Median Search

You have to find the <b>Median</b> of the given <b>N\*M</b> Matrix without using any extra space. Here the matrix have all the rows sorted.<br>

Assume N\*M is odd.

Input consists of two integers N and M followed by a N\*M Matrix.

Output is Median of the given Matrix.

3 3

1 2 3

3 4 5

5 6 7

4

1<=N<=10^3 ,

1<=M<=10^3

where 1<=A[ i ][ j ]<=10^9<br>

N\*M is odd. Here the input is [1,2,3,3,4,5,5,6,7], so Median is 4.

*#include*<bits/stdc++.h>

using namespace std;

int findMedian(vector<vector<int> > &A) {

int mina = INT\_MAX, maxa = INT\_MIN;

int r = A.size();

int c = A[0].size();

int elements = (r\*c + 1) / 2;

*for*(int i = 0; i < r; i++) {

mina = min( mina, A[i][0]);

maxa = max( maxa, A[i][c-1]);

}

int mid;

*while*(mina < maxa) {

mid = mina + (maxa - mina)/2;

int curr\_count = 0;

*for*(int i = 0; i < r; i++) {

curr\_count += upper\_bound(A[i].begin(), A[i].end(), mid) - A[i].begin();

}

*if*(curr\_count >= elements) {

maxa = mid;

}

*else* {

mina = mid+1;

}

}

*return* mina;

}

int main() {

int r, c;

cin >> r >> c;

vector<vector<int> > A;

*for*(int i = 0; i < r; i++)

*for*(int j = 0; j < c; j++)

cin >> A[i][j];

cout << findMedian(A);

}

#include<bits/stdc++.h>

using namespace std;

int findMedian(vector<vector<int> > &A) {

}

int main() {

int r, c;

cin >> r >> c;

vector<vector<int> > A;

for(int i = 0; i < r; i++)

for(int j = 0; j < c; j++)

cin >> A[i][j];

cout << findMedian(A);

}