



Dashboard > Algorithms > Implementation > ACM ICPC Team

Badge Progress (Details)



Points: 1696 Rank: 16126

Your ACM ICPC Team submission got 25.00 points.

Share

Tweet

[Try the Next Challenge](#) | [Try a Random Challenge](#)

ACM ICPC Team

by dheeraj

Problem

Submissions

Leaderboard

Discussions

Editorial

Topics

There are a number of people who will be attending [ACM-ICPC World Finals](#). Each of them may be well versed in a number of topics. Given a list of topics known by each attendee, you must determine the maximum number of topics a 2-person team can know. Also find out how many ways a team can be formed to know that many topics. Lists will be in the form of bit strings, where each string represents an attendee and each position in that string represents a field of knowledge, 1 if its a known field or 0 if not.

For example, given three attendees' data as follows:

```
10101
11110
00010
```

These are all possible teams that can be formed:

| Members | Subjects |
|---------|-----------------|
| (1, 2) | [1, 2, 3, 4, 5] |
| (1, 3) | [1, 3, 4, 5] |
| (2, 3) | [1, 2, 3, 4] |

In this case, the first team will know all 5 subjects. They are the only team that can be created knowing that many subjects.

Input Format

The first line contains two space-separated integers n and m , where n represents the number of attendees and m represents the number of topics.

Each of the next n lines contains a binary string of length m . If the i^{th} line's j^{th} character is 1, then the i^{th} person knows the j^{th} topic.

Constraints

$$2 \leq n \leq 500$$

$$1 \leq m \leq 500$$

Output Format

On the first line, print the maximum number of topics a 2-person team can know.

On the second line, print the number of ways to form a 2-person team that knows the maximum number of topics.

Sample Input

```
4 5
10101
11100
11010
00101
```

Sample Output

5
2

Explanation

Calculating topics known for all permutations of 2 attendees we get:

(1,2) → 4
(1,3) → 5
(1,4) → 3
(2,3) → 4
(2,4) → 4
(3,4) → 5

The 2 teams (1, 3) and (3, 4) know all 5 topics which is maximal.

Easy

Submitted 35848 times
Max Score 25

Need Help?

[View Discussions](#)[View Editorial Solution](#)[View Top Submissions](#)

Rate This Challenge:



Resources

[Finding Max Min](#)[Bitwise OR](#)[Download problem statement](#)[Download sample test cases](#)[Suggest Edits](#)

Current Buffer (saved locally, editable)

C++14



```
1 #include <iostream>
2 #include <vector>
3 #include <map>
4 #include <algorithm>
5
6 int main()
7 {
8     int n; std::cin >> n; //the number of attendees
9     int m; std::cin >> m; //represents the number of topics(str length)
10
11     std::map<int,int> answer;
12     std::vector<std::string> vec(n);
13     for(auto &it: vec) std::cin >> it;
14
15     int maxTopics = std::numeric_limits<int>::min();
16
17     for(int i = 0; i < n-1; ++i)
```

```
18 {
19     for(int j = i+1; j < n; ++j)
20     {
21         std::string A = vec[i];
22         std::string B = vec[j];
23         int count = 0;
24
25         for(int s = 0; s < m; ++s)
26             if( (A[s] == '1') || (B[s] == '1'))
27                 ++count;
28
29         maxTopics = std::max(maxTopics, count);
30         answer[count]++;
31     }
32 }
33 std::cout << maxTopics << std::endl << answer[maxTopics] << std::endl;
34 return 0;
35 }
36
```

Line: 36 Col: 1

[Upload Code as File](#)☐ Test against custom input

Run Code

Submit Code

Congrats, you solved this challenge!

Challenge your friends: [f](#) [t](#) [in](#)

✓ Test Case #0

✓ Test Case #1

✓ Test Case #2

✓ Test Case #3

✓ Test Case #4

✓ Test Case #5

✓ Test Case #6

✓ Test Case #7

You've earned 25.00 points.

Next Challenge