

2016 ACM-ICPC Asia Taiwan Online Programming Contest Solution Sketch

Li-Cheng Lan, Jyun-Jie Liao, Pin-Chang Pan, Min-Zheng Shieh, Shang-Hsin Yu

本次題目保證 Python 3 可以解出的題目有：A、B、C、D、F、H。詳見 BitBucket 上的資料。

Problem A

此題是個模擬題。如何好好處理輸入是最大難點。將括號、逗號與空白跳過，除了自己寫程式碼之外，可以利用 C 語言的 `strtok`、Java 的 `StringTokenizer` 來處理。如果是針對這題，也可以使用 `scanf` 直接處理輸入。

Problem B

可以窮舉所有可能的點對應方法，總共只有六種。判斷所有可能性都不相等的話，就輸出no，否則輸出yes。

另解：判斷兩個圖是否邊的數量相同。是的話就yes，否則就no。仔細觀察後會發現所有簡單圖形都是三角形去掉若干邊。而去除邊數量相同，圖形仍舊是同構的。此法會較快寫出。

Problem C

此遊戲 Cook 贏 Levin 就輸、Levin 贏 Cook 就輸。因此，Cook 一開始能贏，必然是選了一手讓 Levin 怎麼玩都輸，反之 Cook 輸定的局面是，不管怎樣做 Levin 都能有一手會獲勝。觀察遊戲過程所經過的子盤面，其實可以由每個變數「被設定為真、被設定為假、尚未被設定」來決定，很多決策過程中會碰到重複的子盤面，至多只有 3^{10} 盤面，因此可以使用 Dynamic programming 來解決此題。此題真正麻煩之處，在於有效率的計算出 Boolean expression 的值。C/C++/Java 得靠自己撰寫程式來處理 expression evaluation，而 Python 可以直接由內建的 `eval()` 來計算 Boolean expression。只要搭配足夠有效率的 Dynamic Programming 便能解。而 C/C++/Java 則可以相對沒效率的演算法，搭配較快的 expression evaluation 解題。

Problem D

如果有一個if-goto 是往後跳的，可以保證換成do-while 無法編譯，因為do-while 必然是往前跳的。假定輸入中沒有向後跳的，所有的if-goto 都會回到一個前方的line_x 的標籤。我們將第 b 行上跳回第 a 行的if-goto 以區間 $[a, b]$ 代表。如果今天有兩個區間 $[a_1, b_1]$ 、 $[a_2, b_2]$ 滿足 $a_1 < a_2 < b_1 < b_2$ ，那這個程式碼將無法轉換成do-while 形式，因為原先位於 b_1 行，該

跳去第 a_1 行的 if-goto，將會跳到第 a_2 行，造成輸出不一致的結果。而其他情形都不會影響到。因此我們只需要檢查是否有此類狀況會發生即可。

實作上可以將區間依據起點排序，起點相同時依據終點排序，利用一個 Stack 來儲存區間，如果 Stack 是空的，或是新進的一個區間完全被 Stack 頂端的區間所覆蓋，就將該區間放到 Stack 頂端。如果堆疊頂端的區間 $[a_1, b_1]$ 跟新進區間 $[a_2, b_2]$ 滿足 $a_1 < a_2 < b_1 < b_2$ ，則直接停止，輸出 bad。否則，我們將 Stack 頂端的區間丟棄，直到上述兩種狀況其中一個成立為止。

如此實作時間複雜度 $O(n \log n) + O(n)$ 。即使使用 Python 也會通過。可以作到 $O(n)$ ，留給讀者想像。

Problem E

首先觀察能夠使用的道路，連接所有目的地之後會形成一個樹形圖，而在樹形圖上做旅行推銷員問題，答案會是邊長總和的兩倍。先觀察所有的葉節點（只有一條邊）要拜訪與離開，都需要經過連結該點的邊。接下來把原本連結葉節點的邊移除，新產生的葉節點也能套用此觀察，反覆為之，就可以得知答案至少是邊長總和兩倍。而任何一棵數，可以對起點進行一個深度優先搜尋 (Depth-First Search) 方式的遍歷，這剛好會拜訪過所有節點，並回到起點，且每條邊只經過兩次，與下限相同，答案就是邊長總和兩倍。

剩餘的工作就是找出如何造出邊長總和最小的樹形圖，即是一個最小生成樹問題，請搜尋「曼哈頓距離」與「最小生成樹」，取得剩餘的所需資訊。

Problem F

先紀錄縫合怪所經過的格子邊界，接下來就能造出一個最多 1025×1025 個格子的圖。接下來套用標準的 Flood fill 演算法算出有多少區域，或是用互斥集 (Disjoint sets) 直接將相鄰的兩格做聯集 (Union)，最後統計有多少集合即可算出答案。此類方法以 C/C++/Java 實作應該會通過。

另解：可觀察由縫合怪膽汁構造出來的圖是一個連通 (Connected) 的平面圖 (Planar Graph)。因此可套用歐拉公式 $F = 2 - V + E$ ： F 為區域數、 V 為點數、 E 為邊數。只要在圖形建立完成的當下，由邊數與點數直接求出答案。此法即 Python 3 的 Sample code 所使用的方法。

Problem G

由於單字長度最多只有 8，因此可以先列舉開頭一半以及結尾一半可以形成的餘數，並統計各種有多少個。接下來，對每個開頭一半形成的餘數 a ，去檢查結尾一半，是否有 b 滿足 $a \times b \equiv r \pmod p$ ，如果有，就紀錄 a 與 b ，以及其對應的字串。如果發現超過一個，那就 ambiguous，找不到就 not a word。

檢查 $a \times b \equiv r \pmod p$ 是否成立時，要留意 a 未必有乘法反元素。即 $r = 0$ 時要特例處理。

Problem H

簽到題。按照題目說明照做。

Problem I

待命題者提供。