

CPP1702 Coding Assignment 4

海戰棋問題

HW41 電子檔 2018/05/04 09:00 前上傳至 moodle
 HW42&43 電子檔 2018/05/14 09:00 前上傳至 moodle

運用在工業與資訊管理的範圍

基本邏輯推導、智慧決策機制之分析與設計。

相關 C++ 課程內容

讀檔、結構、陣列、向量 vector、隨機選取 rand()。

問題背景與概述

「海戰棋」(Battleship Board Game) 是一款簡易理解的雙方對戰遊戲，給定一個網格棋盤平分一半，雙方各自將數艘戰艦按規則（譬如不得相鄰或置於對角）佈署在各自網格內，遮起來不讓對方看到己方佈署，再輪流發射炮彈至對方海域，一枚砲彈轟炸一網格，直至對方艦隊全數被擊沈為止。本次作業簡化此遊戲的設定，讓同學僅扮演攻擊方，讀入給定的艦隊佈署海域地圖資料後，在沒有偷看答案的情況之下，一次發射一枚炮彈，實作兩種射擊策略：(1)「亂槍打鳥」策略(Random Search, RS)，(2)「智慧搜尋」策略(Intelligent Search, IS)，先以小例子列印各步驟結果以確認步驟正確性，最後測試7×14、8×13、10×10、12×12各 2500 個總共 10000 個海域圖之後，統計兩種射擊策略的命中率，以及該命中率相對於其下限之改善幅度比率（命中改善率）。

以下以圖一簡介本問題之設定：首先，我們將海域劃分成橫向 M 格、縱向 N 格的 $M \times N$ 個網格區域，每個網格區域以其中心點座標 (i, j) 代表之，其中 $i = 0, \dots, M-1$, $j = 0, \dots, N-1$ 。假設我們以由上而下、由左至右的方式從 0 依序將這些網格編號，亦即第 (i, j) 格之編號為 $i * N + j$ ，如圖一(a)所示。若在圖上擺置長度佔 2、3 網格之戰艦各 2 艘，戰艦只能橫或直著擺置（不能沿對角放），各船之周圍網格（橫縱相鄰格與對角格）必須為海水，網格內以 0、1 分別代表海水與船身，如圖一(b)所示稱之為「解答圖」；「作戰圖」則為對方海域，由全部網格初始為 0 開始，如圖一(c)所示。

	0	1	2	3	4	5
0	0	1	2	3	4	5
1	6	7	8	9	10	11
2	12	13	14	15	16	17
3	18	19	20	21	22	23

(a) 4×6 海域網格編號與二維座標 (i, j)

	0	1	2	3	4	5
0	1	1	1	0	1	0
1	0	0	0	0	1	0
2	1	1	0	0	0	0
3	0	0	0	1	1	1

Length	#
2	2
3	2

(b) 解答圖：2、3 網格長度各有 2 艘船及其位置

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

2	0	0	-1	0	-1
0	0	-1	0	2	0
0	2	0	0	-1	-1
0	-1	0	2	0	0

2	2	2	-1	2	-1
-1	0	-1	0	2	-1
2	2	0	0	-1	-1
-1	-1	0	2	2	2

Status	meaning
0	unknown
-1	missed shot
2	hit target

(c) 作戰圖：開始時全海域狀況皆未知(0) → 依序射擊 10 發在(1,2), (2,1), (2,4), (3,1), (0,0), (0,3), (3,3), (1,4), (2,5), (0,5) → 再依序射擊 9 發在(3,0), (2,0), (3,5), (1,0), (0,2), (0,4), (3,4), (1,5), (0,1)

