More practice questions: leetcode.com, glassdoor.com, geeksforgeeks.org
Books: Elements of Programming Interviews, Cracking the Coding Interview
Have questions you want answered? Contact the instructor, or ask on Quora. You can post questions and follow the instructor and other people who write about algorithms.

Try to find optimized solutions, and provide a time and space complexity analysis with every solution for the algorithms questions.

Easy Problem

Starting from the top of a pyramid of numbers, you start at the topmost row and walk down, taking one value from each row and going one step to the right or to the left from your previous position with each new row, until you reach the bottom row. Your goal is to find the maximum sum path when traversing the pyramid in this way.

Example Input:

```
        55
       94 48
      95  30  96
     77  71  26  67
```

Output: 321

Explanation: One legal path is 55 -> 94 ->95 -> 77. Its sum is 321, and that is the maximum sum that can be obtained in this example. Something like 55->94->96->77 would not have been a legal path, because 96 is not just one step to the right of 94.

Medium Problem

In a party there are n different-flavored cakes of volume V1, V2, V3 ... Vn each. Need to divide them into K people present in the party such that

- All members of party get equal volume of cake (say V, which is the solution we are looking for)
- Each member should get a cake of single flavour only (you cannot distribute parts of different flavored cakes to a member).
- Some volume of cake will be wasted after distribution, we want to minimize the waste; or, equivalently, we are after a maximum distribution policy

You are given the volumes of the cakes V1, V2, V3 ... Vn as well as the parameter K. Output the maximum amount of cake each person can get.

Example Input: Cake sizes: [8, 7, 11], K: 7

Output: 3.5

Explanation: The maximum slice size is 3.5. 2 people can get that slice size from the cake of size 7, 2 people from the cake of size 8, and 3 people from the cake of size 11.


Hard Problem

(From Google Code Jam 2010)

You have a rectanglar grid of black and white squares. Your task is to make as many large square chess boards as possible. A chess board is a piece of the grid that is a square, with sides parallel to the sides of the grid, with cells colored in the pattern of a chess board (no two cells of the same color can share an edge).

Each time you cut out a chess board, you must choose the largest possible chess board left in the grid. If there are several such boards, pick the topmost one. If there is still a tie, pick the leftmost one. Continue cutting out chess boards until there is no area left. You may need to go as far as cutting out 1-by-1 mini chess boards.

Here is an example showing the a sample grid and the first few chess boards that will be cut out of it.
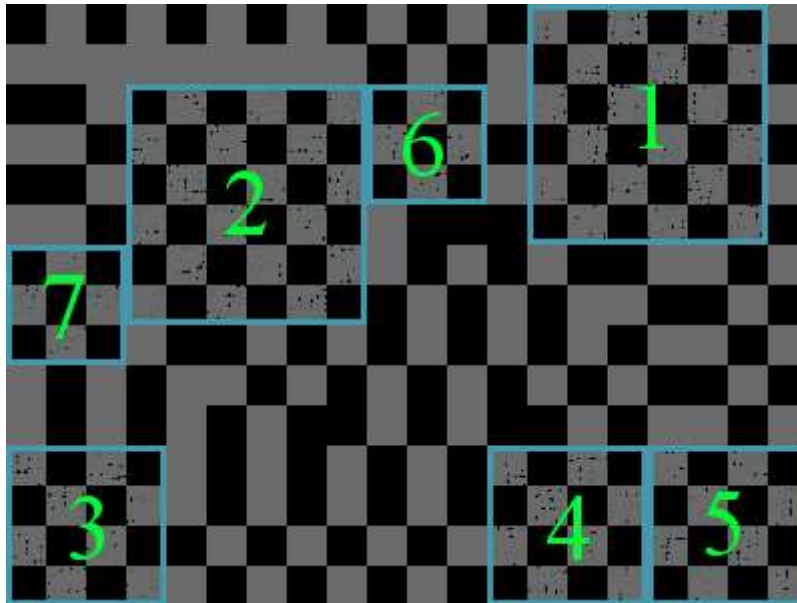
Input: You are given a matrix of booleans representing the m by n grid.

Output: For each test case, output the number of chessboards of each size you will get by following the cutting procedure described above.