# 程式競賽導論 112-2 普台競程讀書會

陳柏安

2023-03-17

■ 關於我

② 課程說明

③ 競賽程式

4 時間估算

陳柏安 程式競賽導論 2023-03-17 2

# 關於我

### 關於我

- 高三庚班 陳柏安
- 本學期的讀書會負責人兼講師
- 2022、2023 中投區學科能力競賽 佳作
- 被耽誤的競程選手
- 常用的 handle: gary\_cba
- mail: garychan60122gmail.com
- 我弱
- 想看更多請到: https://hackmd.io/@garycba/ry7DegY8j

### 我高中三年的故事

■ 太弱了

### 我高中三年的故事

- 太弱了
- ■以下空白

■ 112 特選生清大資工 Koying

- 112 特選生清大資工 Koying
- 高一才開始學程式 C++

- 112 特選生清大資工 Koying
- 高一才開始學程式 C++
- 付出許多時間在競程與社群貢獻

- 112 特選生清大資工 Koying
- 高一才開始學程式 C++
- 付出許多時間在競程與社群貢獻
- APCS 55、全國學科能力競賽三等獎...

6 / 42

陳柏安 程式競賽導論 2023-03-17

- 112 特選生清大資工 Koying
- 高一才開始學程式 C + +
- 付出許多時間在競程與社群貢獻
- APCS 55、全國學科能力競賽三等獎...
- 別人成果的背後是...

6 / 42

- 112 特選生清大資工 Koying
- 高一才開始學程式 C + +
- 付出許多時間在競程與社群貢獻
- APCS 55、全國學科能力競賽三等獎...
- 別人成果的背後是...
- 他的經歷 https://github.com/Koyingtw/spec-share

# 課程說明

■推廣競程

- ■推廣競程
- 希望能夠幫大家開闢一個新的道路

- 推廣競程
- 希望能夠幫大家開闢一個新的道路
- 課業固然重要,但興趣才是我們學習下去的動力,同時我也希望讓 大家的興趣成為大家最大的助力

- 推廣競程
- 希望能夠幫大家開闢一個新的道路
- 課業固然重要,但興趣才是我們學習下去的動力,同時我也希望讓 大家的興趣成為大家最大的助力
- 我覺得這才是競程的精神

■ 希望這樣的精神能傳承下去

- 希望這樣的精神能傳承下去
- 一起加油!

### 讀書會進行方式

- 每周一、非淨日 19:00 到 22:00
- 19:00 到 21:00 上課
- 21:00 到 22:00 提問與練習
- 三位講師 我、陳映杰、楊宗勳,分配擔任各堂課的講師
- 每節課會有演算法概念講解 -> 帶實際題目-> 課後習題練習

### 讀書會規則

#### 1. 出席

- 讀書會於每星期一、非淨日的晚間 19:00 22:00
- 學員應盡量準時出席,若有遲到或缺席,應事先請假

陳柏安 2023-03-17 11/42

### 讀書會規則

#### 1. 出席

- 讀書會於每星期一、非淨日的晚間 19:00 22:00
- 學員應盡量準時出席,若有遲到或缺席,應事先請假

#### 2. 參與

- 讀書會採用講師授課的方式進行,課堂中請尊重講師,請勿坐自己 的事,若屢勸不聽將會強制控制電腦
- 課堂中歡迎多多提問,能把講師問倒最好

11 / 42

### 讀書會規則

#### 1. 出席

- 讀書會於每星期一、非淨日的晚間 19:00 22:00
- 學員應盡量準時出席,若有遲到或缺席,應事先請假

#### 2. 參與

- 讀書會採用講師授課的方式進行,課堂中請尊重講師,請勿坐自己 的事,若屢勸不聽將會強制控制電腦
- 課堂中歡迎多多提問,能把講師問倒最好

#### 3. 學習

- 課後請利用時間盡量將該堂課所給的題目寫完
- 若遇到不會的題目,可以留到下次上課提出討論

### 課程目標

- 精熟基礎演算法與資料結構
- APCS 4 級分以上
- 其他的就靠實力累積吧

### 課程相關訊息

- 讀書會主頁 https://hackmd.io/@putailNF/rk2MpUvip
- 課程講義 https://hackmd.io/@putailNF/rkKZaQQDp
- 課程題單 https://hackmd.io/@putaiINF/r1daKnNBn

### 課程內容

■七大主題

### 課程內容

- 七大主題
- 基礎資料結構 Basic Data Structure
- 進階資料結構 Advanced Data Structure
- 枚舉 Enumerate
- 貪心 Greedy
- 基礎圖論 Basic Graph
- 進階圖論 Advanced Graph
- 分治 Divid and Conquar
- 動態規劃 Dynamic Programming

