

474. Ones and Zeroes

[Add to List](#)[Description](#) (?tab=Description)[Submission](#) (?tab=Submission)[Solutions](#) (?tab=Solutions)Total Accepted: **6036** Total Submissions: **16382** Difficulty: **Medium** Contributors: **piy9** (/piy9/)[Notes](#)

In the computer world, use restricted resource you have to generate maximum benefit is what we always want to pursue.

For now, suppose you are a dominator of **m** 0s and **n** 1s respectively. On the other hand, there is an array with strings consisting of only 0s and 1s.

Now your task is to find the maximum number of strings that you can form with given **m** 0s and **n** 1s. Each 0 and 1 can be used at most **once**.

Note:

1. The given numbers of 0s and 1s will both not exceed 100
2. The size of given string array won't exceed 600.

Example 1:

Input: Array = {"10", "0001", "111001", "1", "0"}, m = 5, n = 3
Output: 4

Explanation: This are totally 4 strings can be formed by the using of 5 0s and 3 1s, which are "10","0001","1","0"

Example 2:

Input: Array = {"10", "0", "1"}, m = 1, n = 1
Output: 2

Explanation: You could form "10", but then you'd have nothing left. Better form "0" and "1".

[Hide Company Tags](#) [Google \(/company/google/\)](#)[Hide Tags](#) [Dynamic Programming \(/tag/dynamic-programming/\)](#)Have you met this question in a real interview? [Discuss \(https://discuss.leetcode.com/category/605\)](https://discuss.leetcode.com/category/605)[Pick One \(/problems/random-one-question/\)](/problems/random-one-question/)[Editorial Solution](#)

C++



```
1 class Solution {
2 public:
3
4     int findMaxForm(vector<string>& strs, int m, int n) {
5         vector<vector<int>> dp(m+1, vector<int>(n+1));
6         int cntZero = 0, cntOne = 0;
7         for(auto str : strs) {
8             cntZero = cntOne = 0;
9             for(auto c : str) {
10                 if(c == '0') cntZero++;
11                 else if(c == '1') cntOne++;
12             }
13             for(int i = m; i >= cntZero; --i) {
14                 for(int j = n; j >= cntOne; --j) {
15                     dp[i][j] = max(dp[i][j], dp[i-cntZero][j-cntOne] + 1);
16                 }
17             }
18         }
19         return dp[m][n];
20     }
21 };
```

Custom Testcase ☐[Send Feedback \(mailto:admin@leetcode.com?subject=Feedback\)](mailto:admin@leetcode.com?subject=Feedback)

Run Code

Submit Solution

[Frequently Asked Questions \(/faq/\)](#) | [Terms of Service \(/tos/\)](#)

[Privacy](#)

Copyright © 2017 LeetCode

 Notes