圖一. $M=4, N=6$ 之海域擺放與射擊範例

開始之後，假設我們使用 RS 策略，隨機選取當下作戰圖上數值為 0 的網格來射擊，依其結果可分成「落空」(missed shot，在作戰圖上標註「-1」)跟「擊中」(hit target，在作戰圖上標註「2」)兩種結果。舉例來說，若我們依序射擊 10 發在(1,2), (2,1), (2,4), (3,1), (0,0), (0,3), (3,3), (1,4), (2,5), (0,5)，則作戰圖將變成如圖一(c)中間所示，當下為止的命中率為 $4/(4+6)=40\%$ ；若再依序射擊 9 發在(3,0), (2,0), (3,5), (1,0), (0,2), (0,4), (3,4), (1,5), (0,1)，則命中所有 $2 \times 2 + 3 \times 2 = 10$ 個船格，當下為止的命中率為 $10/(10+9)=52.63\%$ ，在其下限值 $10/(4 \times 6)=41.67\%$ 可改善的空間中 $(100\%-41.67\%)$ ，改善了 $100. \times (52.63-41.67)/(100.-41.67) = 23.14\%$ 。

以上舉例使用 RS 策略，雖然命中率可能不太好，但至少其每次的隨機選取不會重複選出先前已經選過的網格，亦即每次一定選取某個當下狀態仍為「0」的網格，不會選到「-1」或「2」等先前已選過的網格，這點請同學務必遵守。

不過，藉由戰艦的擺放規則與已知的條件，我們可以進一步分析建構一些更具「智慧」的射擊方式，提昇命中率，這個部分我們將開放給同學們自行構思，而評分亦將根據同學們提出的 IS 策略優劣有所調整（射擊次數越少，即命中率越好得分越高）。在此我們頂多先提示同學一些或許可用的 IS 策略供參考：

- **選取網格法則**：若能將狀態為 0 的網格再依其擊中之可能性細分成 4 群，分別標註其狀態為 4、3、1、0，其中標註為 4 之網格為最有可能為「船身」，3 次之，1 則代表絕對不可能（必為海水），0 則仍無法判斷。如此一來，優先選取 4 那群網格，再來是 3 那群，最後才是 0 那群，而 1 那群絕對不選。至於該如何將原來為 0 的網格如此細分？就考驗各位同學的「智慧」了。
- **依射擊結果標註未知網格之法則**：這是本次作業最核心的部分，也就是你的程式「智慧」之所在。各位必須自行分析推理，譬如若「落空」的話，可否依該網格之相鄰網格的狀態（譬如若其上下左右網格中有狀態為「2」的船身格，則？？？）；反之，若「擊中」的話，可否推論出其鄰近某些網格的狀態必為？？？建議各位先用紙筆推論出規則，再試著用程式實作之。舉戰艦長度為例，倘若目前尚有長度至少為 2 的戰艦尚未被擊沉，則剛被擊中的網格之鄰近網格應該會有較高的機會為船身，因此至少可將其鄰近狀態為 0 的網格之狀態修改成 3 或 4。此外，若以上推論確認了某些新或舊的網格狀態，或可再乘勝追擊、如法炮製，依據剛剛推論而得的某些網格狀態，同樣檢驗其鄰近網格的狀態，再接著推論出其它某些網格的狀態。如此一來，可用程式盡量將可再被推論出的網格狀態繼續推論下去，舉一反三、連鎖推理，可能將原先狀態為 4、3、0 的網格推論出其狀態應為 1、或是將狀態為 3、0 的網格推論出其狀態應可為 4、或是將狀態為 0 的網格推論出其狀態可變為 3 等等，因此而能避免誤擊中狀態為 1 的網格，優先選狀態為 4、3 的網格射擊，或可提早擊中船身網格，提昇命中率。

以上之 IS 策略理應包含某種程度的隨機性，較不會受到特殊擺置方式的影響。舉例來說，在網格選取法則部分，倘若現在狀態為 4 的網格有 10 個，程式理應「隨機」地自這 10 個網格選取，而非一律依優先選取編號較小的網格。為能公平量測程式的「智慧」性，我們提供 10000 組隨機排列的測試網路供同學測試，如此一來其平均的命中率較有參考性。由於每個測資的船格與總格數不一，命中率的理論下限值為「船格數/總格數」，我們擬再計算每組測資之「命中改善率」= $(\text{命中率} - \text{命中率下限}) / (100\% - \text{命中率下限})$ ，全部 10000 組命中改善率之 mean、medium、stdev。

