



Time Remaining: 01:39:42

[ ranw0001@e.ntu.edu.sg ]

## Rainfall Test

[Back to Question List](#)

### Question 1 / 1 (Rainfall)

A group of farmers has some elevation data, and we're going to help them understand how rainfall flows over their farmland.

We'll represent the land as a two-dimensional array of altitudes and use the following model, based on the idea that water flows downhill:

If a cell's neighboring cells all have higher altitudes, we call this cell a sink; water collects in sinks. Two cells are neighbors if the rows and columns each differ by at most one. Hence an interior cell will have eight neighbors, a cell on the edge will have five neighbors, and a cell in a corner will have three neighbors.

Otherwise, water will flow to the neighboring cell with the lowest altitude. If a cell is not a sink, you may assume it has a unique lowest neighbor and that this neighbor will be lower than the cell.

Cells that drain into the same sink – directly or indirectly – are said to be part of the same basin.

Your challenge is to partition the map into basins. In particular, given a map of elevations, your code should partition the map into basins and output the sizes of the basins, in descending order.

Assume the elevation maps are square. Input will begin with a line with one integer,  $S$ , the height (and width) of the map.

The next  $S$  lines will each contain a row of the map, each with  $S$  integers – the elevations of the  $S$  cells in the row. Some farmers have small land plots such as the examples below, while some have larger plots. However, the size of the farmland exceeds  $1000 \times 1000$ .

Time Remaining: 01:39:42

Note: The input uses unix line endings ( $\backslash n$ ). If you try to view the sample inputs on a windows machine with a program that does not convert line endings (like Notepad), you will see the input appear all on a single line.

Your code should output a space-separated list of the basin sizes, in descending order. (Trailing spaces are ignored.)

While correctness and performance are the most important parts of this problem, a human will be reading your solution, so **please make an effort to submit clean, readable code**. In particular, do not write code as if you were solving a problem for a competition.

A few examples are below.

Input:

3

1 5 2

Output:

7 2

The basins, labeled with A's and B's, are:

A A B

A A B

2 4 7

A A A

Time Remaining: 01:39:42

3 6 9

Input:

Output:

There is only one basin in this case.

1

1

10

Input:

Output:

The basins, labeled with A's, B's, and C's, are:

5

10 8 7

A A A A A

1 0 2 5 8

A A A A A

2 3 4 7 9

B B B C C

3 5 7 9 9

1 2 5 5 3

3 3 5 1 0

B B C C C

B B C C C

Time Remaining: 01:39:42

Input:

4

0 3 2 3

3 2 1 4

3 4 3 3

5 5 2 0

Output:

6 5 5

The basins, labeled with A's, B's, and C's, are:

A A B B

A A B B

A B C C

A C C C

 [Download sample testcases as zip](#) ['Compile & Test' will run your code against these testcases]

Time Remaining: 01:39:42

Pick your language

C

C++

Java

PHP

Python

C#

Ruby

Perl

Haskell

Scala

Clojure

Go

Javascript

```
2 #include <stdio.h>
3 int main() {
4     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
5     return 0;
6 }
```



Keyboard Shortcuts Available

**Compile & Test**

**Submit Answer**

Time Remaining: 01:39:42

