

1. (a) rational 的定義是指對目前有的輸入可以選擇最好的決定，無關整體資訊。  
False #
- (b) 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。 True #
- (c) agent program 所接收的是當下的資訊，而 agent function 可以統合以前整個 program 接收的資訊做為輸入，產出反應。  
False #
- (d) agent 所擁有的動作和決定仍會造成它是否 rational。一個永不作出反應的 agent 仍不會是 rational。  
False #
- (e) 撲克牌仍有機率性，若以大老二為例，若有人直接拿到一條龍，則 agent 也不可能不輸  
False #

2. (a) 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: 塗色後即會轉移到該格塗色後的狀態  
Goal test: 所有鄰格顏色皆不同  
Path cost: 塗色的次數

(b) States: (猴子在的高度  $0 \sim 8$  foot, 房間平面座標)

Initial state: (0 foot, 任一平面座標)

Actions: 爬  $0 \sim 8$  foot、疊箱子、移動箱子

Transition models: 如動作所述

Goal test: 到香蕉所在位置

Path cost: 動作的次數

(c) States: 在已讀檔案中的記錄數

Initial states: 第0個記錄

Actions: 檢查當前的記錄、印出錯誤訊息

Transition models: 檢查完當前檢查下個、印出訊息後停止

Goal test: 印出錯誤訊息

Path cost: 檢查次數

(d) States:  $(0, 0, 0) \sim (12, 8, 3)$  for 12, 8, 3 gallons jugs

Initial states:  $(0, 0, 0)$

Actions: 一瓶倒到另一瓶、加水加滿、整瓶倒地上

Transition models: 如上述

Goal test: 其中一瓶是1加侖

Path cost: 倒的次數

3.

① state: 在解題的過程中的結點

② state space: 所有同個系統中的狀態的集合

③ search tree: 一種樹狀的搜尋結構, 可以從父節點依序往下到各子節點

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

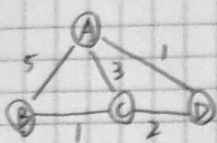
⑥ action: 針對一狀態可執行的操作

⑦ transition model: 一狀態經一個 action 後轉移到下一狀態的描述

⑧ branching factor: 一狀態在搜尋樹上經一 action 後可能產生的狀態數

4. (a) False #

如果有下列圖：



若要從 A 到 B，則 DFS 會在一歩時找到，

但 A\* 會走 A → D → C → B 以達到最短路徑

(b) True #

$h(n)=0$  會使得只能看到過去的狀態，因此不會 overestimates 到目標的 cost.

(c) False #

在 A\* 中，我們仍可使用離散資料來做近似

(d) False #

如果該搜尋樹有無限節點，就有可能找不到，因此不是 complete

(e) False #

如果該石頭可以一次移動多格，則 Manhattan distance 會錯估距離(步數)

5.

(a) True #

在 fully observable 且雙方都是 rational 的情形，則可直接選期下一步對方的動作，與對方的策略無關

(b) False #

以撲克牌的吹牛為例，若知道對方吹牛的策略，則一定會有幫助

(c) False #

backgammon 為機率性的遊戲，因此不可能有永遠不輸的情況