作業目的及假設

本次作業主要分三部分：PART1 讀檔、以 struct 結構儲存各網格 l 之相關資料（包含橫縱座標、編號、狀態、解答等），將不同狀態的網格分別使用 vector 儲存不同狀態的網格，實作 RS 策略並列印過程；PART2 進一步實作 IS 策略並列印過程，撰寫 IS 設計概念於書面報告；PART3 大量測試兩種策略的表現，計算並輸出其平均命中率，撰寫資料分析結果於書面報告。PART1&2 將訓練 vector 使用方式，以及分析推理實作能力；而 PART3 則練習大量處理分析資料的能力。本題

假設戰艦只能橫或直（不可斜角）著擺置，各船之周圍網格（橫縱相鄰格與對角格）必須為海水。

程式要求及作法

本作業大概可切割成以下三個 PART：(PART1&PART2 各自單一程式、PART3 使用 batch 檔)

PART1[40%] 宣告、讀檔、RS 策略：[2018/05/04(星期五)09:00 上傳程式]

請使用相同的命名方法宣告下列結構、變數及動態陣列，額外需求的自訂變數可自行命名。

```
struct Zone{           //此為各網格之 struct
    int row;           //儲存其縱軸座標，即由上而下第 row 列，row=0,...,M-1
    int col;           //儲存其橫軸座標，即由左而右第 col 行，col=0,...,N-1
    int id;            //儲存其編號，id=0,...,MN-1
    int status;        //儲存其狀態，status= -1,0,1,2,3,4（或可再自行定義）
    bool isB;};        //儲存其答案，若為船身則 isB=1，否則為海水 isB=0

int M, N,              //海域之縱、橫軸總格數
    n_BT,              //戰艦類型(Battleship Type)總數，譬如 n_BT=2 代表 2 種船型
    *BTL,              //各船型之長度，譬如 BTL[1]=2 代表第 2 類船型之長度為 2
    *BTN,              //各船型之船數，譬如 BTN[3]=1 代表第 4 類船共有 1 艘
    n_targets,         //全部之船格總數，譬如圖 1(b)之 n_targets=(2+3)*2=10
    n_shots,           //記錄當下已擊發之炮彈總數
    n_hits,            //記錄當下已擊中之炮彈總數，當 n_hits==n_targets 則終止測試
    **BMap;            //記錄作戰圖，BMap[i][j]= -1, 0, 1, 2, 3 or 4
bool **SMap,           //記錄解答圖，若座標(i,j)為船身，則 SMap[i][j]=1，否則為 0
    isDebug;           //詳細模式(isDebug=1,PART1&2 用)，列出每步驟過程
                        //扼要模式(isDebug=0,PART3 用)，僅列出最後統計資料
double hr,             //命中率 hitrate 百分比=100.*n_hits/n_shots 以%為單位
    hrlb,              //命中率百分比下限=100.*n_targets/(M*N) 以%為單位
    hrimprate;         //命中改善率=100.*(hr-hrlb)/(100.-hrlb) 以%為單位
                        //代表在可改善的空間(100.-hrlb)中，hr 的改善(hr-hrlb)比率

vector<Zone> Target0,   //儲存狀態為 0 的網格，初始時全部網格皆存於此
    Target1,           //儲存狀態為 1 的網格，除 debug 外作用不大，或可不用到
    Target3,           //儲存狀態為 3 的網格，通常為射擊網格之相鄰網格
    Target4;           //儲存狀態為 4 的網格，因不易找到，可能也用不到
```

string dataname; //記錄輸入之測試檔名，譬如 data_4_6_1.txt

請使用者輸入檔名，譬如讀取 data_4_6_1.txt 檔案後，其格式如下：

```
dataname      M=4, N=6, BTN[]={2,2}
data 4 6 1.txt 4 6 2 2 2 3 2
1 1 1 0 1 0   n_BT=2, BTL[]={2,3},
0 0 0 0 1 0   長度為3的船：(0,0)~(0,2); (3,3)~(3,5)
1 1 0 0 0 0   長度為2的船：(0,0)~(2,1); (2,0)~(2,1)
0 0 0 1 1 1
```


