

# 資料結構實務

鍾健雄  
2021

## 本日課程

- 課程介紹
  - 授課教師、課程目標、課程規劃、成績計算
- 大學生程式能力檢定測驗 (CPE)
- 一點點演算法概念
- 瘋狂程設系統操作練習

## 授課教師

- 鍾健雄
- 美國維吉尼亞大學 (UVA)
  - 系統暨資訊工程博士 2003 研究興趣
    - Python程式應用、資料探勘、資料視覺化、空間大數據分析
    - 離散事件系統模擬、企業架構建模分析(UML)、ICS資訊安全
- 手機 0919341293
- 電子郵件 jc7qxccit@gmail.com

## 課程目標

- 提供同學資料結構實習
- 鍛鍊同學程式撰寫技巧
- 協助同學通過CPE測驗
- 提升同學編程解題能力

## 課程規劃

- 程式撰寫語言
  - C/C++
- 程式開發環境
  - CodeBlocks
- 上機考試系統
  - 瘋狂程設系統
- 授課方式
  - 課程講解、實作練習、課堂小考、課後作業

## 成績計算

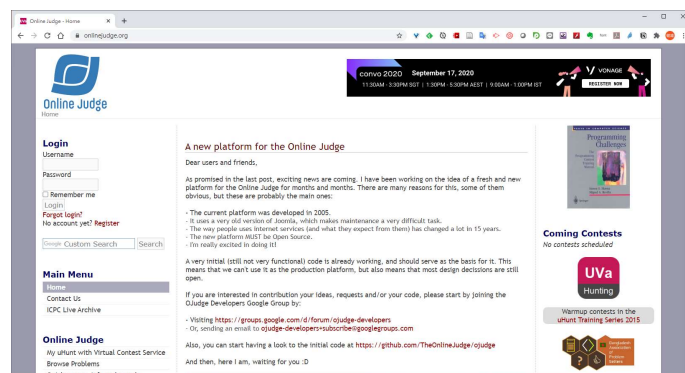
- 平時成績 80%
  - 課程參與 10
  - 平時小考 40
  - 課後作業 30
- 期中成績 20%
  - 模擬 CPE 考試
- 加分機制 (加分後總分超過100以100分計算)
  - 參加CPE測驗，測驗結果
    - 完成一題 + 10
    - 完成二題 + 30
    - 完成三題 + 50
    - 四題以上 平時作業及小考滿分

## 大學生程式能力檢定 (CPE測驗)

- **ACM ICPC ( International Collegiate Programming Contest · 國際大學程式競賽 )**
- **1970年美國**[Texas A&M University](#)**大學程式比賽**
- **1991年：**亞洲首支隊伍參加世界總決賽 - 國立交通大學。
- **1995年：**[台灣](#)首度舉辦[亞洲區域賽](#)

## UVA 線上評審 (<http://uva.onlinejudge.org/>)

- 線上即時評分系統 ( 電腦自動評分 )
- 題目來源：**ACM ICPC**
- 題目總數：超過**3600**題



## 題目格式

- 題目描述
- 輸入格式
- 輸出格式
- 輸入樣本
- 輸出樣本

Probability has always been an integrated part of computer algorithms. Where the deterministic algorithms have failed to solve a problem in short time, probabilistic algorithms have come to the rescue. In this problem we are not dealing with any probabilistic algorithm. We will just try to determine the winning probability of a certain player.

A game is played by throwing a dice like thing (it should not be assumed that it has six sides like an ordinary dice). If a certain event occurs when a player throws the dice (such as getting a 3, getting green side on top or whatever) he is declared the winner. There can be  $N$  such player. So the first player will throw the dice, then the second and at last the  $N$ -th player and again the first player and so on. When a player gets the desired event he or she is declared winner and playing stops. You will have to determine the winning probability of one (The  $I$ -th) of these players.

