



第三題：拆地雷 (Minesweeper) [此題為互動題 Interactive]

問題敘述

蝸牛島是一座四面環海的島嶼，我們可以用 $N \times N$ 的坐標系來表示這座島嶼，其中島嶼的左上角座標為 $(1, 1)$ ，右下角的座標為 (N, N) 。且蝸牛島與海相鄰的 4 邊皆為沙灘，中間為森林，如下圖所示：

沙灘	沙灘	沙灘	沙灘
沙灘	森林	森林	沙灘
沙灘	森林	森林	沙灘
沙灘	沙灘	沙灘	沙灘

蝸牛島是遠古時期，板條大戰時的第一線戰場，至今為止，島上仍分布著各式各樣的陷阱。經過調查後，學者發現蝸牛島上，總被埋藏了 N 個地雷，且 **地雷只有可能被埋在森林底下**。而傳說中，在蝸牛島的某處藏著「尖叫雞的祕寶」。

蝸牛老師一直以來都很想要得到「尖叫雞的祕寶」，為此，蝸牛老師開發出了兩個特殊道具「地雷偵測器」以及「地雷拆除器」。蝸牛老師將他所製作出的兩樣道具交給了你，並告訴了你：

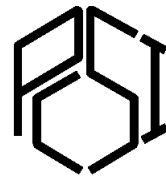
「你的期末成績將會根據你拆除的地雷數量，以及你拆除的速度所決定。」

你仔細地研究了蝸牛老師的道具後發現：

「地雷偵測器」：給定一組座標 (x, y) 後，他會告訴你以 (x, y) 為中心的 3×3 方格中（包含他自己那格）有多少顆地雷。

「地雷拆除器」：選定一組座標 (x, y) ，並且將該位置的地雷拆掉，如果該位置沒有地雷，或是地雷已經被拆除，則沒有事情會發生，你最多可以使用地雷拆除器 N 次。

為了不要在期末時看到資訊科被當掉，於是你決定自己先寫個程式來模擬拆地雷。



實作細節

C++

你需要在首行加入 `#include "Minesweeper.h"`，並完成以下函式：

```
void defuse_mines(int N);
```

- 評分程式一開始會呼叫 `defuse_mines` 並傳入 N 值。
- 傳入的 N 值滿足 $4 \leq N \leq 500$ 。
- 請在此函數內實作拆除地雷。

你的程式可以呼叫以下函式：

```
int detect(int x, int y);
```

- 代表你使用了「地雷偵測器」，偵測了座標 (x, y) 的位置，`detect` 會回傳一個整數，代表以 (x, y) 為中心的 3×3 方格中（包含他自己那格）有多少顆地雷。
- 傳入之參數 x, y 必須為 $[1, N]$ 之間的整數。
- 該函式只能被呼叫 125000000 次。

```
void defuse(int x, int y);
```

- 代表你使用了「地雷拆除器」，拆除了座標 (x, y) 的地雷，如果該位置沒有地雷，或是地雷已經被拆除，則沒有事情會發生。
- 傳入之參數 x, y 必須為 $[2, N - 1]$ 之間的整數。
- 該函式只能被呼叫 N 次。

如果不符合上述條件限制，你的程式會被判為 **Wrong Answer**；否則你的程式會依據你呼叫函數的次數以及拆除的地雷數量來評斷給分。



Python3

你需要完成以下函式：

```
def defuse_mines(N: int)
```

- 評分程式一開始會呼叫 `defuse_mines` 並傳入 N 值。
- 傳入的 N 值滿足 $4 \leq N \leq 500$ 。
- 請在此函數內實作拆除地雷。

你的程式可以呼叫以下函式：

```
def detect(x: int, y: int) -> int
```

- 代表你使用了「地雷偵測器」，偵測了座標 (x, y) 的位置，`detect` 會回傳一個整數，代表以 (x, y) 為中心的 3×3 方格中（包含他自己那格）有多少顆地雷。
- 傳入之參數 x, y 必須為 $[1, N]$ 之間的整數。
- 該函式只能被呼叫 125000000 次。

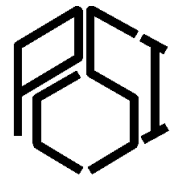
```
def defuse(x: int, y: int)
```

- 代表你使用了「地雷拆除器」，拆除了座標 (x, y) 的地雷，如果該位置沒有地雷，或是地雷已經被拆除，則沒有事情會發生。
- 傳入之參數 x, y 必須為 $[2, N - 1]$ 之間的整數。
- 該函式只能被呼叫 N 次。

請在 `defuse_mines` 函式內的第一行加入 `from __main__ import detect, defuse` 來導入函式，才能做使用。如：

```
def defuse_mines(N: int):  
    from __main__ import detect, defuse
```

如果不符合上述條件限制，你的程式會被判為 **Wrong Answer**；否則你的程式會依據你呼叫函數的次數以及拆除的地雷數量來評斷給分。



互動範例

考慮以下的測試資料： $N = 6$ ，地雷的位置如下圖（注意你其實看不到這張表）：

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
(2,1)	地雷				(2,6)
(3,1)		地雷	地雷		(3,6)
(4,1)	地雷			地雷	(4,6)
(5,1)		地雷			(5,6)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

