CSI3108-01 2018. 10. 02

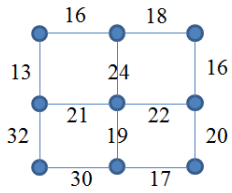
Programming HW#3

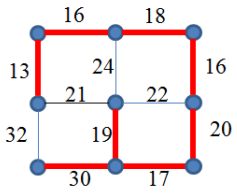
Max 30 points

Due on Oct. 8(Mon) 2018, by 5pm

Implement **Union-Find operations** discussed in the class for finding an MST (Minimum Spanning Tree) in a lattice. A lattice is an ***n*x*n*** array whose elements are positive integers. The followings are a sample 3x3 lattice, its corresponding graph, and an MST in the graph.

1	15	3
12	9	13
20	10	7





In converting a lattice into its corresponding weighted graph, we add two adjacent elements in the lattice to get the weight for the edge connecting two points associated with the elements in the lattice; for example, the top-left point is connected to its left-hand side point with 16 = 1 + 15, because first two elements in the first row of the lattice have 1 and 15, respectively.

[Input]
The first line has the number of test cases. From the next line, each test case is given in *n*+1 lines. In the first line of each test case, *n* is given. In the next *n* lines, the 2-dimensional array of a lattice is given row-by-row, one row per line. The maximum value of *n* is 10,000.

[Output]
For each case, first print the case number as '#*x*', where *x* is a case number, starting from 1 and then on the same line print the **weight of MST**, the **highest rank** during the sequence of union-find operations in obtaining the MST using the Kruskal's algorithm.

[Sample Input and Output]
Input

```
20    // total test cases
3     // n=3, Test case #1
1 15 3
12 9 13
20 10 7
…
```

Output

```
#1 149 2 //the weight of MST =149, 2 is the highest rank during the sequence of union-find operations.
…
```