### Input

Input will contain an integer  $S$  ( $S \leq 1000$ ) at first, which indicates how many sets of inputs are there. The next  $S$  lines will contain  $S$  sets of inputs. Each line contain an integer  $N$  ( $N \leq 1000$ ) which denotes the number players, a floating point number  $p$  which indicates the probability of the happening of a successful event in a single throw (If success means getting 3 then  $p$  is the probability of getting 3 in a single throw. For a normal dice the probability of getting 3 is  $1/6$ ), and  $I$  ( $I \leq N$ ) the serial of the player whose winning probability is to be determined (Serial no varies from 1 to  $N$ ). You can assume that no invalid probability ( $p$ ) value will be given as input.

### Output

For each set of input, output in a single line the probability of the  $I$ -th player to win. The output floating point number will always have four digits after the decimal point as shown in the sample output.

### Sample Input

```
2
2 0.166666 1
2 0.166666 2
```

### Sample Output

```
0.5455
0.4545
```

## 題目難易程度分級

- **一顆星**：學習完計算機概論之後即可解答（專家級設計師大約可於**10分鐘**撰寫完畢）
- **兩顆星**：學習完資料結構之後才能解答或是苦工題（專家級設計師大約可於**10 ~ 30分鐘**撰寫完畢）
- **三顆星**：要有好的演算法或數學方法才能解答（專家級設計師大約可於**30 ~ 100分鐘**撰寫完畢）
- **四顆星**：要有特殊的演算法或是綜合多種演算法才能解答（專家級設計師需要超過**100分鐘**才能撰寫完畢）
- **五顆星**：超越四顆星的極特殊題目

## 國際計算機器協會程式競賽台灣協會

- **推動委員會**：負責資源與庶務之整合。
- **技術委員會**：由教練與命題老師組成，負責培訓與命題事務。
- **大學程式能力檢定委員會 ( Collegiate Programming Examination Committee, 簡稱CPE Committee )**：共同舉辦CPE程式檢定

## 大學程式能力檢定(CPE)

- 大學程式能力檢定 **CPE (Collegiate Programming Examination)**
- 線上程式設計、電腦自動評判，[採ACM ICPC排名方式](#)
- CPE用途：
  - 單一課程上機考試
  - 學系畢業檢定
  - [研究所入學考](#)、廠商徵才
  - 提升個人程式設計能力 ( 比賽之練習 )
- 網址：<http://cpe.cse.nsysu.edu.tw>



## CPE辦理方式

- 每年辦理四次，大約為每年的3、6、9、12月
- 考生若為大專學生，則是免費報名。
- 報名後，無故缺席而未到考，將取消其後一次考試資格。
- 電腦現場上機考試
- 考試時，封閉與考試無關之網路。考生除紙本字典外，不能攜帶任何資料。
- 題目來源：Uva題目庫

## CPE計分規則

- 絕對成績：A,B,C,F 等級距
  - A+：6題或6題以上
  - A：4題至5題
  - A-：3題
  - B：2題
  - C：1題

## CPE排名規則

- 與ACM ICPC排名規則相同
- CPE考試時間為3小時
- 每個題目結果只有「對」與「錯」
- 答對題數較多者，排名較前
- 答對題數相同者，以解題時間總和決定排名
- 解題時間為比賽開始至解題正確所花時間，再加上罰扣時間（每送出題解錯誤一次罰加20分鐘）
- 答錯的題目不計時間及罰扣時間
- 計分範例：甲生開賽後10分鐘答對A題，25分鐘送出B題（但錯誤），32分答對B題。總時間為 $10+32+20*1=62$ 分

## CPE程式設計規範

- 輸出與輸入都不是視窗介面
- 輸入與輸出均採取「標準輸入」(stdin)與標準輸出」(stdout)，不可使用檔案讀寫。撰寫程式時，於C語言，可使用如scanf與printf函式；於C++，可使用如cin與cout物件。
- 資料全為純文字資料，必須完全依照題目的輸入與輸出格式。**程式必須通過評判伺服器的測試資料(不公開)，才算「答對」。**
- **測試資料的格式一定按照題目所給予的輸入與輸出格式**，撰寫程式時，無需檢查格式是否正確。
- 所撰寫的程式必須有正確的副檔名，如filea.c或filea.cpp等，並且「選擇正確的語言」送繳程式。