### 課前準備

- 加入 PTITC Discord 社群 https://discord.gg/AKK8N4EE2t
- 建好自己的環境
- 帶著清楚的頭腦來上課



### 競賽程式

■ 又稱為演算法競賽

- 又稱為演算法競賽
- 我們常常將競賽程式講成競程

- 又稱為演算法競賽
- 我們常常將競賽程式講成競程
- 主要是在比誰能在時間內解出最多、最有效率的程式

- 又稱為演算法競賽
- 我們常常將競賽程式講成競程
- 主要是在比誰能在時間內解出最多、最有效率的程式
- 題目內容涉及資料結構和演算法的運用

- 又稱為演算法競賽
- 我們常常將競賽程式講成競程
- 主要是在比誰能在時間內解出最多、最有效率的程式
- 題目內容涉及資料結構和演算法的運用
- 輸入 -> 程式運算 -> 輸出

陳柏安 2023-03-17 17/4

### 演算法和資料結構又是什麼?

■ 程式語言 (C++, Python, Java...) 為工具

### 演算法和資料結構又是什麼?

程式設計 = 演算法 + 資料結構

- 程式語言 (C++, Python, Java...) 為工具
- 資料結構資料儲存方式及架構

陳柏安 2023-03-17 18 / 42

## 演算法和資料結構又是什麼?

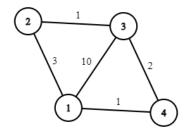
程式設計 = 演算法 + 資料結構

- 程式語言 (C++, Python, Java...) 為工具
- 資料結構資料儲存方式及架構
- 演算法是有效率解決問題的方法

陳柏安 程式競賽導論 2023-03-17 18/42

# 太抽象? 舉個例子

- Google 導航尋找最短的路徑
- 要排序一堆亂數



# 有哪些競賽?

- 國際資訊奧林匹亞 IOI & 台灣資訊奧林匹亞 TOI
- 資訊學科能力競賽 區賽 & 全國
- 網際網路程式設計全國競賽 NPSC
- 大學程式能力先修檢測 APCS
- 少年圖林計畫 YTP
- 成功大學高中生程式設計邀請賽 NCKU

陳柏安 程式競賽導論 2023-03-17 20/42

# 有哪些活動可以參加?

### 1. 課程

- 資訊之芽 (語法班 & 算法班)
- 普台程式設計培訓

### 2. 營隊

- APCS camp
- IOI camp
- ION camp

- 1. 特殊選材
  - 不用考學測

陳柏安 程式競賽導論 2023-03-17 22 / 42

### 1. 特殊選材

- 不用考學測
- 要有出色的競賽成績,或是你很特殊

陳柏安 程式競賽導論 2023-03-17 22 / 42

- 不用考學測
- 要有出色的競賽成績,或是你很特殊
- 在程式上要付出很多心力

- 不用考學測
- 要有出色的競賽成績,或是你很特殊
- 在程式上要付出很多心力
- 這條路不比學測輕鬆,越早決定要特選越好

- 不用考學測
- 要有出色的競賽成績,或是你很特殊
- 在程式上要付出很多心力
- 這條路不比學測輕鬆,越早決定要特選越好
- 2. 學測-APCS 組
  - 還是要準備學測,標準相對的比較低

- 不用考學測
- 要有出色的競賽成績,或是你很特殊
- 在程式上要付出很多心力
- 這條路不比學測輕鬆,越早決定要特選越好
- 2. 學測-APCS 組
  - 還是要準備學測,標準相對的比較低
  - 可以說是第二次的特選

#### 1. 特殊選材

- 不用考學測
- 要有出色的競賽成績,或是你很特殊
- 在程式上要付出很多心力
- 這條路不比學測輕鬆,越早決定要特選越好
- 2. 學測-APCS 組
  - 還是要準備學測,標準相對的比較低
  - 可以說是第二次的特選
  - APCS 觀念 4 實作 4 (以上)

22 / 42

### 1. 特殊選材

- 不用考學測
- 要有出色的競賽成績,或是你很特殊
- 在程式上要付出很多心力
- 這條路不比學測輕鬆,越早決定要特選越好

### 2. 學測-APCS 組

- 還是要準備學測,標準相對的比較低
- 可以說是第二次的特選
- APCS 觀念 4 實作 4 (以上)
- 名額少

# 各式各樣的 Judge

- CSES
  - 內容最完整,很多經典題
  - 僅能用 C++
  - 全英文題目,可搭配 USACO 練習
- 2. Codeforces
  - Rating + (200 to 300)
  - 全英文題目
- 3. Zerojudge
  - 水題很多,搞人的題也很多,建議配題單刷
  - APCS 歷屆試題
- 4. TIOJ
  - 如果你真的很猛你已從頭刷到尾,不然一樣配題單刷
- Leetcode
  - 配主題刷

