

# 迷宮產生器之演算法

## 需求

二維陣列的迷宮，並且保證有解，而且其終點解必為最遠處。

## 初步架構

- 1、迷宮走的每一步都要知道四周的格子為何，確認為無路可走時要回朔。
- 2、先定義所需資料，包含格子代表的意義、陣列每格代表的方位、走過的路徑、格子數、上下左右的向量。

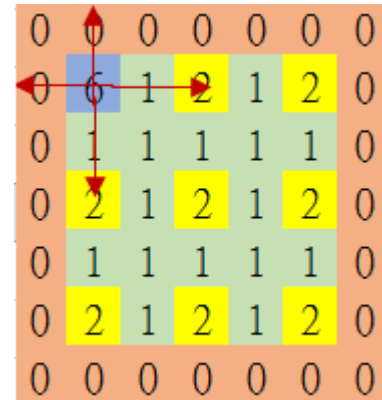
```
/*      /*      constructor(props) {
0:外框      0:上      super(props)
1:間隔      1:右      this.state = {
2:未造訪      2:下      maze: [],
3:已造訪      3:左      gridNum: 25,
4:已回朔      */
5:不合法
6:起點
7:終點
*/
}
```

- 3、並且先初始化空迷宮為(假設格子為三格)

0	0	0	0	0	0	0
0	6	1	2	1	2	0
0	1	1	1	1	1	0
0	2	1	2	1	2	0
0	1	1	1	1	1	0
0	2	1	2	1	2	0
0	0	0	0	0	0	0

## 寫碼過程

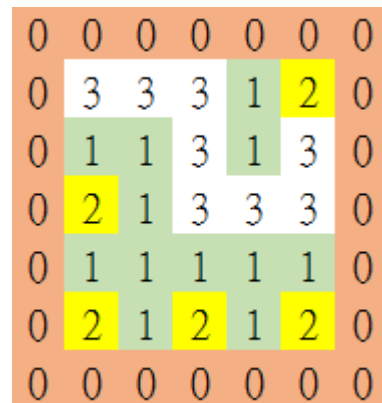
1、目前偵測XY為[1,1], 此時會抓取上右下左2格的格子為何, 此圖的結果為['05','12','12','05'], 當周遭有12時, 12的優先權最高, 從兩個12中選一個, 假設選了往右的12, 則路徑roadRun為`1`, 並且把走過的格子改為3(已造訪)



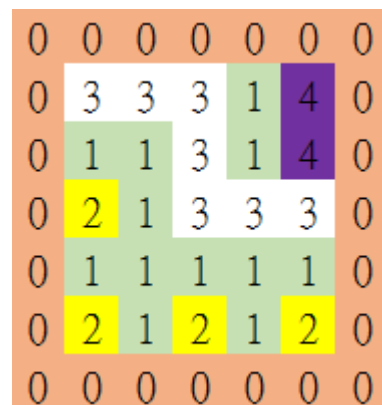
2、重複1的步驟, 此時周圍為['05','12','12','33'], 12優先權最高, 擇一。



3、目前路徑為`1210`, 周圍為['05','05','33','13'], 沒有12, 且有包含33, 可得知是回朔點, 此時把回朔點的路徑以及XY push到list, 並把路徑字串長度-1, 選擇往33走, 但走過的路改成4(已回朔)。  
roadRun = `1210`,  
backPointList=[{X:5,Y:1,pathLength:4}]

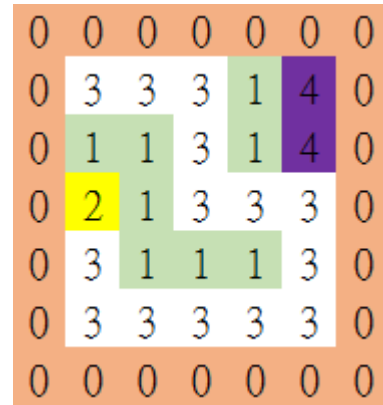


4、此時roadrun為`121`, 周圍為['44','05','12','33'], 選優先權最高的12走

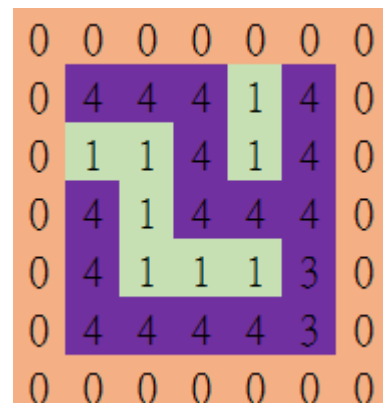


5、此時又遇到折返點, push至backPointList。

```
roadRun = `1212330`
backPointList=[
  {X:5,Y:1,pathLength:4},
  {X:,Y:,pathLength:7}
]
```



6、最後走到1,1時, 照backPointList的pathLength長度來sort, 把最長的當作終點即完成。



成果