## CPE評審伺服器回傳之訊息(1)

訊息	意義
COMPILER-ERROR	程式碼未通過編譯。(點入連結可以查閱編譯器所產生的錯誤訊息。)
<b>CORRECT</b>	程式已經正確，並通過測試。
NO-OUTPUT	程式沒有輸出任何資料。
PENDING	送出的程式碼仍在處理中。
PRESENTATION-ERROR	輸出的結果正確，但格式錯誤，例如未依規定空格或換行(多空格或少空格，多換行或少換行)。

## CPE評審伺服器回傳之訊息(2)

訊息	意義
RUN-ERROR	無法順利將程式執行完畢，亦即程式執行過程發生錯誤，例如記憶體存取錯誤。
TIMELIMIT	程式執行所花費的時間超過題目限制。程式可能落入無窮迴圈，或是必須改進解題方法。
WRONG-ANSWER	輸出的結果錯誤。(若輸出的格式產生過大錯誤，也可能造成此結果。)

## CPE解題想法

- 了解題意(英文出題，加強英文能力！利用輸入/出猜出提意)
- 挑選適當的演算法(含資料結構)
- 估計程式效率(時間、記憶體使用)
- 程式測試資料
- 善用既有資源

## 理解題意

- 關鍵字
  - **Given, Let's, Define, Solve...**
- 排版不同的部分
  - 圖片
  - 表格

## THE ONE-HANDED TYPIST (UVA10393)

Jimmy has a job of typing documents for ACM or he rather had a job until his unfortunate skiing accident. With some of his fingers in a cast, he is finding it difficult to type as he cannot press all the keys of a keyboard and hence his job is in danger.

Jimmy has come to you for help. He needs to prove to his boss that he can still type long words (his boss likes long words because it makes him look smart). **Given** a list of fingers (fingers are identified by integers) that cannot be used (due to his accident), and a list of words Jimmy can add to his boss's documents to make him look smart, **find** all the longest words that Jimmy can type.

Jimmy uses standard fingering, meaning he can type each of these letters with a finger. The list below shows a finger number and the characters it can press.

左手	keys	右手	keys
1 (小拇指)	q, a, z	6 (拇指)	space(空白键)
2 (無名指)	w, s, x	7 (無名指)	u, h, n, u, j, m
3 (中指)	e, d, c	8 (中指)	i, k
4 (食指)	r, f, v, t, g, b	9 (食指)	o, l
5 (拇指)	space(空白键)	10 (小拇指)	p, ;, /

## 解法：找出某些字母不出現的最長單字

1. 讀取單字列表中一個未處理的單字。
2. 若單字中包含禁止的字母，則回到1讀取下一個字。
3. 如果單字長度比目前最長的單字長，就清除答案陣列，然後放入此單字。
4. 如果一樣長，則放入答案的陣列中。
5. 如果還有單字還沒處理，回到1。
6. 輸出答案陣列。

5個手指不能動

1. 5 5 ↖ ↗ 老闆要求5個詞彙
2. 6 7 8 9 10 ← 不能動的手指頭
3. ~~the~~
4. stewardesses
5. ~~have~~
6. ~~funny~~
7. ~~hair~~

左手	keys	右手	keys
1 (小拇指)	q, a, z	6 (拇指)	space(空白鍵)
2 (無名指)	w, s, x	7 (無名指)	u, h, n, u, j, m
3 (中指)	e, d, c	8 (中指)	i, k
4 (食指)	r, f, v, t, g, b	9 (食指)	o, l
5 (拇指)	space(空白鍵)	10 (小拇指)	p, ;, /