我統整的題單: https://hackmd.io/@garycba/r1daKnNBn

# 各式各樣的線上賽

#### 1. Codeforces

- 全英文
- 比賽時間通常落在周末 22:00 到 01:00
- 不限程式語言

#### 2. Atcoder

- 全英文或日文
- 比賽時間通常落在周末 20:00 到 22:00
- 不限程式語言
- C++ 不支援 <bits/stdc++.h> 標頭檔

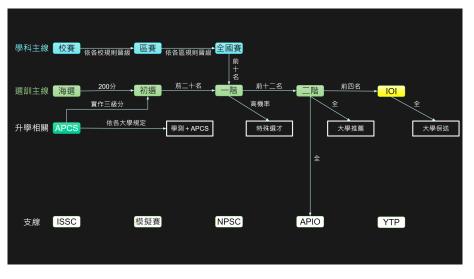


Figure: 競賽路線

◆ロト ◆個ト ◆差ト ◆差ト を めなべ

陳柏安 程式競賽導論 2023-03-17 25 / 42

# 時間估算

陳柏安 程式競賽導論 2023-03-17 26 / 42

# 何謂時間複雜度

■拿計時器在旁邊算

陳柏安 程式競賽導論 2023-03-17 27 / 42

# 何謂時間複雜度

- 拿計時器在旁邊算
- 運算量隨數據上升而上升的指數關係

陳柏安 程式競賽導論 2023-03-17 27 / 42

# 何謂時間複雜度

- ■拿計時器在旁邊算
- 運算量隨數據上升而上升的指數關係
- 大致了解數據量與運算次數的關係

陳柏安 程式競賽導論 2023-03-17 27 / 42

# 簡單運算次數

- 所謂簡單運算就是程式裡的單一一次的加減法、變數設定、交換陣列數字
- 而我們令 T(N):「當輸入為 N 時,簡單運算的次數」

陳柏安 程式競賽導論 2023-03-17 28 / 42

請問這個程式碼的 T(n) 簡單運算做幾次? (請到 slido 回答)

```
for(int t=1; t<=10; t++){
    for(int i=1; i<=n; i++){
        for(int j=1; j<=n; j++){
             cout<<"Hello World!\n";</pre>
```

◆□▶ ◆御▶ ◆巻▶ ◆巻▶ ○巻 - 夕久で

陳柏安 程式競賽導論 2023-03-17 29 / 42

# 漸進的時間複雜度

```
Big - O : f(x) = O(g(x)) 表示 g(x) 的成長趨勢大於等於 f(x) Big - Omega : f(x) = O(g(x)) 表示 g(x) 的成長趨勢小於等於 f(x) Big - Theta : f(x) = O(g(x)) 表示 g(x) 的成長趨勢與 f(x) 相同
```

■ 表示一個函數接近極限 (趨勢) 的行為

◆□▶ ◆□▶ ◆壹▶ ◆壹▶ □ 9000

陳柏安 程式競賽導論 2023-03-17 30 / 42

# 漸進的時間複雜度

```
Big - O : f(x) = O(g(x)) 表示 g(x) 的成長趨勢大於等於 f(x) Big - Omega : f(x) = O(g(x)) 表示 g(x) 的成長趨勢小於等於 f(x) Big - Theta : f(x) = O(g(x)) 表示 g(x) 的成長趨勢與 f(x) 相同
```

- 表示一個函數接近極限 (趨勢) 的行為
- 在演算法執行時間上我們支關心時間的上界,所以用 Big O

◆□▶ ◆□▶ ◆豆▶ ◆豆 ◆ めへで

# 漸進的時間複雜度

```
Big - O : f(x) = O(g(x)) 表示 g(x) 的成長趨勢大於等於 f(x) Big - Omega : f(x) = O(g(x)) 表示 g(x) 的成長趨勢小於等於 f(x) Big - Theta : f(x) = O(g(x)) 表示 g(x) 的成長趨勢與 f(x) 相同
```

- 表示一個函數接近極限 (趨勢) 的行為
- 在演算法執行時間上我們支關心時間的上界,所以用 Big O
- 詳細的數學證明請大家自行搜尋,我數學不好

陳柏安 程式競賽導論 2023-03-17 30 / 42

# Big - O

$$f(x) = 999N^{100} + 999N^2 + 1 =$$



陳柏安 程式競賽等論 2023-03-17 31 / 42

# Big - O

$$f(x) = 999N^{100} + 999N^2 + 1 = O(N^{100})$$



陳柏安 程式競賽導論 2023-03-17 31 / 42

■ 一次基本運算為 O(1) (加減運算、變數設定...)

