

## 人工智慧作業二：標靶治療

Due: 2023-04-07

網路上有一個好玩的益智遊戲「標靶治療」  
<https://www.novelgames.com/zh-HK/targetedtherapy>，  
你玩了以後，腦筋一定會越來越好(林老師玩了也有變聰明了)。

- 一開始癌細胞會分布在一直線上。
- 每次你選擇一顆癌細胞來消滅它，其它癌細胞就會分裂成兩個，離開原本的地方移到其左右方的位置上。兩顆移到同一位置的癌細胞會結合為一。
- 繼續用這個標靶治療方法消滅癌細胞，直至把所有癌細胞都消滅，你就完成目標了。

現在你即將成為這個難題的解題設計師，  
未來可能有機會被AI公司延攬，因此你要好好  
做這一題喔。



HW2-1

• 本作業將要求你使用第3章的Iterative Deepening Search(IDS)及  
Iterative Deepening A\*(IDA\*)，分別撰寫兩支程式來解決這個問題  
操練的要項

- 盤面怎麼表示？走步要如何產生？節點要存那些資訊？如何判別  
重覆？會不會跑不停？記憶體會不會爆掉？所得結果會是最佳解  
嗎？使用那一種heuristic結果比較好？如何估算時間及空間的消  
耗量？
- Hint：此題亦可參考網路上的開源碼程式來修改。  
但請你一定要註明原始程式碼的來源。

**作業報告(請務必依序撰寫，我們才能逐項評分)**

請將兩支程式之原始碼、執行檔、測試用的輸入檔(請依照輸入格  
式，我們才有辦法測試)及作業報告上傳到moodle。請務必確認系  
統已收到你的資料。

- 詳細說明你所使用之機器軟硬體規格及作業系統、開發軟體版  
本、如何執行程式相關資訊。另外請提供你的連絡電話，以便  
不時之需。
- 首先你先設法製作測試用的輸入檔至少5個。盤面尺寸需有大有  
小，解題難度也不同，由你決定。請說明你如何製作這些測試  
用的輸入檔。(提示：可考慮用亂數產生)
- 兩支程式之原始碼中應加註解，請說明如何執行這兩支程式。

給定的初始盤面不一定每個位置都有癌細胞。

輸入檔格式(一律存在c:\input.txt或自訂位置)

1 1 1 0

輸出檔格式(一律存在c:\output.txt或自訂位置)

Total run time = 0.004 seconds.

An optimal solution has 3 moves:

2 2 3

你只需輸出一組最佳解即可。

如果無解(會有無解的嗎?)，則請輸出：

Total run time = 0.08 seconds.

There is no solution.



HW2-2

- 請說明第一支程式IDS你使用甚麼方法、甚麼資料結構、  
甚麼技術(操練要項)來解決這個問題，並請說明你測試一  
些盤面時的表現如何、耗用的時間及空間為何(假設盤面  
有n個位置)、你的程式能解到多大盤面的題目？請你用一  
些例子輔助說明。(解題愈快、說明愈清楚的程式成績愈  
高。)
- 請說明第二支程式IDA\*你使用甚麼方法、甚麼資料結構、  
甚麼技術(操練要項)來解決這個問題，並請說明你測試一  
些盤面時的表現如何、耗用的時間及空間為何(假設盤面  
有n個位置)、你的程式能解到多大盤面的題目？請你用一  
些例子輔助說明。(解題愈快、說明愈清楚的程式成績愈  
高。)
- 請說明你做此作業所碰到的一些狀況及困難。
- 請列出你的參考文獻(含網站)來源，並請說明  
參考了那些部份用於作業中。





## 注意事項

- 本課程允許遲交作業(三天內)，遲交的話，一天內約扣2分，二天內約扣4分，三天內約扣6分。
- 提醒：報告及程式應親自為之。報告如有抄襲同學之狀況(不管被抄襲或抄襲別人)，一律依校規送校方處理。請你也要保護好你的智財權，別讓別人勞而獲。因此作業請勿抄襲同學的作業。根據以往的經驗，非常容易被發現。(因為修課人數不是大數據的規模)
- 作業報告著重於解決這個問題的構想、理由，還有實驗設計以及結果說明、分析的完整性，即使實驗數據不好看、測試效能不佳，只要有完整的結果分析、或是說明有獨到的見解都能拿高分。換句話說，報告的評分重點在你對於作業的用心程度，自己構思所佔的部份是否足夠，而不僅僅是測試效能、實驗數據的美觀。

8. 額外加分：如果你能利用一些現有的或自己想出來的進階技術，來得出一些成果，則另有加分。範例如下：

破解：在某個 $n$ 範圍內，證明所有初始盤面均有解，或哪些 $n$ 均有解，哪些 $n$ 就不一定有解。可能的高階技術是bitboard。

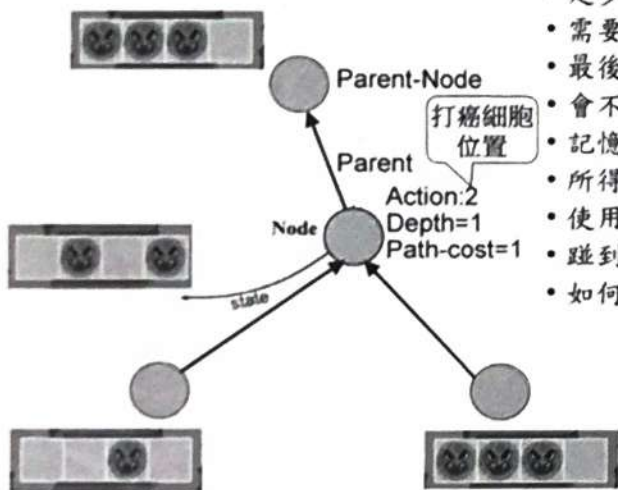
製作殘局庫：用回溯法(Retrograde Analysis)，在少量癌細胞或較小 $n$ 的盤面，得出打癌細胞位置，存入殘局庫中。

額外苦功：除IDS及IDA\*外，實作第四章的任一個演算法，看看有沒有更棒？

推廣：若些遊戲推廣至「二維盤面」，規則如何改變？還會有解嗎？遊戲還能玩嗎？

天才解法：有辦法不用搜索就知道解法嗎？或不需指數時間就得到解法嗎？

## 操練的要項



- 盤面怎麼表示？
- 棋盤全部資訊要存在每個節點嗎？
- 節點要存那些局部資訊？
- 走步要如何產生？
- 需要判別重覆？
- 最後答案如何取出來？
- 會不會跑不停？
- 記憶體會不會爆掉？
- 所得結果會是最佳解嗎？
- 使用那一種heuristic結果比較好？
- 碰到dead end可提早backtracking？
- 如何估算時間及空間的消耗量？

## bitboard高階技術

[https://www.chess.com/blog/the\\_real\\_greco/move-finding-the-engine-way](https://www.chess.com/blog/the_real_greco/move-finding-the-engine-way)

## heuristic $h(n)$ 聰明的估算

