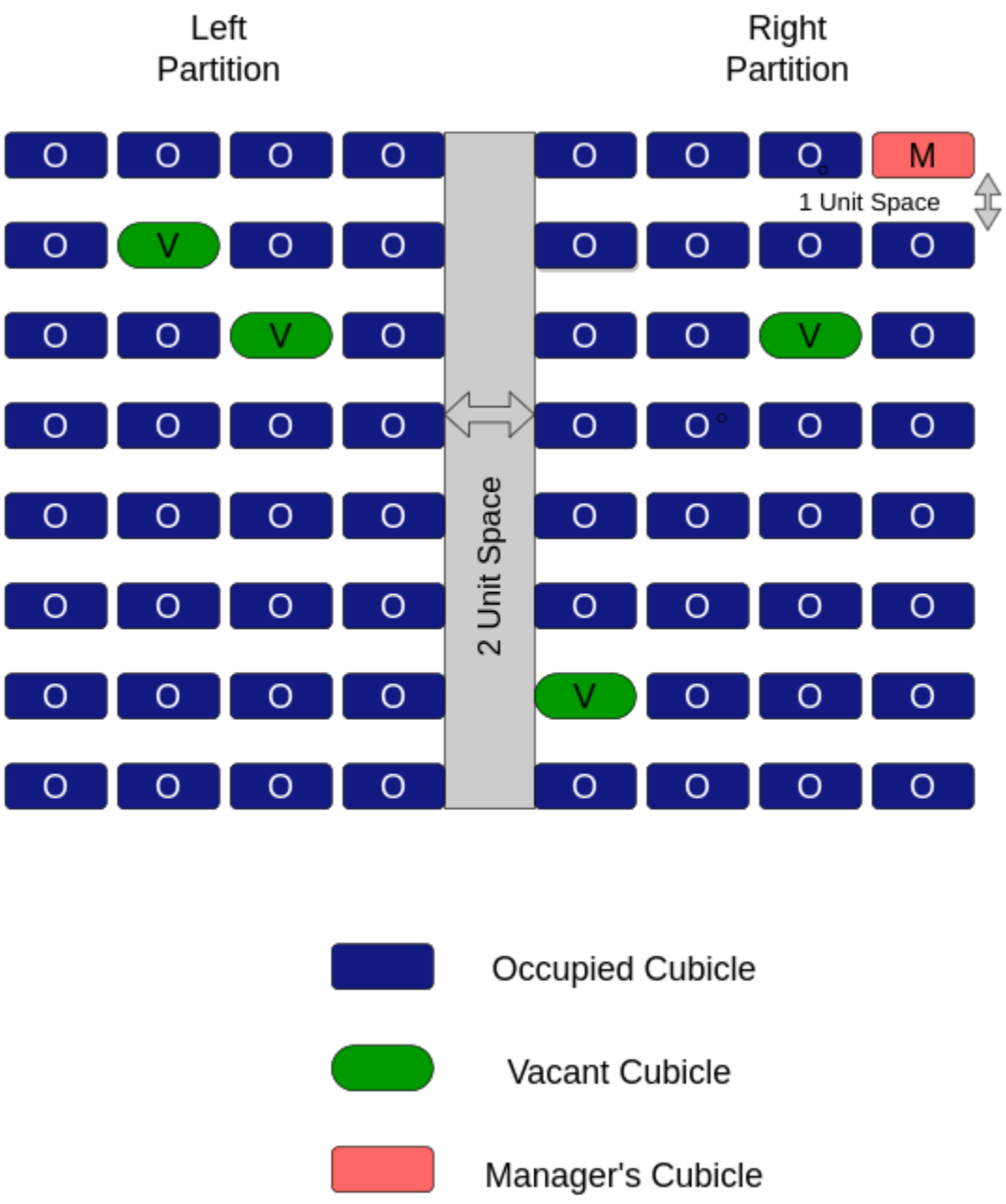
9 Aug,2021

Lazy Intern (100 Marks)

Deepanshu applied for an internship last month and went through multiple rounds. After all this process, he got selected for the internship and is very excited. He went to the office happily and received the orientation. He got to know that he will be working under the Tech Manager. On the first day at the office, he realized that he has to walk a lot as there are a lot of discussions required and meetings are set every now and then. The manager's cubicle **M** is located far from where he is seated and he has to take a long path to reach him and he is very lazy for this.

