

程式能力 檢定

資工13 姜翰廷

OUTLINE

- 1. 程式能力檢定簡介
- 2. 檢定的模式
- 3. 題目形式
 - 暴力法
 - 數學
 - 資料結構和演算法的應用
- 4. 如何準備檢定考試
 - 資料結構和演算法
 - 程式競賽訓練
 - 其他練習網站



程式能力檢定簡介

- 本學年度開始新生皆需通過「程式能力檢定」通過方可畢業
- 目的是要提升整體學生的程式實作能力
- 增加學生畢業後工作的競爭力
- 培養分析並解決問題的能力



檢定的模式

- 一人一機
- 時間三小時
- 給予大約10個題目
- 程式只有「全對or全錯」
- 寫對一定以上的題數才算通過



ONLINE JUDGE AND SCORE BOARD

#	AFFIL.	TEAM	SCORE		10404	10566 ●	10655 ●	10708 ●	11067 ●	11174 ●
1	交大 資訊工程系	廖 挺富	9	650	1 (3 + 0)	1 (23 + 0)	1 (113 + 0)	problem 'Sumsets' + 0	+ 0	1 (61 + 0)
2	交通大學 資科工碩	康 皓華	7	588	1 (0 + 0)	2 (150 + 20)	1 (32 + 0)	0	1 (10 + 0)	2 (163 + 20)
3	中央大學 資訊工程學系	Jim Chang	6	474	1 (8 + 0)	3 (70 + 40)	2 (93 + 20)	0	1 (7 + 0)	1 (35 + 0)
4	交大 資工系資電組103	王 傑民	6	476	1 (28 + 0)	1 (27 + 0)	2 (15 + 20)	0	1 (80 + 0)	4
5	清大 資訊工程學系13級	姜 翰廷	6	523	1 (0 + 0)	3 (33 + 40)	3 (62 + 40)	0	3 (16 + 40)	2 (82 + 20)
6	成大 資訊工程學系	葉 昇翰	5	514	1 (2 + 0)	1 (59 + 0)	5	0	1 (96 + 0)	2 (172 + 20)
7	中山 CSE	Liu Allen	5	578	1 (0 + 0)	3 (59 + 40)	2 (62 + 20)	8 (175 + 140)	0	0
8	成大 資工100	韓 政廷	5	580	4 (49 + 60)	3 (21 + 40)	3	0	2 (60 + 20)	7 (125 + 120)
9	成大 資工系100級	彭 成玉	5	643	1 (8 + 0)	1 (89 + 0)	0	0	3 (169 + 40)	4 (124 + 60)
10	中山 CSE102	陳 慶耀	4	286	1 (1 + 0)	2 (73 + 20)	4	0	1 (22 + 0)	0
11	成功大學 資訊工程系所	Li Kuan-Hsien	4	296	2 (18 + 20)	1 (64 + 0)	0	0	1 (37 + 0)	2 (137 + 20)
12	清華大學 資訊工程學系	Kuo Chun-Ting	4	329	1 (19 + 0)	1 (15 + 0)	0	3	0	1
13	成功大學 資訊工程學系	Huang pin-chieh	4	353	3 (11 + 40)	1 (75 + 0)	3 (123 + 40)	0	2 (44 + 20)	0
14	交大 資訊工程學系	施 詠翔	4	397	4 (36 + 60)	1 (50 + 0)	5	0	2 (54 + 20)	1

10404:PRIMARY ARITHMETIC

- Time Limit: 3 sec
- **Description**
- Children are taught to add multi-digit numbers from right-to-left one digit at a time. Many find the "carry" operation - in which a 1 is carried from one digit position to be added to the next - to be a significant challenge. Your job is to count the number of carry operations for each of a set of addition problems so that educators may assess their difficulty.
- **Input**
- Each line of input contains two unsigned integers less than 10 digits. The last line of input contains 0 0.
- **Output**
- For each line of input except the last you should compute and print the number of carry operations that would result from adding the two numbers, in the format shown below.



10404: PRIMARY ARITHMETIC

○ Sample Input

123 456

555 555

123 594

0 0

○ Sample Output

No carry operation.

3 carry operations.

1 carry operation.



10404:PRIMARY ARITHMETIC

- 題目大意：給你兩個正整數 a, b ，請輸出 $a+b$ 總共會有幾次的進位
- 例如：555 和 555總共有3次的進位，個位數的5相加為10產生一次進位，十位數的5和進位相加為11也產生一次進位，百位數的5和進位相加為11產生第三次的進位。



10404:PRIMARY ARITHMETIC

- **Solution**：利用程式模擬直式加法的過程，並一邊記有沒有進位。



10305:ORDERING TASKS

- **Description**

- John has n tasks to do. Unfortunately, the tasks are not independent and the execution of one task is only possible if other tasks have already been executed.

- **Input**

- The input will consist of several instances of the problem. Each instance begins with a line containing two integers, $1 \leq n \leq 100$ and m . n is the number of tasks (numbered from 1 to n) and m is the number of direct precedence relations between tasks. After this, there will be m lines with two integers i and j , representing the fact that task i must be executed before task j . An instance with $n = m = 0$ will finish the input.



