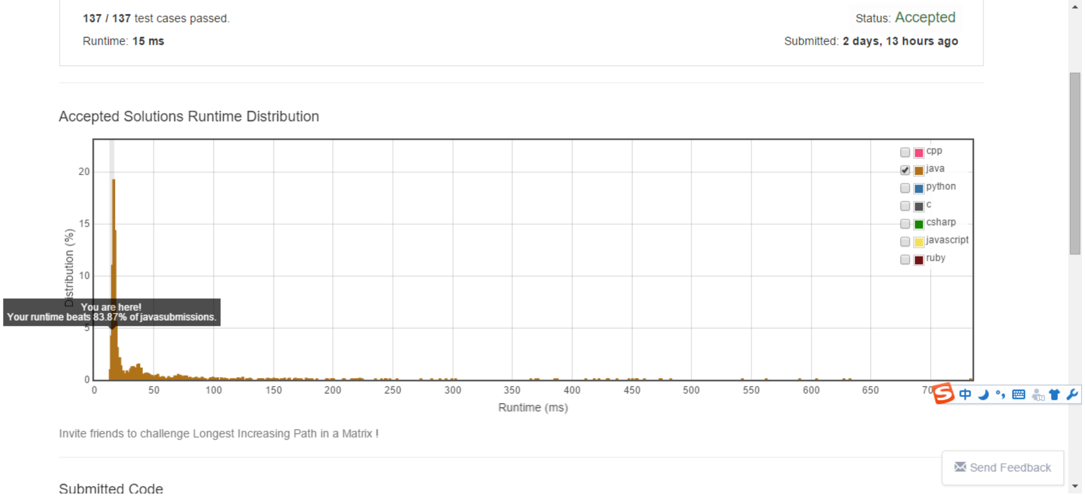
PA10

LeetCode ID：ChenJiayang



137 / 137 test cases passed.

Status: Accepted

Runtime: 15 ms

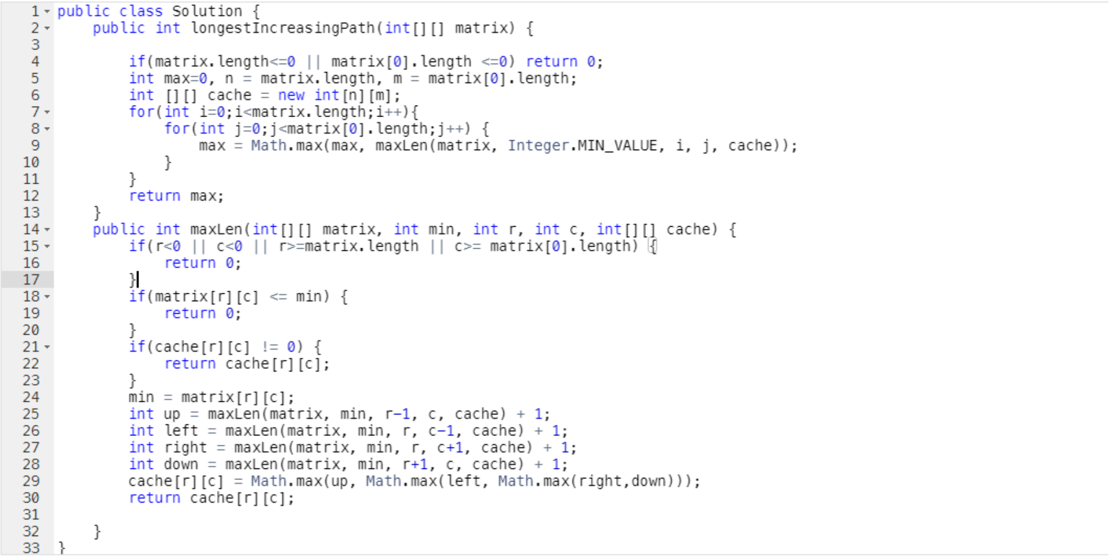
Submitted: 2 days, 13 hours ago

You are here!
Your runtime beats 83.87% of javasubmissions.

Invite friends to challenge Longest Increasing Path in a Matrix !

Submitted Code

Code：Java



```java
public class Solution {
    public int longestIncreasingPath(int[][] matrix) {

        if(matrix.length<=0 || matrix[0].length <=0) return 0;
        int max=0, n = matrix.length, m = matrix[0].length;
        int [][] cache = new int[n][m];
        for(int i=0;i<matrix.length;i++){
            for(int j=0;j<matrix[0].length;j++) {
                max = Math.max(max, maxLen(matrix, Integer.MIN_VALUE, i, j, cache));
            }
        }
        return max;
    }
    public int maxLen(int[][] matrix, int min, int r, int c, int[][] cache) {
        if(r<0 || c<0 || r>=matrix.length || c>= matrix[0].length) {
            return 0;
        }
        if(matrix[r][c] <= min) {
            return 0;
        }
        if(cache[r][c] != 0) {
            return cache[r][c];
        }
        min = matrix[r][c];
        int up = maxLen(matrix, min, r-1, c, cache) + 1;
        int left = maxLen(matrix, min, r, c-1, cache) + 1;
        int right = maxLen(matrix, min, r, c+1, cache) + 1;
        int down = maxLen(matrix, min, r+1, c, cache) + 1;
        cache[r][c] = Math.max(up, Math.max(left, Math.max(right,down)));
        return cache[r][c];

    }
}
```