

計算機程式設計二

Week 10 作業講解

2171 - I2P(II)2020_Chen_week10_HW

<https://acm.cs.nthu.edu.tw/contest/2171/>

賴御誠 編著

Overview

- 12155 - Cat-Toast Crisis
- 12521 - Break my heart
- 12522 - Thanos' Return

12155 - Cat-Toast Crisis

難易度：★★★☆☆

先備知識：深度優先搜尋法(DFS)、C Structure

Description

- 科學家發現，如果兩個貓吐司距離小於 r 時，就會發生碰撞產生黑洞並消失不見
- 另外，如果多個貓吐司都會發生碰撞的話，他們都會碰撞再一起並且只產生一個黑洞
- 有一天在清大的一個工讀生因為摔倒不小心把放在盒子裡的貓吐司摔了出來並發生反應，他希望能夠知道還剩下多少貓吐司留著，以及產生了多少個黑洞
- 計算貓吐司的距離公式： $\text{sqrt}((x1-x2)^2 + (y1-y2)^2)$

Input/Output

- 第一行包含 n (貓吐司數量) 與 r (距離)
- 接下來數行則是每個貓吐司的座標值 x_i, y_i
- $1 \leq n \leq 1000$
- $1 \leq r \leq 10^4$
- $1 \leq x_i, y_i \leq 10^4$
- 輸出剩餘的貓吐司與產生之黑洞數量，以空格格開，要換行

Input/Output

- 6 4

- 1 7

- 1 3

- 5 3

- 5 15

- 1 15

- 500 500

- 1 2

Input/Output

- 以 Sample IO 來說，
- $(1\ 7), (1\ 3), (5\ 3), (5\ 15), (1\ 15), (500\ 500) \rightarrow \text{黑洞}, \text{黑洞}, (500\ 500)$
- 因此會產生兩個黑洞與剩下一個貓吐司

12521 - Break my heart

難易度：★★★★☆

先備知識：Linked List(C)、Sets(C++)

Description

- 維護一個包含 integer 的序列，並且會有 n 個指令，包含：
- insert：插入一個 a_i 到序列裡面，若已經存在則不做任何動作
- print：遞增順序印出序列的數字
- min：印最小的數字，若沒有數字則不做任何動作
- range_erase：清除介於 $l \leq a_i \leq r$ 的所有元素 a_i
- 你可以使用 `std::set` 來解這個問題，並且可能需要 `set.begin()`, `set.size()`, `set.erase()`, `set.lower_bound()`, `set.upper_bound()`, `set.insert()`，以及 `iterator` 或 `auto(C++11)`
- 也可以使用 C 來解決這個問題

Input/Output

- 第一行為 n 個指令數量 ($1 \leq n \leq 5000$)
- 接著會有指令動作
- a_i ($0 \leq a_i \leq 10^9$)
- 對於 print 指令遞增輸出數字，並且使用空白隔開(最後一個沒有空白)
- 每行要換行
- 對於 mini 指令輸出最小數字，每行要換行

Input/Output

- 11
- print
- range_erase 3 53
- min
- insert 90
- insert 36
- min
- insert 1
- print
- insert 26
- range_erase 7 57
- print

- 36
- 1 36 90
- 1 90

12522 - Thanos' Return

難易度：★★★★☆

先備知識：Class、Operator Override、Friend

Description

- 給定 R 與 C 為矩陣 M 的列與行
- 運算符號 + 與 * 還有運算次數會被給定
- 對於 + 來說需要計算：

$$\sum_{i=1}^T M = \underbrace{M + M + \cdots + M}_T$$

- 對於 * 來說需要計算：

$$\prod_{i=1}^T M = \underbrace{M \times M \times \cdots \times M}_T$$

Description

- 這個問題是 Partial Judge ，因此部分函式已經完成，包含：
- Matrix()(constructor): 產生空的矩陣物件
- const int &getrow(): 取得物件的列數
- const int &getcol(): 取得物件的行數
- const int *operator[] (const int &x) const: 定義新的運算符號，讓 mat[a][b] 可以被直接檢視而不需要其他操作
- void print(): 印出整個矩陣

Description

- 你需要完成以下函式：
- `Matrix(int r, int c)(constructor)`: 產生一個列 `r` 個行 `c` 個的矩陣物件，並將元素值設為 0
- `int *operator[] (const int &x)`: 定義新的運算符號，讓 `mat[a][b]` 可以被直接存取而不需要其他操作
- `Matrix operator+ (const Matrix &x) const`: 執行加法運算並輸出結果
- `friend Matrix operator* (const Matrix &x, const Matrix &y)`: 執行乘法運算並輸出結果，注意這個函式被定義為 `friend`

Description

- 為了避免數字過大，所有數字都需要用 10007 取餘數
- 同時，取餘數後必須要是正數，比如： $-2395 \% 10007 = 7612$

Input/Output

- 第一行包含 R 、 C 、 T 、 OP
- 接著會有 R 行，每行有 C 個數字，共 $R \times C$ 個數字代表整個矩陣的所有元素
- $1 \leq R, C \leq 100$
- $0 \leq T \leq 10^{18}$
- $-10^5 \leq M \leq 10^5$
- 輸出運算結果，輸出格式請參閱 `function.h` 的 `print` 函式

Input/Output

- 3 3 10 *

- 7 5 3

- 8 5 1

- 6 2 4

- / 6849 668 3180 \

- | 7322 184 5769 |

- \ 5945 2776 948 /