10305:ORDERING TASKS

- **Output**

- For each instance, print a line with **n** integers representing the tasks in a possible order of execution.

- **Sample Input**

- 5 4

- 1 2

- 2 3

- 1 3

- 1 5

- 0 0

- **Sample Output**

- 1 4 2 5 3



10305:ORDERING TASKS

- 題目大意：John有 n 件事情要做，不幸的是這些事情並不是各自獨立的，而是有相依性的。換句話說可能有某件事情一定要在另一件事情做完之後才能做。



10305:ORDERING TASKS

- **Solution:** 利用一種叫做topological sort的方法，做廣度優先搜尋後，以離開的節點的時間做為標記，最後在反過來輸出即可得到解。



4138 - ANTI BRUTE FORCE LOCK

- Lately, there is one serious problem with Panda Land Safe Box: several safes have been robbed! The safes are using old 4-digits rolling lock combination (you only have to roll the digit, either up or down, until all four of them match the key). Each digit is designed to roll from 0 to 9. Rolling-up at 9 will make the digit become 0, and rolling-down at 0 will make the digit become 9. Since there are only 10000 possible keys, 0000 through 9999, anyone can try all the combinations until the safe is unlocked.



4138 - ANTI BRUTE FORCE LOCK

- What's done is done. But in order to slow down future robbers' attack, Panda Security Agency (PSA) has devised a new safer lock with multiple keys. Instead of using only one key combination as the key, the lock now can have up to N keys which has to be all unlocked before the safe can be opened. These locks works as the following:
- Initially the digits are at 0000.
- Keys can be unlocked in any order, by setting the digits in the lock to match the desired key and then pressing the UNLOCK button.
- A magic JUMP button can turn the digits into any of the unlocked keys without doing any rolling.
- The safe will be unlocked if and only if all the keys are unlocked in a minimum total amount of rolling, excluding JUMP (yes, this feature is the coolest one).
- If the number of rolling is exceeded, then the digits will be reset to 0000 and all the keys will be locked again. In other word, the state of the lock will be reset the cracking is failed.
- PSA is quite confident that this new system will slow down the cracking, giving them enough time to identify and catch the robbers. In order to determine the minimum number of rolling needed, PSA wants you to write a program. Given all the keys, calculate the minimum number of rolls needed to unlock the safe.



4138 - ANTI BRUTE FORCE LOCK

- **Input**

- The first line of input contains an integer T , the number of test cases follow. Each case begins with an integer N ($1 \leq N \leq 500$), the number of keys. The next N lines, each contains exactly four digits number (leading zero allowed) representing the keys to be unlocked.

- **Output**

- For each case, print in a single line the minimum number of rolls needed to unlock all the keys.

- Explanation for the 2nd case:

- Turn 0000 into 1111, rolls: 4
- Turn 1111 into 1155, rolls: 8
- Jump 1155 into 1111, we can do this because 1111 has been unlocked before.
- Turn 1111 into 5511 rolls: 8
- Total rolls = $4 + 8 + 8 = 20$



4138 - ANTI BRUTE FORCE LOCK

- **Sample Input**

- 4
- 2 1155 2211
- 3 1111 1155 5511
- 3 1234 5678 9090
- 4 2145 0213 9113 8113

- **Sample Output**

- 16
- 20
- 26
- 17



4138 - ANTI BRUTE FORCE LOCK

- **Solution:** 將這些狀態轉為節點，每一個節點間的距離可設為密碼相差的距離，最後使用最小生成樹的演算法就可以找到解答



如何準備程式能力檢定

- 熟悉基本的 C 語言語法
 - 像是基本input/output、if、for、while、function等都要熟練
 - 各種資料型態如int, long long ,double ,float等
- 基本的演算法與資料結構
- 思考問題的思維
- 將解法實現成為可正確執行之程式
 - 多練習！



資料結構與演算法

- 資料結構（data structure）是電腦中儲存、組織資料的方式。
- 演算法（Algorithm）是指完成一個任務所需要的具體步驟和方法。也就是說給定初始狀態或輸入資料，能夠得出所要求或期望的終止狀態或輸出資料。



程式競賽訓練

- 這是一門下個學期系上新開的課程
- 3學分
- 分成基礎組和進階組，評分方式和作業會不同，開學後會視實際參加的人來分組
- 上課時間也是由參加的學生和教授及助教討論
- 僅受理加簽
- 基礎組適合完全沒有程式競賽經驗並想嘗試參加的人或者是希望能增強自己的程式實作能力的人
- 進階組適合有接觸比較深入的一些資料結構與演算法的人，或自認為程式能力很強的人



其他練習的網站

- UVa online judge
 - 世界上最有名的線上評測網站
 - <http://uva.onlinejudge.org>
- ZeroJudge
 - 主要是高中生在練習的網站
 - <http://zerojudge.tw>
- TIOJ
 - 一些有熱情的高中生和大學生所架設的練習網站
 - <http://tioj.redirectme.net:8080/JudgeOnline/>
- 線上協同教學平台：
 - 教育部自己在推的網站
 - <http://140.118.170.42/e-Tutor/>