讀檔完後，程式應詢問「Debug mode(1:yes)?」，輸入 1 代表 isDebug=1，亦即程式將會列印相關細節資料與各步驟，協助使用者驗證程式有否按照想法執行。

程式應該算出 n_targets=10，初始化 n_shots=n_hits=0，

上述之 4×6 的 0,1 陣列即為 SMap[][]；而 4×6 的 BMap[][] 先初始化為 0；

將所有的 24 個網格存入 Target0 這個 vector 內；

假設自 Target0 以 RS 策略隨機選取出(1,2), (2,1), (2,4), (3,1), (0,0), (0,3), (3,3), (1,4), (2,5), (0,5), (3,0), (2,0), (3,5), (1,0), (0,2), (0,4), (3,4), (1,5), (0,1)等 19 個網格，在 isDeg=1 時輸出：

```
A 4x6 map, 2 types, length={2,3}, number={2,2}, n_targets=10
Trial 1: zone 8=(1,2) missed
Trial 2: zone 13=(2,1) hit 1, 9 to go!!
Trial 3: zone 16=(2,4) missed
Trial 4: zone 19=(3,1) missed
Trial 5: zone 0=(0,0) hit 2, 8 to go!!
Trial 6: zone 3=(0,3) missed
Trial 7: zone 21=(3,3) hit 3, 7 to go!!
Trial 8: zone 10=(1,4) hit 4, 6 to go!!
Trial 9: zone 17=(2,5) missed
Trial 10: zone 5=(0,5) missed
Trial 11: zone 18=(3,0) missed
Trial 12: zone 12=(2,0) hit 5, 5 to go!!
Trial 13: zone 23=(3,5) hit 6, 4 to go!!
Trial 14: zone 6=(1,0) missed
Trial 15: zone 2=(0,2) hit 7, 3 to go!!
Trial 16: zone 4=(0,4) hit 8, 2 to go!!
Trial 17: zone 22=(3,4) hit 9, 1 to go!!
Trial 18: zone 11=(1,5) missed
Trial 19: zone 1=(0,1) hit 10, 0 to go!! DONE!!
Hitrate = 10/19 = 52.63%;
Hitrate lower bound = 10/24 = 41.67%
Hitrate Improvement Rate = 100.*(52.63-41.67)/(100.-41.67) = 23.14%
```

若輸入 isDebug=0，則程式將僅列印下列 9 個資料，分別代表學號、輸入檔名、選取機制、M*N、n_targets、n_shots、hr、hrlb、hrimprate，此輸出將被用於 PART3：

```
d95149514 data_4_6_1.txt RS 24 10 19 52.63 41.67 23.14
```

建議將 PART1 程式命名為 **HW41.cpp**，執行檔名為 HW41

!!請注意此部分必須於 2018/05/04(星期五)09:00 上傳程式!!

PART2[40+20%] 宣告、讀檔、IS 策略:[2018/05/14 09:00 上傳程式與說明報告(doc/pdf)]

此部分可直接接著 PART1 的前半部繼續寫，主要困難度在於「智慧」機制的設計與實作，此機制的設計邏輯說明除了簡短描述於 header 部分外，亦必須寫在簡短報告 HW423.doc 上（譬如以 Word 撰寫，佔 20%，請看 PART3 說明）。以下假設自 Target0 以 IS 策略選取出(0,1), (1,1), (0,0), (0,2), (2,4), (2,1), (2,2), (2,3), (2,0), (3,4), (3,3), (3,5), (0,5), (1,4), (0,4)等 15 個網格。

