

AI HW2

1.

1. **False.** rational 的定義是指對目前有的輸入可以選擇最好的決定，與整體資訊無關
2. **True.** pure reflex agent 只能對現在當下的輸入做出反應。令一 agent 為按鈕開關，按一下為開啟，按兩下為關閉，則 pure reflex agent 無法完成此任務（無記憶功能）
3. **False.** agent program 所接收的是當下的資訊，而 agent function 可以統合以前整個 program 接收的資訊作為輸入，產出反應
4. **False.** agent 所擁有的動作和決定仍會造成它是否 rational。一個永不做出反應的 agent 也不可能是 rational 的。
5. **False.** 撲克牌有機率性。以大老二為例，若有人直接拿到一條龍，則 agent 也不可能不輸。

2.

1. States : 若有 n 塊區域，則有 5^n 種狀態

Initial state : 全不塗色

Actions : 塗顏色1、2、3、4

Transition model : 如 Action 所述

Goal test : 所有臨格顏色均不相同

Path cost : 塗色的次數

2. States : (猴子所在高度, 房間 x 座標, 房間 y 座標)

Initial state : (0, 任意, 任意)

Actions : 爬 0~8 尺、搬箱子到(x , y)、疊箱子

Transition model : 如 Action 所述

Goal test : 到達香蕉所在座標

Path cost : 動作次數

3. States : 在已讀檔案中的紀錄數

Initial state : 第0個紀錄

Actions : 檢查當前紀錄、印出錯誤訊息

Transition model : 檢查完當前後查下個

Goal test : 印出錯誤訊息

Path cost : 檢查次數

4. States : (0, 0, 0) ~ (12, 8, 3) for 12, 8, 3 加侖的瓶子

Initial state : (0, 0, 0)

Actions : 一瓶倒到另一瓶、加水加滿、整瓶到掉

Transition model : 如上述

Goal test : 其中一瓶是1加侖

Path cost : 倒的次數

3.

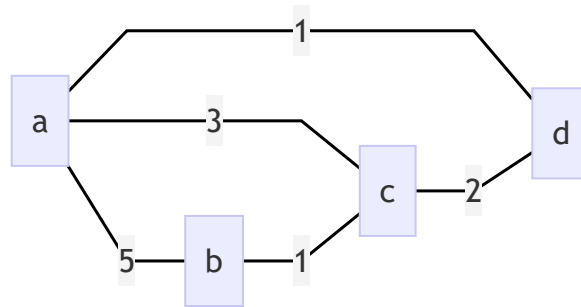
1. state : 在解題過程中的節點
2. state space : 所有同系統中狀態的集合

3. search tree : 一種樹狀的搜尋結構，可以從父節點依序往下搜尋到葉節點
4. search node : 搜尋樹上的各個節點
5. goal : 一問題最終要完成的事情
6. action : 針對一狀態可以執行的操作
7. transition model : 一狀態經一個 action 後轉一到下一個狀態的描述
8. branching factor : 一狀態在搜尋樹上經一 action 後產生的狀態數

4.

1. False.

以下圖為例：



若要從 A 到 B，則 DFS 會在一步時找到，但 A* 會走 A-D-C-B 以達到最短路徑。

2. True.

$h(n) = 0$ 會使得只能看到過去的狀態，因此不會高估到目標的花費。

3. False.

在 A* 中我們能可使用離散資料來近似連續資料

4. False.

若該樹有無限節點，就可能找不到，因此不是complete

5. False.

若可以一次移動多格，曼哈頓距離會錯估步數

5.

1. True.

在 fully observable 且雙方都是 rational 的情況下，可以直接預期對手下一步的動作而無須知道對手的策略

2. False.

以撲克牌吹牛為例，若能知道對方喊牌的策略，則一定會有幫助

3. False.

backgammon 為機率性的遊戲，因此不可能有永遠不輸的情況