**D. Flood Fill**

(Time Limit: 1 sec, Memory Limit: 256 MB)

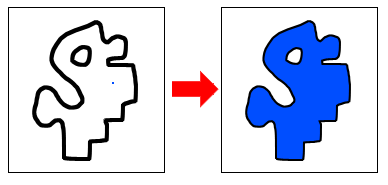
**Description**

Flood Fill是一個很經典的演算法，其目的是將一個封閉圖形裡相鄰的可填色區域給填上顏色的方式，就好比小畫家裡的油漆桶功能，可以將封閉圖形內同一顏色區塊皆填換成另一個顏色(如圖1)。

以此概念來設想，我們將放下的點視為是一個無限擴展水源，每一次水都會往上下左右的格子擴展，並在下一次重複上述步驟，對每個水都往上下左右的格子擴展，直到所有的格子都被淹滿為止，這樣子的做法又稱為BFS Flood Fill算法(如圖2)。

(有BFS就會有DFS，可以先想想DFS要怎麼實作，才會比較容易上手BFS Flood Fill的作法)

一張含有 文字, 填字遊戲 的圖片

自動產生的描述

(圖2)

(圖1)

(圖1)

現在當一個水源被放置在某一個點上，假設每經過一個時間單位，每格水就會往上下左右擴張一次，直到水碰到牆壁或是邊界為止才會停止擴張。

設問在經過時間t之後，水所覆蓋的區域會擴大到多大的面積呢？

(假設t為0的時候水剛倒下去，也就是水源還未開始向外擴展的時候)

**Sample Input/Output**

輸入的第一行有三個數字L,W和t，分別代表這個圖的長、寬和經過的時間，接下來一共會輸入L×W個由3種數字組成的二維平面圖，0代表地板、1代表牆壁、2代表水源(水源會置放1個以上)。

請嘗試計算並輸出此圖在經過時間t之後，這些水一共會佔據多少面積。( 1≤L,W≤100, 0≤t≤10000 )

|  |  |
| --- | --- |
| **Input #1** | **Output #1** |
| 11 11 5  0 0 0 0 0 0 0 0 0 0 0  0 1 0 0 0 0 1 1 1 0 0  0 0 0 0 0 0 0 0 1 0 0  0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0  0 0 0 1 0 2 0 1 0 0 0  0 0 0 1 0 0 0 0 0 0 0  0 0 0 1 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0 | 49 |

|  |  |
| --- | --- |
| **Input #2** | **Output #2** |
| 8 8 4  0 0 0 0 0 0 0 0  0 1 1 1 1 1 1 1  0 1 2 0 0 0 0 0  0 1 0 0 0 0 0 0  0 1 0 0 0 0 0 0  0 1 0 0 0 0 0 0  0 1 0 0 0 0 0 0  0 1 0 0 0 0 0 2 | 30 |