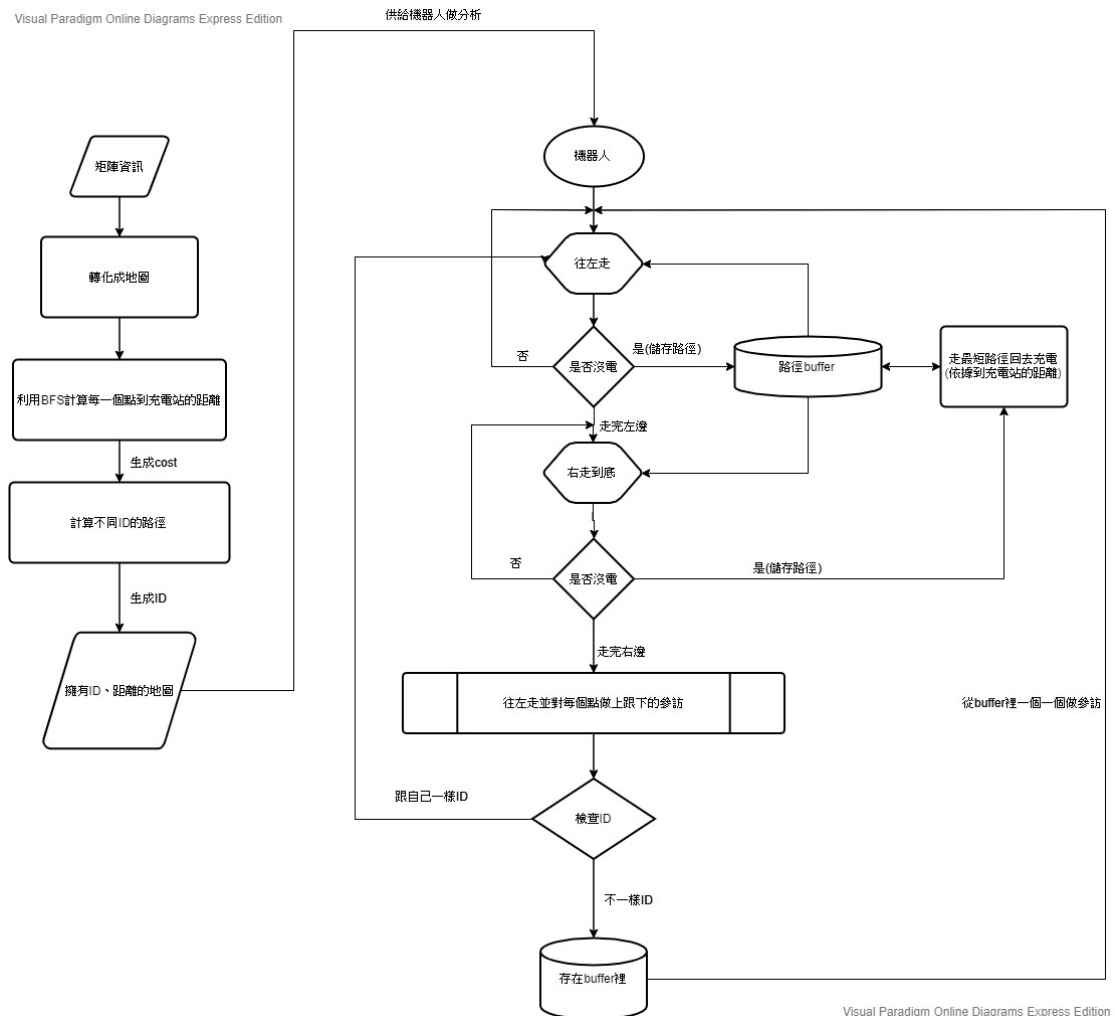


一、Projection Description

A. Program Flow Chart :



B. Detailed Description :

(一) 地圖(map)處理

1. 利用 BFS 做從 R 處路徑參訪，根據 BFS 特性，會針對鄰邊每個節點做完再參訪下個節點。根據這個特性，就能了解到任一點到 R 須走幾步。

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	12	11	10	9	8	7	6	7	8	9	10	11	12	-1
-1	11	10	9	8	7	6	5	6	7	8	9	10	11	-1
-1	10	9	8	7	6	5	4	5	6	7	8	9	10	-1
-1	9	8	7	6	5	4	3	4	5	6	7	8	9	-1
-1	8	7	6	5	4	3	2	3	4	5	6	7	8	-1
-1	7	6	5	4	3	2	1	2	3	4	5	6	7	-1
-1	6	5	4	3	2	1	0	1	2	3	4	5	6	-1
-1	7	6	5	4	3	2	1	2	3	4	5	6	7	-1
-1	8	7	6	5	4	3	2	3	4	5	6	7	8	-1
-1	9	8	7	6	5	4	3	4	5	6	7	8	9	-1
-1	10	9	8	7	6	5	4	5	6	7	8	9	10	-1
-1	11	10	9	8	7	6	5	6	7	8	9	10	11	-1
-1	12	11	10	9	8	7	6	7	8	9	10	11	12	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

