2D Array - DS

Editorial by jwpierce

Given the fixed small size of the problem, brute force is fine.

Points to note:-

1. Negative values possible.
2. Maximum sum can be less than zero.
3. Range of element value is -9 to 9.
4. Numbers to be summed for each hourglass = 7.
5. Minimum possible value for sum = $7 * -9 = -63$.

So we'll initialize our maxSum to -63. From there, we just calculate the sums of all hourglasses and return the maxSum.

Here is an implementation of the function in Python:

```
def array2D(arr):

    # want to find the maximum hourglass sum
    # minimum hourglass sum = -9 * 7 = -63
    maxSum = -63

    for i in range(4):
        for j in range(4):

            # sum of top 3 elements
            top = sum(arr[i][j:j+3])

            # sum of the mid element
            mid = arr[i+1][j+1]

            # sum of bottom 3 elements
            bottom = sum(arr[i+2][j:j+3])

            hourglass = top + mid + bottom

            if hourglass > maxSum:
                maxSum = hourglass

    return maxSum
```

Tested by AllisonP

Problem Tester's code:

```
public class Solution {
    private static final int _MAX = 6; // size of matrix
    private static final int _OFFSET = 2; // hourglass width
    private static int matrix[][] = new int[_MAX][_MAX];
    private static int maxHourglass = -63; // initialize to lowest possible sum (-9 x 7)

    /** Given a starting index for an hourglass, sets maxHourglass
    *   @param i row
    *   @param j column
    **/
    private static void hourglass(int i, int j){
        int tmp = 0; // current hourglass sum

        // sum top 3 and bottom 3 elements
        for(int k = j; k <= j + _OFFSET; k++){
            tmp += matrix[i][k];
            tmp += matrix[i + _OFFSET][k];
        }

        // sum middle element
        tmp += matrix[i + 1][j + 1];
```

STATISTICS

Difficulty: Easy
Time Complexity:
Publish Date: Oct 19 2015
This is a Practice Challenge

NEED HELP?

View discussions
View top submissions