108504507 盧昱瑋 final report

利用 dfs 實現踩地雷中，找出安全地塊的範圍

地塊的基本結構有

```
class node:
    def __init__(self, s):
        self.data = s
        self.locate = [-1, -1]
        self.mark = "unvisited"
        self.next = [None, None, None, None] #[up, left, down, right]
        return
```

data：紀錄此地塊是安全地還是地雷

locate：地塊在整個地圖的位置

mark：紀錄地塊是否已經被尋訪過

next：周圍四個地塊(順序為上左下右)

根據設定的地圖，將每個地塊存到由 linked list 組成的地圖中

```
def build_map(Map):
    linkMap = []

    for i in range(len(Map)):
        tmpMap = []
        for j in range(len(Map[i])):
            tmpMap.append(node(Map[i][j]))
            tmpMap[j].locate = [i, j]
        linkMap.append(tmpMap)

    print(np.shape(linkMap))

    for i in range(len(Map)):
        for j in range(len(Map[i])): #[linkMap[i-1][j], linkMap[i][j-1], linkMap[i+1][j], linkMap[i][j+1]]
            if i-1 < 0:
                #print("{0}, {1}". format(i, j))
                if j-1 < 0:
                    linkMap[i][j].next = [None, None, linkMap[i+1][j], linkMap[i][j+1]]
                elif j+1 > len(Map[i])-1:
                    linkMap[i][j].next = [None, linkMap[i][j-1], linkMap[i+1][j], None]
                else:
                    linkMap[i][j].next = [None, linkMap[i][j-1], linkMap[i+1][j], linkMap[i][j+1]]
            elif i+1 > len(Map)-1:
                #print("{0}, {1}". format(i, j))
                if j-1 < 0:
                    linkMap[i][j].next = [linkMap[i-1][j], None, None, linkMap[i][j+1]]
                elif j+1 > len(Map[i])-1:
                    linkMap[i][j].next = [linkMap[i-1][j], linkMap[i][j-1], None, None]
                else:
                    linkMap[i][j].next = [linkMap[i-1][j], linkMap[i][j-1], None, linkMap[i][j+1]]
            else:
                #print("{0}, {1}". format(i, j))
                if j-1 < 0:
                    linkMap[i][j].next = [linkMap[i-1][j], None, linkMap[i+1][j], linkMap[i][j+1]]
                elif j+1 > len(Map[i])-1:
                    linkMap[i][j].next = [linkMap[i-1][j], linkMap[i][j-1], linkMap[i+1][j], None]
                else:
                    linkMap[i][j].next = [None, linkMap[i][j-1], linkMap[i+1][j], linkMap[i][j+1]]
    return linkMap
```

利用 for 迴圈跑整個地圖，每一個地塊都跑 dfs 函式

```
def dfs(start):
    if start is None:
        return
    if start.data == "*":
        return
    if start.mark == "visited":
        return
    start.mark = "visited"
    print("{0} is visited".format(start.locate))
    for i in range(4):
        dfs(start.next[i])
```

如果地塊不存在、不是安全的、或已經尋訪過就不做 dfs，其他的地塊先標註

成已尋訪，並按照上左下右的順序對周圍四個地塊繼續尋訪。

結果：

預設地圖

```
inputmap = [["-", "-", "-", "*", "-"],
            ["-", "-", "*", "*", "-"],
            ["-", "*", "-", "-", "*"],
            ["*", "-", "*", "*", "-"],
            ["-", "-", "*", "*", "-"]]
```

尋訪順序

```
[0, 0] is visited
[1, 0] is visited
[2, 0] is visited
[1, 1] is visited
[0, 1] is visited
[0, 2] is visited
[0, 4] is visited
[1, 4] is visited
[2, 2] is visited
[2, 3] is visited
[3, 1] is visited
[4, 1] is visited
[4, 0] is visited
[3, 4] is visited
[4, 4] is visited
```