假設 R 放置在中間，其他點的步數就能清楚知道，我將利用此步數尋找最短路徑回 R。

2. 建立每一行的 ID：每一行都有自己的 ID，並往下衍生，直到遇到牆壁或本身已有 ID。如此一來，機器人能夠辨識每條路是否屬於正在走的同一條路。

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
-1	2	2	2	2	2	2	2	2	2	2	2	2	2	-1
-1	3	3	3	3	3	3	3	3	3	3	3	3	3	-1
-1	4	4	4	4	4	4	4	4	4	4	4	4	4	-1
-1	5	5	5	5	5	5	5	5	5	5	5	5	5	-1
-1	6	6	6	6	6	6	6	6	6	6	6	6	6	-1
-1	7	7	7	7	7	7	7	7	7	7	7	7	7	-1
-1	8	8	8	8	8	8	8	8	8	8	8	8	8	-1
-1	9	9	9	9	9	9	9	9	9	9	9	9	9	-1
-1	10	10	10	10	10	10	10	10	10	10	10	10	10	-1
-1	11	11	11	11	11	11	11	11	11	11	11	11	11	-1
-1	12	12	12	12	12	12	12	12	12	12	12	12	12	-1
-1	13	13	13	13	13	13	13	13	13	13	13	13	13	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
-1	2	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	3	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	4	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	5	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	6	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	7	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	8	-1	1	1	1	1	1	1	1	1	1	1	1	-1
-1	9	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	2	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	3	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	4	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	5	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	6	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	7	-1	1	-1	1	1	1	1	1	1	1	1	1	-1
-1	8	-1	1	1	1	1	1	1	1	1	1	1	1	-1
-1	9	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

buildTravelTree()便是用來製作 ID 的 function，將會對不是牆壁的節點

做 ID 的設置 `buildLevel()`。但是實作上會發現，會有部分沒被設 ID 的狀況，原因在於第一次設 ID 時，是從最上方開始建，底下是屬於尚未設 ID 的狀態。因此我當我全部 `buildLevel()` 完後，我會再做一次 `buildLevelCheck()`，功能與 `buildLevel()` 相似，但是多了與下一行 ID 比較的功能，沒被設 ID 的路其 ID 是無限大，因此能用來找到未被設 ID 的路。

(二) Machine 的路徑規劃：

1. `cleanRow()`：此 function 為主要 function，用來判斷與分析接下來該如何走訪路徑。
 - (a) 來回走過一遍一整列。
 - (b) 會再做一次參訪，但同時分析當上面或下面為同 ID 的路，也就是為同一群的路時，優先對其參訪。反之，會先儲存在 `stack` 的 `buffer` 裡(自己寫的)，當參訪完會從 `buffer` 裡對上下不同 ID 的路做參訪。如此遞迴下去。
2. `backtoroot()`：為找尋回 R 處的路。在 `map` 已經對地圖設置 `cost` 此節點回 R 處所需的電量。此時只要一直往 `cost` 比自己小的路走，一定能到達 R 點。
3. `backToNode()`：在回 R 處的同時，會交由一陣列儲存其路經。當回到 R 時，在一剛剛的來時路走原路回去，因為剛剛的路是最短路徑。
4. 一開始時建立一個 `Queue`(自己寫的)存取 `Machine` 的走過路徑，才能在算完步數並輸出後在印出所有路徑。

二、Test case Design：

A. Detailed Description of the Test case：

主要在偵測是否能夠處理有隔板的路徑，並確保所有路徑能夠做到「掃」的動作。

三、Git History：

Initial version：

Commits on Nov 19, 2019	<div>11/19 ...</div> <div>sin committed 3 days ago</div> <div>7e9d80d</div>
Commits on Nov 17, 2019	<div>current commit ...</div> <div>sin committed 5 days ago</div> <div>ad53420</div>
Commits on Nov 16, 2019	<div>initial_2and3Pass</div> <div>sin committed 6 days ago</div> <div>b8e546f</div>
	<div>OLD version</div> <div>sin committed 6 days ago</div> <div>63a5981</div>
	<div>temporary version</div> <div>sin committed 6 days ago</div> <div>0621282</div>
	<div>temporary version</div> <div>sin committed 6 days ago</div> <div>72a0754</div>
Commits on Nov 15, 2019	<div>Uncomplete version</div> <div>sin committed 8 days ago</div> <div>89a153e</div>
	<div>Initial commit</div> <div>sin committed 8 days ago</div> <div>847db5a</div>


Revise version :

Commits on Nov 16, 2019	<div>initial_2and3Pass</div> <div>sin committed 6 days ago</div> <div>b8e546f</div>
	<div>OLD version</div> <div>sin committed 6 days ago</div> <div>63a5981</div>
	<div>temporary version</div> <div>sin committed 6 days ago</div> <div>0621282</div>
	<div>temporary version</div> <div>sin committed 6 days ago</div> <div>72a0754</div>
Commits on Nov 15, 2019	<div>Uncomplete version</div> <div>sin committed 8 days ago</div> <div>89a153e</div>
	<div>Initial commit</div> <div>sin committed 8 days ago</div> <div>847db5a</div>

Branch: revise1 ▾

Commits on Nov 22, 2019

11/22 ...


 sin committed 3 hours ago



a846476



11/22 ...

 sin committed 16 hours ago




a443e01



Commits on Nov 21, 2019

11/21 ...

 sin committed yesterday




a575532



Commits on Nov 19, 2019

11/19 revise ...


 sin committed 3 days ago



9c48f5c



11/19 ...

 sin committed 3 days ago




7e9d80d



Commits on Nov 17, 2019

current commit ...

 sin committed 5 days ago



ad53420

