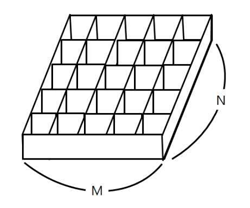
question 1 -

## 문제

철수의 토마토 농장에서는 토마토를 보관하는 큰 창고를 가지고 있다. 토마토는 아래의 그림과 같이 격자 모양 상자의 칸에 하나씩 넣어서 창고에 보관한다.



창고에 보관되는 토마토들 중에는 잘 익은 것도 있지만, 아직 익지 않은 토마토들도 있을 수 있다. 보관 후 하루가 지나면, 익은 토마토들의 인접한 곳에 있는 익지 않은 토마토들은 익은 토마토의 영향을 받아 익게 된다. 하나의 토마토의 인접한 곳은 왼쪽, 오른쪽, 앞, 뒤 네 방향에 있는 토마토를 의미한다. 대각선 방향에 있는 토마토들에게는 영향을 주지 못하며, 토마토가 혼자 저절로 익는 경우는 없다고 가정한다. 철수는 창고에 보관된 토마토들이 며칠이 지나면 다 익게 되는지, 그 최소 일수를 알고 싶어 한다.

토마토를 창고에 보관하는 격자모양의 상자들의 크기와 익은 토마토들과 익지 않은 토마토들의 정보가 주어졌을 때, 며칠이 지나면 토마토들이 모두 익는지, 그 최소 일수를 구하는 프로그램을 작성하라. 단, 상자의 일부 칸에는 토마토가 들어있지 않을 수도 있다.

## 입력

첫 줄에는 상자의 크기를 나타내는 두 정수 M,N이 주어진다. M은 상자의 가로 칸의 수, N 은 상자의 세로 칸의 수를 나타낸다. 단, 2 ≤ M,N ≤ 1,000 이다. 둘째 줄부터는 하나의 상자에 저장된 토마토들의 정보가 주어진다. 즉, 둘째 줄부터 N개의 줄에는 상자에 담긴 토마토의 정보가 주어진다. 하나의 줄에는 상자 가로줄에 들어있는 토마토의 상태가 M개의 정수로 주어진다. 정수 1은 익은 토마토, 정수 0은 익지 않은 토마토, 정수 -1은 토마토가 들어있지 않은 칸을 나타낸다.

## 출력

여러분은 토마토가 모두 익을 때까지의 최소 날짜를 출력해야 한다. 만약, 저장될 때부터 모든 토마토가 익어있는 상태이면 0을 출력해야 하고, 토마토가 모두 익지는 못하는 상황이면 -1을 출력해야 한다.

answer 1 - 시간 초과

import java.util.Scanner;

public class Main {

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

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

public static int m,n,checked=0,checknow=0;

public static int[][] tomato;

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

n=scan.nextInt();

m=scan.nextInt();

tomato=new int[m][n];

int count=-1;

Queue qu = null;

Main my=new Main();

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

tomato[i][j]=scan.nextInt();

if(tomato[i][j]==1){

if(count==-1){

qu=new Queue(i,j);

count++;

checked++;

}else{

my.inputs(i,j,qu);

checked++;

}

}

}

}//for end

while(true){

if(qu==null){

my.check(count);

break;

}

qu=my.pop(qu);

count++;

checked=checknow;

checknow=0;

/\*for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

System.out.print(tomato[i][j]);

}

System.out.println();

}

System.out.println("한바퀴 : "+count); \*/

}

/\*

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

System.out.print(tomato[i][j]);

}

System.out.println();

}

\*/

}//main end

public static void check(int count){

if(count==0){

System.out.println(-1);

return;

}

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

//System.out.print(" i,j : "+ tomato[i][j]);

if(tomato[i][j]==0){

System.out.println("-1");

return ;

}

}

//System.out.println();

}

System.out.println(count-1);

}

private Queue pop(Queue qu){

for(int i=0;i<4;i++){

if((dx[i]+qu.x>=0) && (dy[i]+qu.y>=0) && (dx[i]+qu.x<m) && (dy[i]+qu.y<n)){

if(tomato[dx[i]+qu.x][dy[i]+qu.y]==0){

tomato[dx[i]+qu.x][dy[i]+qu.y]=1;

//System.out.println("i : " + i + " qu.x+dx[i] : " + (dx[i] + qu.x) + " dy[i]+qu.y : " + (dy[i]+qu.y));

//System.out.println("dx[i] : "+ dx[i] + " dy[i] : "+ dy[i]);

inputs(dx[i]+qu.x, dy[i]+qu.y, qu);

checknow++;

