### Sample Output

```
YES
NO
```

#### Explanation

The first test in the input file is:

```
10 10
7283455864
6731158619
8988242643
3830589324
2229505813
5633845374
6473530293
7053106601
0834282956
4607924137
3 4
9505
3845
3530
```

As one may see, the given 2D grid is indeed present in the larger grid, as marked in bold below.

```
7283455864
6731158619
8988242643
3830589324
2229505813
5633845374
6473530293
7053106601
0834282956
4607924137
```

The second test in the input file is:

```
15 15
400453592126560
114213133098692
474386082879648
522356951189169
887109450487496
252802633388782
502771484966748
075975207693780
511799789562806
404007454272504
549043809916080
962410809534811
445893523733475
768705303214174
650629270887160
2 2
99
99
```

The search pattern is:

```
99
99
```

This cannot be found in the larger grid.

[f](#) [t](#) [in](#)

Submissions: [28042](#)



Max Score: 30



Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆


[More](#)

Current Buffer (saved locally, editable)  

C++  

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 bool matrix_has_submatrix(vector<string> &m, vector<string> &sm) {
6     for (int i = 0; i < m.size(); i++) {
7         for (int j = 0; j < m[i].size(); j++) {
8             bool has = true;
9
10            for (int k = 0; k < sm.size(); k++)
11                for (int l = 0; l < sm[k].size(); l++)
12                    if (m[i+k][j+l] != sm[k][l] || ((i+k > m.size()) && j+l > m[0].size())) {
13                        has = false;
14                        goto skip;
15                    }
16
17            skip:
18            if (has) {
19                return true;
20            }
21        }
22    }
23    return false;
24 }
25
26 int main(){
27     int t;
28     cin >> t;
29     for(int a0 = 0; a0 < t; a0++){
30         int R;
31         int C;
32         cin >> R >> C;
33         vector<string> G(R);
34         for (int G_i = 0; G_i < R; G_i++) {
35             cin >> G[G_i];
36         }
37
38         int r;
39         int c;
40         cin >> r >> c;
41         vector<string> P(r);
42         for(int P_i = 0; P_i < r; P_i++){
43             cin >> P[P_i];
44         }
45
46         cout << (matrix_has_submatrix(G, P) ? "YES" : "NO") << "\n";
47     }
48     return 0;
49 }
50
```

Line: 1 Col: 1

 Upload Code as File ☐ Test against custom input

Run Code

Submit Code

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)