在讀入檔案後，程式應詢問「Debug mode(1:yes)?」，若輸入 1 代表 isDebug=1，亦即程式將會列印相關細節資料與各步驟，協助使用者驗證程式有否按照想法執行，其應輸出類似下述內容：

```
A 4x6 map, 2 types, length={2,3}, number={2,2}, n_targets=10
Trial 1: zone 1=(0,1) hit 1, 9 to go!!
Trial 2: zone 7=(1,1) missed
Trial 3: zone 0=(0,0) hit 2, 8 to go!!
Trial 4: zone 2=(0,2) hit 3, 7 to go!!
Trial 5: zone 16=(2,4) missed
Trial 6: zone 13=(2,1) hit 4, 6 to go!!
Trial 7: zone 14=(2,2) missed
Trial 8: zone 15=(2,3) missed
Trial 9: zone 12=(2,0) hit 5, 5 to go!!
Trial 10: zone 22=(3,4) hit 6, 4 to go!!
Trial 11: zone 21=(3,3) hit 7, 3 to go!!
Trial 12: zone 23=(3,5) hit 8, 2 to go!!
Trial 13: zone 5=(0,5) missed
Trial 14: zone 10=(1,4) hit 9, 1 to go!!
Trial 15: zone 4=(0,4) hit 10, 0 to go!! DONE!!
Hitrate = 10/15 = 66.67%
Hitrate lower bound = 10/24 = 41.67%
Hitrate Improvement Rate = 100.*(66.67-41.67)/(100.-41.67) = 42.86%
```

若輸入 isDebug=0，則程式將僅列印下列 9 個資料，分別代表學號、輸入檔名、選取機制、M*N、n_targets、n_shots、hr、hrlb、hrimprate，此輸出將被用於 PART3：

```
d95149514 data_4_6_1.txt IS 24 10 15 66.67 41.67 42.86
```

以上只是一個可能的作法，其中還有一些細節可參考 [HW4IS_eg1.pdf](#)。

Target0 在 PART1 中僅儲存尚未被射擊過的網格，然而在 PART2 中可以將這些網格再依推論進一步細分成 Target1、3、4，其中只有屬於 Target3、4 的網格才有用，因為 Target1 用於儲存推論而得的「海水」網格，旨在避免程式選取（其實只要各網格 struct 內的 status 在過程中有隨之更新即可，Target1 這個 vector 或可忽略不做）；Target4 較不易判斷，可能要到最後幾步才能確定，因此 Target4 這個 vector 亦可忽略不做；Target3 存取相對更有機會擊中的網格，通常可將每次射擊的網格相鄰且狀態為 0 之網格加入 Target3，因此 HW3 如何搜尋上下左右相鄰網格的方式在此就會派上用場。

簡言之，PART2 大概的機制為(1)射擊時依 Target4→3→0 的順序選取網格，絕對不選 Target1 的網格；(2)依據射擊結果將某些網格由 Target0 歸類至 Target1、3、4，或是由 Target3 歸類至 Target1、4。以上過程必須熟悉如何自 vector 中搜尋並刪除、新增元素等操作，其中「搜尋並刪除」必須自行寫 for/while loop 一一比對 vector 內的元素，找到其在 vector 內的 index 之後，再將之刪除。

建議將 PART2 程式命名為 **HW42.cpp**，執行檔名為 HW42；書面報告命名為 **HW423.doc**

!!請注意此部分必須於 2018/05/14 09:00 上傳程式與智慧機制之說明報告(doc/pdf)!!

PART3[20%] 大量測試 10000 組測資以分析 RS/IS 之平均效能：**[2018/05/14 09:00 上傳程式與說明報告(doc/pdf)]**

此部分旨在讓同學練習如何處理大量測資。若 PART1&2 有實作成功的話，可直接以 PART1&2 的程式來修改，主要在 isDebug 部分輸入 1。由於目前 PART1&2 的程式僅能一次處理一個資料檔，因此若欲處理 10000 筆資料，勢必使用一些進階技巧。在此，我們打算讓同學練習 batch script 的使用方式，亦即將每次測試單一資料檔的過程以文字檔 **HW43.bat** 記錄起來，重複同樣過程 10000 次（每次針對新的資料檔），將程式之輸出導出至一個新的文字檔 **HW43output.txt**。接著即可使用 Excel 打開 **HW43output.txt**，以空格(space)當成分隔方式，即可容易地整理兩個程式各自處理 10000 筆測資的結果。此部分更詳細的操作方式與步驟說明將在課堂（特別是實習課）教授。

