

## AI-hw1 report

r12922a12

### ● Autograde result

```
Provisional grades
=====
Question q1: 5/5
Question q2: 5/5
Question q3: 10/10
Question q4: 15/15
Question q5: 5/5
Question q6: 9/9
-----
Total: 49/49
```

### ● Depth First Search (1%)

使用 `stack (LIFO)` 來儲存可以走的路，從起始點開始朝著一個方向不停地向下尋找，當此條路徑沒有可以繼續往下尋找的點後，回溯至上一個分岔處，繼續探索下一條路徑。當找到目標點即回傳路徑

### ● Breadth First Search (1%)

使用 `queue (FIFO)` 儲存可以走的路，從起始點開始，先探索起始點周圍的點，之後再向外一圈進行探索，以此類推，直至找到目標點，回傳路徑，此時的路徑會是最短路徑

### ● Uniform Cost Search (1%)

使用 `priorityQueue` 儲存點以及其 `cost (current node cost + previous cost)`，最開始的點 `cost = 0`，從起始點開始向外探索，每次找 `priorityQueue` 中累積 `cost` 最小的點進行探索，直至找到 `goal` 後回傳路徑，確保回傳最短路徑。

### ● A\* Search (null Heuristic) (1%)

`null heuristic` 即  $h(n) = 0$ ，

A\* 演算法利用  $f(n) = g(n) + h(n)$  計算路徑的 `cost`，當  $h(n) = 0$  時成本只考慮路徑成本  $g(n)$ ，此狀況下演算法基本上與 `uniform cost search` 相同，利用 `list` 儲存各可以探索的點和其累積的 `cost`，每次選擇累積 `cost` 最小的點進行探索，找到目標點即回傳路徑。

### ● Breadth First Search (Finding all the Corners) (1%)

利用 `BFS` 方法找尋所有的角落

1. 取得現在 `state` 的位置
2. 確認 `state` 的四周的點
  - a. 確認沒有障礙物的方向點是否是 `corner`，  
是 `corner` 則標記 `corner` 為已找到  
不是 `corner` 的話將位置計入 `successor` 在下一步繼續尋找

- A\* Search (Corners Problem: Heuristic) (1%)

cornersHeuristic 函式希望 return A\* algorithm 的  $h(n)$  (cost):

1. 得到目前的位置以及 corner 的位置
2. 得到目前還沒有找到的 corner
3. 假設目前的位置與各 corner 的距離是 manhattanDistance
4. 更新目前 state 的 heuristic\_cost = sum of the manhattanDistance of current state with the unvisited corners
5. return heuristic\_cost

- Describe the difference between Uniform Cost Search and A\* Contours (2%)

Uniform Cost Search 和 A\* 主要的不同在於：是否使用了 heuristic cost 來估計目前的點到目標點的剩餘成本，這幫助 A\* 能夠更有效率地往目標方向尋找，使得通常 A\* 可以更快地找到最優解，相較之下 Uniform Cost Search 僅考慮已經產生的路徑成本，不考慮到達預估目標的剩餘成本。

- Describe the idea of Admissibility Heuristic (2%)

Heuristic Cost 是估計從任一點到目標點的成本的啟發式，一個 admissibility Heuristic 在於其估計值永遠小於等於任一點到目標點的最低成本，同時確保演算法不會為了似乎更短的路徑忽視實際更短的路徑，幫助 A\* 演算法高效率且確保尋找到最短路徑。