人工智慧導論 Program assignment#2

學號：411086006 學生：張愷和

所有程式皆使用C++，並用wsl上linux環境中的gcc編譯檔案

介紹一些宣告在global scope的參數：



N 棋盤大小

max\_saved 儲存state狀態大小

一張含有 文字 的圖片

自動產生的描述

物件的一些參數

cost:state到達的步數

board:state現在的盤面

hvalue:state的heuristic數值

print():印出state盤面，以Q表示queen位置，以 . 表示空位

一張含有 文字 的圖片

自動產生的描述

各個search的priority\_queue comp function

一張含有 文字 的圖片

自動產生的描述

將輸入的parent node展開為child node，將以展開的child輸入vector<obj> child\_node回傳至各個search去判斷，首先先使用一個array紀錄有哪些row已經有放queen，接著將child複製parent盤面，對child下一行(沒有queen的那行)放置queen，因此假設為8\*8的盤面，就會產生8個child，我在對放置queen再多做一下解釋，假設第一行以及第三行有放queen，那就會先從第二行expand，假設第一行以及第二行有放queen，就會先從第三行放queen。

一張含有 文字 的圖片

自動產生的描述

判斷盤面上的每個queen會不會互相攻擊，f是記錄在第i row上的queen在哪一column。

一張含有 文字 的圖片

自動產生的描述

判斷有沒有達到goal的狀態，先判斷state的盤面有沒有解，有解就判斷queen的數量有多少，如果數量到達8個(8\*8)或15個(15\*15)，就回傳有解。

一張含有 文字, 螢幕 的圖片

自動產生的描述

計算現在盤面上有多少queen，越多queen就先search，當作heuristic value。

一張含有 文字 的圖片

自動產生的描述

Ucs ，先將start輸入進priority\_queue，將max\_state加1，紀錄初始盤面，接著先去判斷有沒有解，如果有解就把當前state經過的cost、expand紀錄的盤面數值以及goal state盤面輸出，如果沒有解，就把parent展開，接著判斷每個child是否有解，如果有解就push進pq，如果沒解就忽略，到最後priority\_queue search完沒有解，輸出No solution。

一張含有 文字, 監視器, 螢幕 的圖片

自動產生的描述

Ids ，限制dls深度，如果在限定深度沒有找到解，就將深度+1，然後繼續重新search。

一張含有 文字 的圖片

自動產生的描述

dls ，先使用stack紀錄start state，將max\_state加1，紀錄初始盤面，用stack可以達到讓第一個state一直往下搜索的狀況，接著先去判斷有沒有解，如果有解就把當前state經過的cost、expand紀錄的盤面數值以及goal state盤面輸出，如果沒有解，接著判斷state search的深度是否到了限制深度，如果沒有就把parent展開，接著判斷每個child是否有解，如果有解就push進stack，如果沒解就忽略，到最後stack search完沒有解，輸出No solution。

一張含有 文字 的圖片

自動產生的描述

Greedy search，search的方法跟ucs差不多，只是將cost的優先大小排列，改成以heuristic value優先大小排列，而heuristic計算的方法，是在判斷要不要儲存expand出來的child時，如果child有解，就先計算child的heuristic數值，然後push進pq繼續search。

一張含有 文字 的圖片

自動產生的描述

Astar search ，跟greedy search差不多，只是在pq排列的方法，由hvalue小到大優先排列，更改為hvlaue+cost大小，由小到大優先排列，排列完後就將最前面的state進行search。

一張含有 文字 的圖片

自動產生的描述

RBFS，主要的概念是將pq中最佳以及第二佳解的state留下去用recursive expand，而剩下的拋棄掉，不過在這題目中，拋棄掉的狀況有可能是最佳解，因此最後找到的解有可能不是最佳解，來介紹一下程式好了，首先判斷有沒有解，如果沒解就對node進行expand，接著判斷child有沒有解，有解就push進pq search，接著就進while迴圈進行挑選最佳以及第二佳的state，接著將這兩個state進入recursive下去expand，直到找到解，如果沒有找到解就return 1e9，return 1e9到RBFS function，輸出No solution。

一張含有 文字 的圖片

自動產生的描述

在main function裡就輸入有多少queen以及排列queen的位置，接著判斷初始盤面有沒有解，如果有解就進行每一個search，如果沒有解，就輸出No solution。

至於15-queen，就只是把global scope宣告的N = 8改成 N = 15，15-queen就可以解了。