以下是書面報告內容：

- (1) **[20%算在 PART2]** 先將 PART2 的「智慧機制」設計邏輯解釋清楚，必要時可畫圖說明
- (2) **[20%算在 PART3]** 同學們必須自行使用 Excel 畫出 HW41 與 HW42 各自產生 10000 組的 命中改善率(hrimprate) 分佈直方圖（將命中改善率小數點位數改為 0，製作其各自之直方圖 histogram，橫軸為命中改善率的數值，譬如 52.12%就是直接當成 52，縱軸為該命中改善率之次數），共兩個圖，同時應計算此 10000 組命中改善率之平均(mean)、中位數(medium)、標準差(stddev)。請同學再自行嘗試分析看這些測試結果中有沒有什麼特別需要注意之處，若有則請在報告中說明之。

解題建議

- 老師估計初學者至少花 10hr 才能將本題作好，特別是 PART2。為了讓 PART2 做得更順，本作業將拆成兩次繳交，PART1 期限為 2018/05/04（亦即公告後約 10 日），PART2&3 期限為 2018/05/14（Quiz2 當天，距 PART1 期限約 10 日）。請同學們務必趕緊處理，最好是可以在 2018/05/04 前已經寫到 PART2 至少一半了，不然還要準備考試還有其它科作業，另外 HW5 應該也會提早公告。
- PART1&2 在 vector 內的元素選取部分，應該要用隨機選取的方式，不過站在 debug 的立場，或許同學們可考慮在這個部分先以使用者輸入的選取方式（亦即非隨機），照著本作業說明的例子依序輸入範例選取的網格，看看程式是否會正常操作產生預期的結果，等到 ok 之後再將這個選取網格部分改成隨機選取（這裡當然要用到一些小技巧去做）。
- PART2 的智慧選取機制考驗各位的「智慧」與實作能力，我們鼓勵大家多嘗試想一些機制去避免選海水網格、更快擊中船格。可以預期的是，你的檢查機制越複雜，你的程式就會越長。在此老師不限制各位使用函式(function，尚未正式教到)去實作，不過請記得你應該要將你想的機制於書面報告 **HW423.doc** 解釋清楚。
- PART2 的 60%內其實有 20%（約 5-10%來自書面報告、10-15%來自 PART3 的命中率表現）為 extra credit，會由同學們在 PART3 中 10000 組測試結果的 HW42 表現來評分。我們可能會將所有同學在 HW42 的測試表現（可能會用助教另外準備新的 10000 組測資）整合起來 RANKING 一下，依效能表現來給這 20%。

***作業繳交應注意事項**

1. 作業需繳交電子檔以及書面(列印程式檔及程式結果)
2. 電子檔請於作業繳交截止時間以前上傳至 <http://moodle.ncku.edu.tw>
 - 2.1 請同學先建立一個資料夾，資料夾名稱為“學號_hw4”，例如學號為 hxxxxx，則資料夾名稱則為 **hxxxxx_hw4**
 - 2.2 將 PART1 程式檔案名稱存為 **HW41.cpp**，PART2 程式檔案名稱存為 **HW42.cpp**，PART3 的 batch 檔案 **HW43.bat**、輸出檔 **HW43output.txt** 與書面報告檔 **HW423.doc** (或 HW423.pdf)，並將以上 **5 個檔案** 存於上述設立之 **學號_hw4** 資料夾中
 - 2.3 最後將**整個學號_hw4** 資料夾壓縮成 zip 檔(**學號_hw4.zip**)，再上傳至 moodle 系統(！注意！：除上述檔案外，**請勿**將 .o, .exe, 及 10000 個測試檔上傳，**違者可能會被扣分**)
3. 此次作業將分兩次繳交，其中 PART1 必須在 **2018/05/04(星期五)09:00** 上傳**程式**，書面作業則於 **2018/05/07 09:15** 課堂繳交。
PART2&3 必須在 **2018/05/14 09:00** 上傳**程式**，書面作業則於當日 **09:15** 課堂繳交。
!!注意!!05/14 的 Quiz2 很可能會跟上次一樣型式，因此請同學們務必記得及時繳交