

人工智慧導論 Program assignment #2

不能用 DevC++跑!!!要用 CodeBlock!!!因為 DevC++的預設編譯器不支援很多我所使用的語法!!!

1. 程式架構
 - A. 設定輸入樣式並設置棋盤的初始狀態
 - B. 使用函式 Solution()輸出演算法執行的過程與最終結果
 - C. 使用函式 Check_Is_Save()檢驗所有皇后是否安全，避免皇后間彼此能互相攻擊
 - D. 使用不同的搜尋法解 8 queens。以下將會詳述各搜尋法的細節。
 - E. 若要改成 15 queens，#define QNUM 要改成 15，#define Col 要改成 14。(我這裡有直接改再弄一個 CPP 給 15 皇后)
2. 各搜尋法的細節
 - A. IDS
 - (1) 每次要放皇后進去棋盤的時候就先檢查是否會受到其他皇后攻擊，放妥後增加樹的深度，也就是移至下一個 column 並放置下一個皇后到棋盤上。
 - B. UCS
 - (1) 每次在放新的皇后時挑選總花費最少的位置，使得皇后之間的距離最靠近，也就是最緊密的擺法。
 - C. GBFS
 - (1) 使用 heuristic function 計算將皇后擺放到每一個格子上會有幾種被攻擊的可能，利用 priority queue 每次選擇 heuristic function 的值最小的格子將皇后擺放上去。
 - D. A*
 - (1) 同 GBFS 的兩個步驟，建立 heuristic function 以及 priority queue。
 - (2) 再決定下一個皇后的擺放位置時，將每一個可能成為下一個擺放位置的格子的花費以及從 heuristic function 得到的值相加，以此總花費挑選花費最少的格子作為下一個皇后的擺放位置。
 - E. RBFS
 - (1) 利用 priority queue 依照皇后擺放到每個格子所需的花費做排序。
 - (2) 然後利用 HashTable 紀錄走過的路徑，以防止當 Limit 被改過後再次回到初始狀態。
 - (3) 不斷更新可能會被擺放到的格子的總花費。
 - (4) 選擇擺放所需花費最少的格子作為下一個皇后的擺放位置。

