

Diagram illustrating the transformation of matrix A into matrix B :

100	200	100
200	50	200
100	200	100

→

137	141	137
141	138	141
137	141	137

1	2	3	4	5	
6	7	8	9	10	
11	12	13	14	15	
16	17	18	19	20	
21	22	23	24	25	

$$\underline{J-1 < 0} \quad \underline{J+1 > n}$$

① J-150

$$\begin{aligned} \hookrightarrow \text{temp}[m][n] &= \text{arr}[i][j+1] \\ &+ \text{arr}[i-1][j] \\ &+ \text{arr}[i-1][j+1] \\ &+ \text{arr}[i+1][j] + \\ &\text{arr}[i+1][j+1] \end{aligned}$$

$$\begin{aligned} & a_{i,j}[i][j+1] + a_{i,j}[i][j-1] \\ & + a_{i,j}[i-1][j-1] + a_{i,j}[i-1][j] + \\ & a_{i,j}[i-1][j+1] + a_{i,j}[i+1][j-1] + \\ & a_{i,j}[i+1][j] + a_{i,j}[i+1][j+1] \end{aligned}$$

LeetCode

Home / X

[Problems](#)
[Daily Planner](#)
[The Clipboard](#)
[2023 Discrete](#)
[YouTube Music](#)
[300 Days of Dev](#)
[100 Days of ML](#)
[Coding Club](#)
[ML Projects](#)
[ML Roadmap](#)
[AI and Data Sci...](#)

[All Bookmarks](#)

[Daily Question](#)

[Dynamic Layout](#)
[Premium](#)

Description
Editorial
Solutions (548)
Submissions

361. Image Smoother

Easy

1461 Solved 341 Likes 20 Comments

Image smoother is a filter of the size 3×3 that can be applied to each cell of an image resulting from the average of the cell and the eight surrounding cells (i.e., the average of the nine cells of the blue square in the below image). If one or more of the surrounding cells of a cell is not present, we do not consider it in the average (i.e., the average of the four cells of the red smoother).

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

```

class Solution {
public:
    vector<vector<int>> imageSmoother(vector<vector<int>>& M) {
        if(M.empty()) return vector<vector<int>>{};
        vector<vector<int>> result(M.size(), vector<int>(M[0].size(), 0));
        vector<pair<int,int>> dir({-1,-1}, {0,-1}, {1,-1}, {-1,0}, {0,0}, {1,0}, {-1,1}, {0,1}, {1,1});

        for(int i=0; i<M.size(); i++) {
            for(int j=0; j<M[i].size(); j++) {
                int num = 0, deno = 0;
                for(const auto& d:dir) {
                    int xi = i+d.first, yj = j+d.second;
                    if(xi<0 || xi>=M.size() || yj<0 || yj>=M[0].size()) {
                        deno++;
                        num+=M[i][j];
                    }
                }
                result[i][j]=num/deno;
            }
        }
        return result;
    }
};

```

Saved to local

Terminal

Run

Debug