找出Jimmy可以完成type的最長詞彙，  
如長度相同以字母順序排列

## 挑選適合演算法

- 排序、搜尋
  - 題目出現 排順序、找出特定的值
- 貪婪演算法 (Greedy Algorithm)
  - 思考在不選當下最佳時，是否有最佳解
- 動態規劃 (Dynamic Programming)
  - 可由先前的結果推導出後面結果
- 圖形走訪 (Graph Traversal)
  - 題目的狀態可想像成圖中的節點
  - 找出最短完成法
- 最小生成樹 (Minimum Spanning Tree)
  - 題目出現 找出連接所有點的最短長度
- 最短路徑 (Shortest Path)
  - 題目抽象化
  - 多節點、節點間連線有權重，找出兩點間的最小權重和
- 最大流 (Maximum Flow)
  - 最複雜，沒有固定的形式
  - 將問題轉化為節點、邊

## 估計程式效率

- 估計記憶體用量
- 估計時間
- 從限制條件猜測解法

## 估計記憶體用量

- **Global** ( 全域 )
  - 全域變數
- **Stack** ( 堆疊 )
  - 區域變數、函式參數、函式返回位置
- **Heap**
  - 動態產生的資料

## 估計時間

- 時間複雜度
  - $n$ 個迴圈 =  $O(n)$
- 時間函式庫
  - `difftime()`
- 範例
  - ```
for (int i =0; i < m; i++){  
    for(int j=0; j < n; j++){ ... }  
}
```
  - 時間複雜度 =  $O(mn)$ 。

## 從限制條件猜測解法

- 時間限制過長 ( 超過一分鐘 )
  - 暴力解
  - 尋寶問題 ( 2009年高中能力競賽全國賽 )
- 時間限制很短 ( 0.1秒 )
  - 公式解
  - UVA438 : The Circumference of the Circle

## 程式測試資料

- 測試資料範圍
- 軟體測試
- 邊界測試 (Boundary Test)
- 產生測試資料

## 測試資料範圍

- 測資範圍
  - 沒有表明不會出現，只要符合此規則就有可能會出現。
- 範例
  - 10400 : The  $3n+1$  problem
  - 10460 : You can say 11

## 10460: YOU CAN SAY 11(UVA10929)

- 題目 : <http://uva.onlinejudge.org/external/109/10929.html>
- 題意 : 判斷輸入的整數是否為 11 的倍數。
- 題意範例 :
 

輸入112233，顯示112233 is a multiple of 11.

輸入112234，顯示112234 is not a multiple of 11.
- 解法 : 整數中奇數位的和與偶數位的和，兩者相減之差若為11之倍數，則此數為11的倍數；若相減之差不是11的倍數，則不是11的倍數。
- 討論 : 此題須注意的是輸入的數字可能由0開頭，如0011，所以要對此作適當的處理。



## 軟體測試

- 黑箱測試 (Black Box)
  - 檢查程式的輸出結果是否正確
  - 輸入多筆測資，檢查輸出是否符合對應的答案
- 白箱測試 (White Box)
  - 觀察程式運作過程是否正確
  - **gdb**
  - 於程式中插入 `printf()`

## 邊界測試 (Boundary Test)

- 測資邊界值
  - 測資定義的最大、最小值
- 變數型態範圍邊界值
  - **4 bytes int** 能表示的範圍
    - **-2147483648~2147483647**
- 判斷條件邊界值
  - `if(x>0) ...`
  - `else if(x<0) ...`
  - 忽略 `x = 0` 的條件處理，可能造成錯誤

## 善用既有資源

- **Don't Repeat Yourself**
  - 不要寫出重複的程式
  - **10414 : Bangla Numbers(UVA 10101)**
- **C/C++ 語言函式庫**
  - **vector**、**iterator** 使用範例
  - **map**、**set** 使用範例
- **Sort 函式庫**
  - **11069: Sort! Sort!! and Sort!!! (UVA11321)**