}

}

}

checked--;

if(qu.qu==null){

return null;

}else{

if(checked==0){

return qu.qu;

}else{

return pop(qu.qu);

}

}

}

private void inputs(int x,int y,Queue qu){

if(qu.qu==null){

qu.qu=new Queue(x, y);

}else{

inputs(x,y,qu.qu);

}

}

public static class Queue{

int x,y;

Queue qu;

public Queue(int x,int y){

this.x=x;

this.y=y;

}

}

}

answer 2 - 검색방법을 바꿧으나 역시나 시간초과 …

import java.util.Scanner;

public class T\_7576 {

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

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

public static int m,n,checked=0,checknow=0,checkzero=0;

public static int[][] tomato;

//Queue nowqu;

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

n=scan.nextInt();

m=scan.nextInt();

tomato=new int[m][n];

int count=-1;

Queue qu = null;

T\_7576 my=new T\_7576();

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

tomato[i][j]=scan.nextInt();

if(tomato[i][j]==1){

if(count==-1){

qu=new Queue(i,j);

count++;

checked++;

}else{

my.inputs(i,j,qu);

checked++;

}

}else if(tomato[i][j]==0){

checkzero++;

}

}

}//for end

while(true){

if(qu==null){

my.check(count);

break;

}

qu=my.pop(qu);

count++;

checked=checknow;

checknow=0;

}

}//main end

public static void check(int count){

if(count==0){

System.out.println("-1");

}else if(checkzero == 0){

System.out.println(count-1);

}else{

System.out.println("-1");

}

}

private Queue pop(Queue qu){

for(int i=0;i<4;i++){

if((dx[i]+qu.x>=0) && (dy[i]+qu.y>=0) && (dx[i]+qu.x<m) && (dy[i]+qu.y<n)){

if(tomato[dx[i]+qu.x][dy[i]+qu.y]==0){

tomato[dx[i]+qu.x][dy[i]+qu.y]=1;

checkzero--;

inputs(dx[i]+qu.x, dy[i]+qu.y, qu);

checknow++;

}

}

}

checked--;

if(qu.qu==null){

return null;

}else{

if(checked==0){

return qu.qu;

}else{

return pop(qu.qu);

}

}

}

private void inputs(int x,int y,Queue qu){

if(qu.qu==null){

qu.qu=new Queue(x, y);

}else{

inputs(x,y,qu.qu);

}

}

public static class Queue{

int x,y;

Queue qu;

public Queue(int x,int y){

this.x=x;

this.y=y;

}

}

}

answer 3 - 변수를 몇개 추가하여 queue input 방식을 바꿈 good

import java.util.Scanner;

public class T\_7576 {

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

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

public static int m,n,checked=0,checknow=0,checkzero=0;

public static int[][] tomato;

private static Queue nowqu=null;

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

n=scan.nextInt();

m=scan.nextInt();

tomato=new int[m][n];

int count=-1;

Queue qu = null;

T\_7576 my=new T\_7576();

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

tomato[i][j]=scan.nextInt();

if(tomato[i][j]==1){

if(count==-1){

qu=new Queue(i,j);

nowqu=qu;

count++;

checked++;

}else{

my.inputs(i,j,nowqu);

checked++;

}

}else if(tomato[i][j]==0){

checkzero++;

}

}

}//for end

while(true){

if(qu==null){

my.check(count);

break;

}

qu=my.pop(qu);

count++;

checked=checknow;

checknow=0;

}

}//main end

public static void check(int count){

if(count==0){

System.out.println("-1");

}else if(checkzero == 0){

System.out.println(count-1);

}else{

System.out.println("-1");

}

}

private Queue pop(Queue qu){

for(int i=0;i<4;i++){

if((dx[i]+qu.x>=0) && (dy[i]+qu.y>=0) && (dx[i]+qu.x<m) && (dy[i]+qu.y<n)){

if(tomato[dx[i]+qu.x][dy[i]+qu.y]==0){

tomato[dx[i]+qu.x][dy[i]+qu.y]=1;

checkzero--;

inputs(dx[i]+qu.x, dy[i]+qu.y, nowqu);

checknow++;

}

}

}

checked--;

if(qu.qu==null){

return null;

}else{

if(checked==0){

return qu.qu;

}else{

return pop(qu.qu);

}

}

}

private void inputs(int x,int y,Queue qu){

qu.qu=new Queue(x, y);

nowqu=qu.qu;

}

public static class Queue{

int x,y;

Queue qu;

public Queue(int x,int y){

this.x=x;

this.y=y;

}

}

}