則每一格若用偵測器偵測的話會得到的數字，如下表（注意你其實看不到這張表）：

1	1	1	0	0	0
1	2	3	2	1	0
2	3	4	3	2	1
1	3	4	4	2	1
1	2	2	2	1	1
0	1	1	1	0	0

一個不會被評分程式判斷為 **Wrong Answer** 的互動例子顯示如下：



評分程式端	參賽者端
呼叫 defuse_mines(6)	
回傳 1	呼叫 detect(1,1)
回傳 2	呼叫 detect(2,2)
	呼叫 defuse(2,3)
	呼叫 defuse(2,2)
回傳 4	呼叫 detect(4,3)
	呼叫 defuse(2,2)
回傳 1	呼叫 detect(5,2)
回傳 1	呼叫 detect(1,2)
回傳 2	呼叫 detect(3,1)
	呼叫 defuse(4,5)
回傳 1	呼叫 detect(5,3)
	呼叫 defuse(3,4)
	呼叫 defuse(3,3)

評分說明

本題共有 5 組測試題組，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	0	互動範例 ($N = 6$ ，地雷分布如互動範例的圖)。
2	4	$N = 4$ 。
3	11	$N = 5$ 。
4	17	$N \leq 25$ 。
5	68	無額外限制。

- 對於子任務 1、2、3、4：

若你在該子任務中的第 i 筆測資中拆除了 K_i 顆地雷，則令該筆測資的正確率 $P_i = \frac{K_i}{N_i}$ ，而你在該子任務的得分會用以下方式計算：

- 該子任務配分 \times 該組所有測試資料中你獲得最小的 P_i 。

- 對於子任務 5：

若你在該子任務中的第 i 筆測資中拆除了 K_i 顆地雷，並且使用了偵測器 Q_i 次，則令該筆測資的正確率 $P_i = \frac{K_i}{N_i}$ ，且你在該子任務的得分會依照以下方式計算，並取該組所有測試資料中計算出來最小者作為最後的得分：

- 若 $Q_i > 250000$ ，你將不會獲得任何分數。
- 若 $200000 < Q_i \leq 250000$ ，你將會獲得 $17 \times P_i$ 分。
- 若 $40000 < Q_i \leq 200000$ ，你將會獲得 $(7 + \frac{2000000}{Q_i}) \times P_i$ 分。
- 若 $Q_i \leq 40000$ ，你將會獲得 $68 \times P_i$ 分。



範例評分程式

範例評分程式以下列格式讀取輸入：

- 第一列： N
- 第二列： x_1, y_1
- 第三列： x_2, y_2
- ...
- 第 $N + 1$ 列： x_N, y_N

其中 N 如題目所述， x_i, y_i 為選手可自行設計之地雷座標，注意範例評分程式並不會幫助你檢查你所輸入的地雷分布是否符合題目限制。

請注意：使用自己上傳的測試資料進行測試時，沒有下面 MSG 描述的情形時你會得到 **Accepted**。如果你的程式被評為 **Accepted**，範例評分程式會輸出 **Accepted: N q1 q2 c**，其中 $q1$ 表示呼叫 `detect` 函式的總次數； $q2$ 表示呼叫 `defuse` 函式的總次數； c 表示你成功拆除的地雷總數量。如果你的程式被評為 **Wrong Answer**，範例評分程式會輸出 **Wrong Answer: MSG**，其中 **MSG** 格式與意義如下：

- **too many detect queries**：呼叫 `detect` 函式超過 125000000 次。
- **invalid detect query**：不合法的 `detect` 呼叫。
- **too many defuse operations**：呼叫 `defuse` 函式超過 N 次。
- **invalid defuse operation**：不合法的 `defuse` 呼叫。

在 CMS 內的附件檔案中，有一個名為「Minesweeper.zip」的壓縮檔，下載後解壓縮可以找到三個資料夾「cpp」、「py」和「examples」，資料夾的意義分別為：

- **cpp**：內部包含一個檔案「Minesweeper.cpp」，你可以直接上傳該檔案來獲得 4 分，並且往後你要寫的程式都可以參考這份檔案，並在這份檔案內進行修改、編譯及執行。

請注意，檔案內有兩行註解分別為「do not modify above」和「do not modify below」，這兩行意味著希望你盡量只更動被這兩行夾住的區域，若你對互動題並沒有很熟悉，請不要更動外面的區域，更動外面的區域不會也不可能讓你能直接拿到更高的分數。

基本上你不需要更動除此之外的檔案，但若有需求可以在理解運作原理後進行更動。

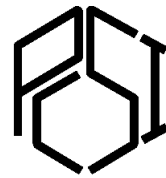
- **py**：內部包含一個檔案「Minesweeper.py」，你可以直接上傳該檔案來獲得 4 分，並且往後你要寫的程式都可以參考這份檔案，並在這份檔案內進行修改。

若要執行程式，請切換到另一個檔案「grader.py」，直接執行後「grader.py」會嘗試與你修改過的「Minesweeper.py」做連結後開始運行。請盡量只更動「Minesweeper.py」，若你對互動題並沒有很熟悉，請不要更動「grader.py」，更動「grader.py」不會也不可能讓你能直接拿到更高的分數。

基本上你不需要更動除此之外的檔案，但若有需求可以在理解運作原理後進行更動。

- **examples**：內部包含互動範例的輸入和合法輸出。

請不要嘗試撰寫題目指定需要函式以外的任何東西，例如自行輸入、輸出等，無論是



「grader.cpp」還是「grader.py」都僅供參考用，並與 Judge 上的有所落差。若有疑似偷取資料的行為，將視為作弊，嚴重者可能會取消資格並以校規處分。