The office is divided into two partitions and is separated by space between them. There are **N** rows of cubicles with four cubicles in each row of partition. All the vacant cubicles are marked with **V** and all the occupied cubicles are marked with **O.** There is a distance of one unit between the two rows of cubicles and there is a distance of two units between the two partitions. Between the cubicles in a row of a partition, there is no space or distance.

A representation of the office is provided below with 8 rows of cubicles.



Deepanshu is very lazy and he wants to cover as minimum as possible distance to reach the Manager's row in the partition where Manager's cubicle M is located. The limitation is that the movement can be only horizontally or vertically. Deepanshu wants to know the minimum distance he has to cover to reach the Manager's row in the partition where Manager's cubicle M is located and he will figure out the vacant cubicle on his own. He is busy with the work and needs your help to determine the minimum distance he has to cover. Can you help him?

Input Format

The first line of input consists of a single integer N, representing the number of rows in the office.
Next N lines contain the representation of the view plan of cubicles in the office.

Note: Space between partitions is represented with space at the location.

Constraints

1<= N <=1000

Output Format

Print the required output in a separate line.

Sample TestCase 1

Input

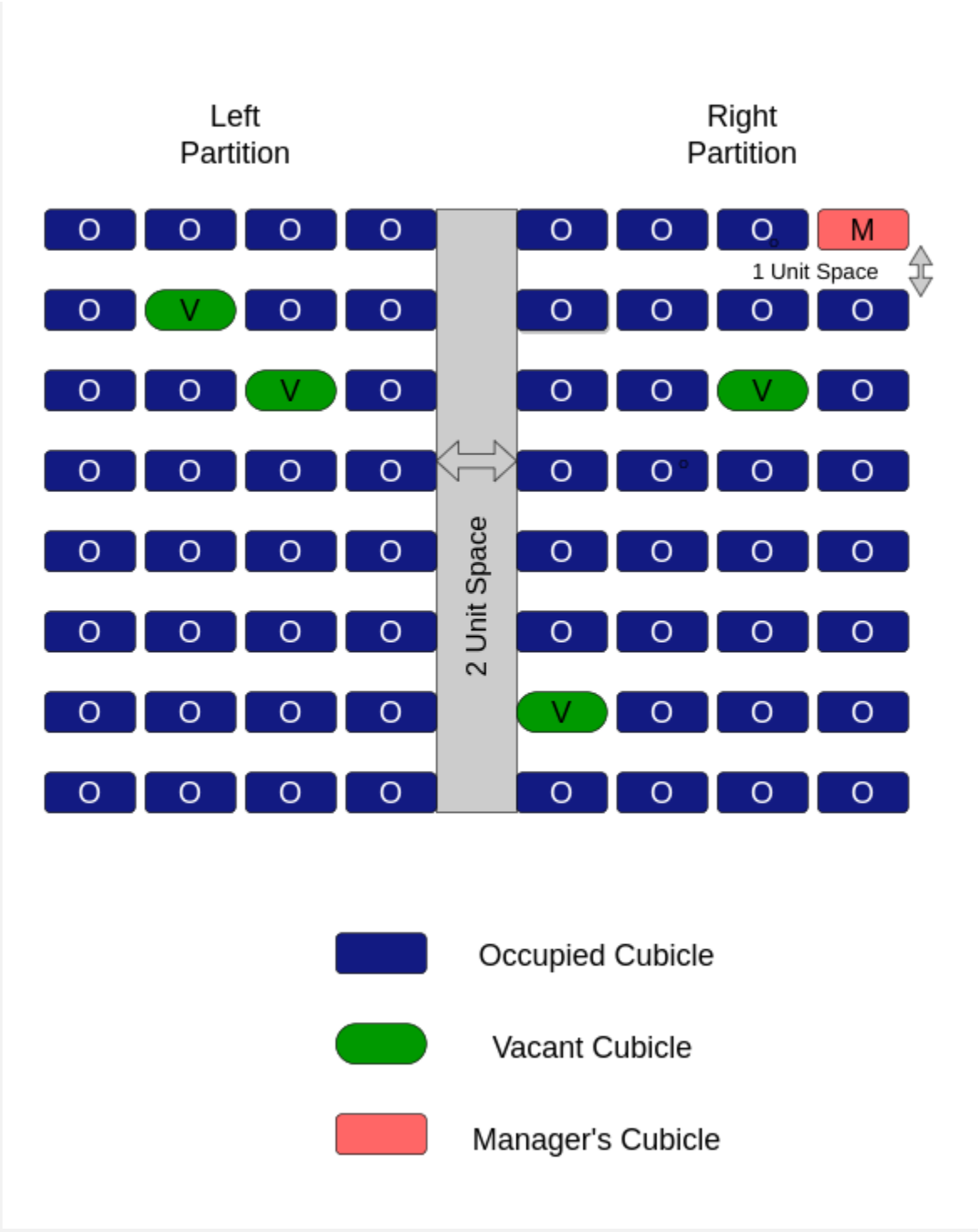
```
8
OOOO OOOM
OVOO OOOO
OOVO OOVO
OOOO OOOO
OOOO OOOO
OOOO OOOO
OOOO OOOO
OOOO VOOO
```

