



3D Surface Area

locked



by anveshi

Problem

Submissions

Leaderboard

Discussions

Editorial



Editorial by anveshi

Problem statement: Given a **3D** figure, find its surface-area.

Approach 1: $O(n^3)$

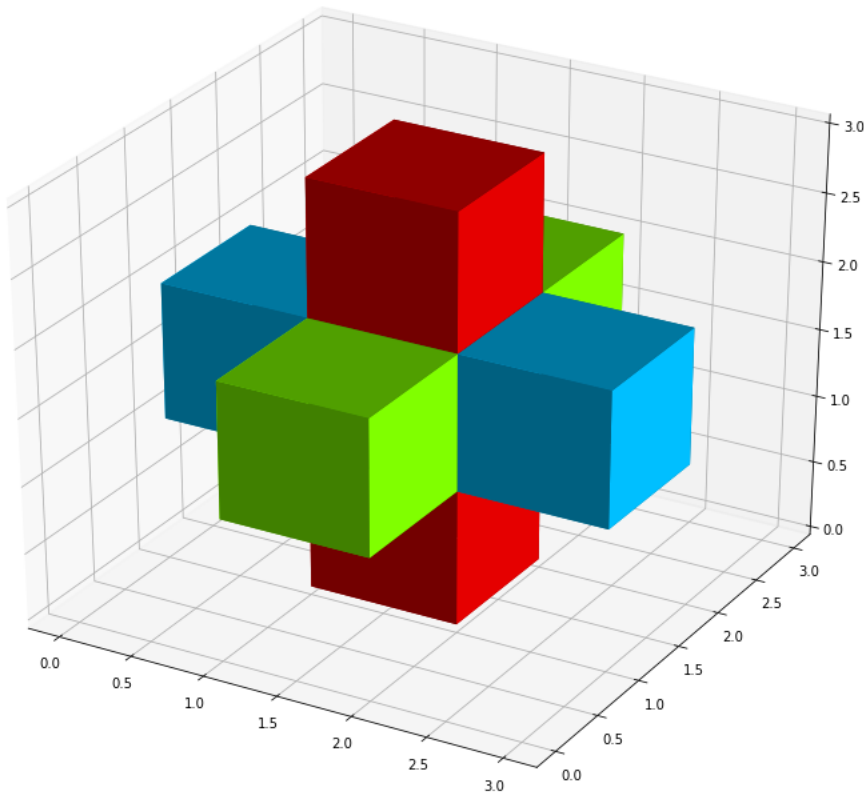
Statistics

Difficulty: Medium

Time Complexity: $O(H * W)$

Required Knowledge: Bruteforce

Publish Date: Sep 12 2017



Since $H, W \leq 100$ we can construct a **3D** Array **block** and put **1's** if there is a cube in that place else put **0**. To find the surface area, we iterate over all (i, j, k) of block such that **block[i][j][k]** is **1** for each such block, we check its **6** adjacent blocks. Each empty adjacent block will contribute **1** to the final surface area. to check **6** adjacent cells, we can have **3** array **dx, dy, dz** in direction **x, y, z** respectively such that

```
int dx[] = {0, 0, 0, 0, 1, -1};
int dy[] = {0, 0, 1, -1, 0, 0};
int dz[] = {1, -1, 0, 0, 0, 0};
```

for cell (x, y, z) its adjacent cells can be obtained as $(x + dx, y + dy, z + dz)$

Approach 2: $O(n^2)$

Instead of calculating surface area contributed by each (i, j, k) we will calculate the surface area due to every vertical column at (i, j) . To calculate the surface area contributed by a vertical column of height h_1 at (i, j) , see its 4 adjacent vertical columns at $(i - 1, j)$, $(i + 1, j)$, $(i, j + 1)$, $(i, j - 1)$. If h_2 is height at its adjacent vertical column, the surface area contributed by (i, j) is $\max(0, h_1 - h_2)$. We will add $2 \times H \times W$ to the surface area calculated by all vertical columns as the area of top and bottom is fixed ($H \times W$).



Set by anveshi

Problem Setter's code :

```
#include <stdio>
#include <cassert>
using namespace std;

int a[102][102];
bool block[102][102][102];

int dx[] = {0, 0, 0, 0, 1, -1};
int dy[] = {0, 0, 1, -1, 0, 0};
int dz[] = {1, -1, 0, 0, 0, 0};

int main() {
    int h, w;
    scanf("%d %d", &h, &w);

    for(int x = 1; x <= h; x++) {
        for(int y = 1; y <= w; y++) {
            scanf("%d", &a[x][y]);
            for(int z = 1; z <= a[x][y]; z++) {
                block[x][y][z] = 1;
            }
        }
    }

    int ans = 0;
    for(int x = 1; x <= h; x++) {
        for(int y = 1; y <= w; y++) {
            for(int z = 1; z <= a[x][y]; z++) {
                for(int i = 0; i < 6; i++) {
                    int x_ = x + dx[i];
                    int y_ = y + dy[i];
                    int z_ = z + dz[i];
                    ans += 1 - block[x_][y_][z_];
                }
            }
        }
    }

    printf("%d\n", ans);

    return 0;
}

// O(h * w)
#include <bits/stdc++.h>
using namespace std;

int a[102][102];

int dx[] = {0, 0, 1, -1};
int dy[] = {1, -1, 0, 0};

int main() {
    int h, w;

    cin >> h >> w;

    for(int x = 1; x <= h; x++)
        for(int y = 1; y <= w; y++)
            cin >> a[x][y];

    int ans = 2 * h * w; // top + bottom
```

```

    for(int x = 1; x <= h; x++) {
        for(int y = 1; y <= w; y++) {
            for(int i = 0; i < 4; i++) {
                int x_ = x + dx[i];
                int y_ = y + dy[i];
                ans += max(0, a[x][y] - a[x_][y_]);
            }
        }
    }
    cout << ans << endl;

    return 0;
}

```



Tested by [Stomach_ache](#)

Problem Tester's code :

```

#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

int a[110][110];
int dx[] = {1, -1, 0, 0};
int dy[] = {0, 0, 1, -1};

int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */

    int n, m;
    cin >> n >> m;

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            cin >> a[i][j];
        }
    }

    int ans = n * m * 2;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            for (int k = 0; k < 4; ++k) {
                int x = i + dx[k];
                int y = j + dy[k];
                if (x < 0 || x >= n || y < 0 || y >= m) ans += a[i][j];
                else if (a[x][y] < a[i][j]) ans += -a[x][y] + a[i][j];
            }
        }
    }
    cout << ans << endl;

    return 0;
}

```

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)