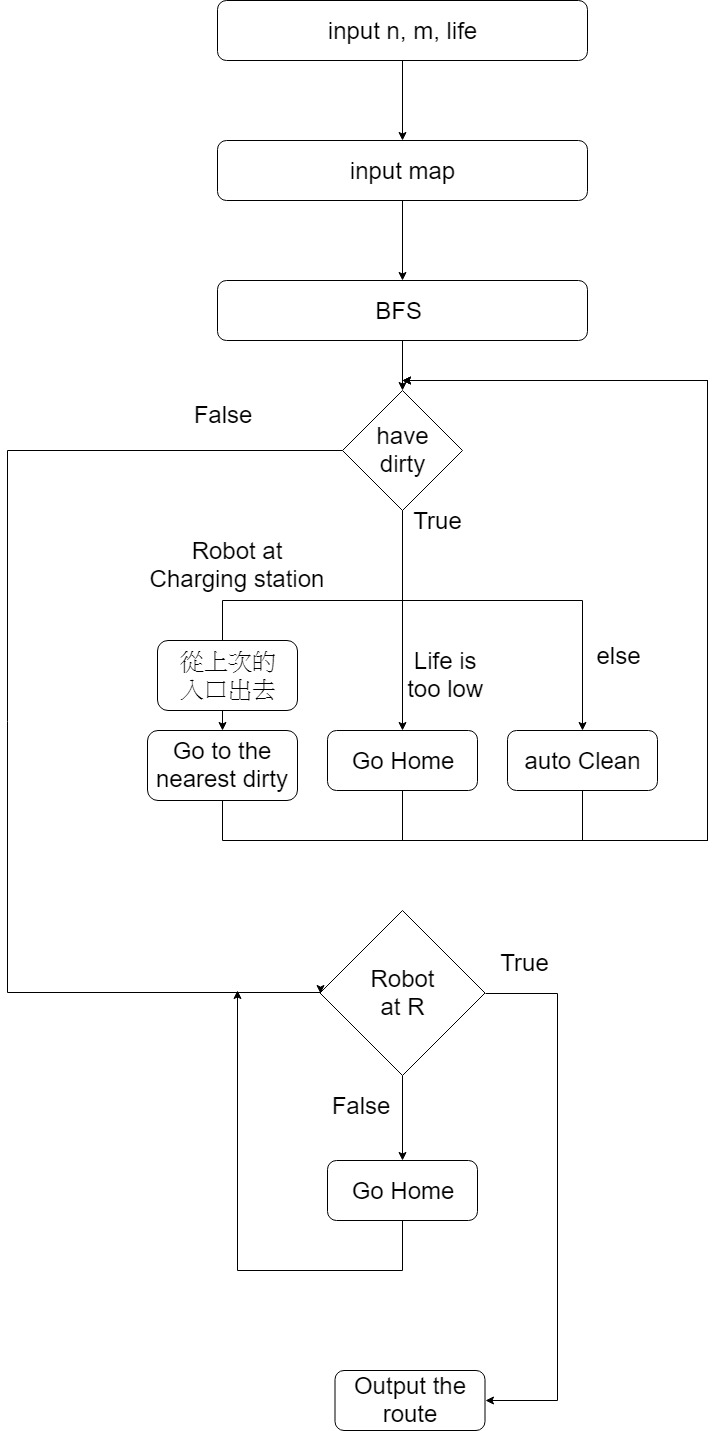
Project 2 Floor Cleaning Robot

106062130 邱詠智

1. Project Description
2. Flow Chart



1. Detailed Description
2. map: 以二維 int 陣列儲存

Charging Station ( R ) = 0, Free space ( 0 ) = 0, Obstacle ( 1 ) = 1e9

第一次BFS，Free space的位置會存入到 R 的最短距離。

以正數表示尚未清理，負數表示已清理。

Ex:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 原圖：   |  |  |  |  | | --- | --- | --- | --- | | 1 | 1 | 1 | 1 | | 1 | 0 | 0 | 1 | | 1 | 0 | 0 | 1 | | 1 | R | 1 | 1 | | 處理後：   |  |  |  |  | | --- | --- | --- | --- | | 1e9 | 1e9 | 1e9 | 1e9 | | 1e9 | 2 | 3 | 1e9 | | 1e9 | 1 | 2 | 1e9 | | 1e9 | R | 1e9 | 1e9 | |

1. Robot mode 1: Go home

從當前位置搜尋相鄰點，走向步數較小( 離充電站較近 )的地方(1~2個)，如果其中1個尚未清理，則走向未清理的地方。

觸發條件：當前life 當前位置與充電站的最短距離。

1. Robot mode 2: auto Clean

走向未清理的相鄰地區，離充電站越遠，優先度越高。

若四周都已清理過，則執行Go home。

觸發條件：

當前life 當前位置與充電站的最短距離，且不在充電站內。

( 不符合 mode 1 or mode 3 時觸發 )

1. Robot mode 3: 從上次的入口出去 + Go to the nearest dirty

為了符合 spec, 當回到起點時，要從上次的入口出去。

走出去後，執行BFS走向距離最近的未清理區域。

實作方式：

另開一張地圖，在BFS過程中，儲存此格子點從哪個方向過來。

搜尋到未清理的點，就能以”哪個方向過來”，到推出起點到此的路徑。

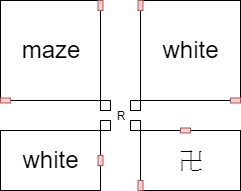
觸發條件：位置在充電站。

( 若為第一次出發則指執行BFS後的部分 )

1. 正確性：

不論處於什麼狀態下，最後一定會回到充電站再重新出發，之後以BFS掃過整張地圖，把未清理的點找出來。

理論上只要所有的0連通且最大life足夠( max life 最遠點的距離\*2 )，必定能清掃完所有地板。



1. Test Case Design

162 \* 200 的矩形，以十字型的道路分成4塊。

如右圖所示。 紅色矩形為各區塊的出入口。

1. maze ( 100\*100 ) :

各條路寬度為1, 使用右手定則可以全部走完。

若沒有走到道路底端，且下次沒有檢查，會產生多餘的步數or無法掃完整個迷宮。

1. white ( 100 \* 100, 59 \* 100 )、卍字型 ( 59 \* 100 ) :

基本測試( white )、有點變化的基本測試 ( 卍 )

1. R的位置、 life = 1009 ( 最遠距離\*2 + 1 ):

如果從左邊出來，必須要橫穿整個maze才能由上方回去充電。

由於生命值的限制，出入口不能只限制在同一方向。

若為了符合 spec，把出入口限制為一個，那就永遠跑不完。

Ex: 由左方出發，掃完迷宮後必須橫穿迷宮，從上方回去充電。

否則電量不夠支撐到進入White.