Output

```
2
```

Explanation

The office representation is as below:



There are 4 Vacant Cubicles in the office, located at positions [(2,2), (3,3), (3,7) and (7,5)].
The minimum distance Deepanshu has to cover is 2 units which is possible from the vacant cubicle located at position (3, 7). He can move 2 units vertically and make it to the Manager. Thus, the output is 2.

Time Limit(X):

0.50 sec(s) for each input.

Memory Limit:

512 MB

Source Limit:

100 KB

Allowed Languages:

C#

Solution:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class CandidateCode {
   static int calDist(ref int Mrow,ref int Mcol, ref int row, ref int col){
      int rowDiff=0,colDiff=0,dist;
      if(Mrow>row)rowDiff=Mrow-row;
      else if(Mrow<row)rowDiff=row-Mrow;
      if(Mcol<4 && col<4)colDiff=0;
      else if(Mcol>=4 && col>=4)colDiff=0;
      else colDiff=2;
      dist=rowDiff+colDiff;
      return dist;
   }

   static void Main(String[] args) {
                        //Write code here
      int N;
      N = Convert.ToInt32(Console.ReadLine());
      string str;
      char[,] mat = new char[N,8];
      int Mrow=0,Mcol=0,min=0,dist;
      for(int i=0;i<N;i++){
         str=Console.ReadLine();
         int j=0;
         foreach(char c in str){
            if(c==' ')continue;
            if(c=='M'){Mrow=i;Mcol=j;}
            mat[i,j]=c;
            j++;
         }
      }
      /*
      if(Mcol!=0 && mat[Mrow,(Mcol-1)] == 'V')min=0;
      else if(Mcol!=7 && mat[Mrow,(Mcol+1)] == 'V')min=0;
      else if(Mrow!=0 && mat[(Mrow-1),Mcol] == 'V')min=1;
      else if(mat[(Mrow+1),Mcol] == 'V')min=1;
      else if(Mrow!=0 && Mcol!=0 && mat[(Mrow-1),(Mcol-1)] == 'V')min=1;
      else if(Mrow!=0 && Mcol!=7 && mat[(Mrow-1),(Mcol+1)] == 'V')min=1;
      else if(Mrow!=(N-1) && Mcol!=0 && mat[(Mrow+1),(Mcol-1)] == 'V')min=1;
      else if(Mrow!=(N-1) && Mcol!=7 && mat[(Mrow-1),(Mcol+1)] == 'V')min=1;
      else*/
      for(int i=0;i<N;i++){
         for(int j=0;j<8;j++){
            if(mat[i,j]=='V'){
               if(min==0){
                  min=calDist(ref Mrow,ref Mcol,ref i,ref j);
               }
               else{
                  dist=calDist(ref Mrow,ref Mcol,ref i,ref j);
```

```
                    if(min>dist)min=dist;
                }
            }
        }
    }
    Console.WriteLine(min);
  }
}
```