陳柏安 程式競賽導論 2023-03-17 32 / 42

- 一次基本運算為 O(1) (加減運算、變數設定...)
- 計算每次迴圈運算次數



陳柏安 程式競賽導論 2023-03-17 32 / 42

- 一次基本運算為 O(1) (加減運算、變數設定...)
- 計算每次迴圈運算次數
- 下圖是一個 O(n) 的程式

陳柏安 程式競賽導論 2023-03-17 32 / 42

- 一次基本運算為 O(1) (加減運算、變數設定...)
- 計算每次迴圈運算次數
- 下圖是一個 O(n) 的程式

```
for(int i=1; i<=n; i++){
    cout<<"Hello World!\n";</pre>
```

2023-03-17

32 / 42

### 請問這個程式碼的複雜度是多少?

```
for(int i=1; i<=n; i++){
    for(int j=1; j<=n; j++){
         cout<<"Hello World!\n";</pre>
for(int i=1; i<=n; i++){
    cout<<"Hello World!\n";</pre>
```

陳柏安 程式競賽導論 2023-03-17 33 / 42

■ 有人會覺得是  $O(n^2 + n)$ 



陳柏安 程式競賽導論 2023-03-17 34 / 42

- 有人會覺得是 O(n² + n)
- 實際上是 O(n²)

陳柏安 程式競賽導論 2023-03-17 34 / 42

- 有人會覺得是  $O(n^2 + n)$
- 實際上是  $O(n^2)$
- 因為當 n 極大時,O(n) 對於  $O(n^2)$ 影響極小

陳柏安 程式競賽導論 2023-03-17 34/42

- 有人會覺得是  $O(n^2 + n)$
- 實際上是 O(n²)
- 因為當 n 極大時,O(n) 對於  $O(n^2)$ 影響極小
- 因此我們可以省略掉 n

陳柏安 程式競賽導論 2023-03-17 34 / 42

- 有人會覺得是  $O(n^2 + n)$
- 實際上是 O(n²)
- 因為當 n 極大時,O(n) 對於  $O(n^2)$ 影響極小
- 因此我們可以省略掉 n 我們稱此為常數

陳柏安 程式競賽導論 2023-03-17 34/42

#### 請問這個程式碼的複雜度是多少?

```
for(int t=1; t<=10; t++){
    for(int i=1; i<=n; i++){
        for(int j=1; j<=n; j++){
             cout<<"Hello World!\n";</pre>
```

陳柏安 程式競賽導論 2023-03-17 35 / 42

■ 啊不就是  $O(10 \cdot n^2)$ ?

陳柏安 程式競賽導論 2023-03-17 36 / 42

- 啊不就是  $O(10 \cdot n^2)$ ?
- 實際上還是 O(n²)

陳柏安 程式競賽導論 2023-03-17 36/42

- 啊不就是 O(10 · n²)?
- 實際上還是 O(n²)
- 係數依舊視為常數

陳柏安 程式競賽導論 2023-03-17 36 / 42

- 啊不就是 O(10 · n²)?
- 實際上還是 O(n²)
- 係數依舊視為常數
- 蝦毀! 為什麼?

陳柏安 程式競賽導論 2023-03-17 36 / 42

- 啊不就是  $O(10 \cdot n^2)$ ?
- 實際上還是 O(n²)
- 係數依舊視為常數
- 蝦毀! 為什麼?
- 因為我們只在乎執行時間的趨勢而已,所以從  $O(n^2)$  便可看出時間 趨勢

36 / 42

陳柏安 程式競賽導論 2023-03-17

- 啊不就是  $O(10 \cdot n^2)$ ?
- 實際上還是 O(n²)
- 係數依舊視為常數
- 蝦毀! 為什麼?
- 因為我們只在乎執行時間的趨勢而已,所以從  $O(n^2)$  便可看出時間 趨勢
- 所以係數也省略

陳柏安 程式競賽導論 2023-03-17 36/42

# 均攤時間複雜度

```
for(int i=1; i<=n; i++){
    if(i == 5){
        for(int j=1; j<=n; j++){
            cout<<j<<'\n';
    }else{
        cout<<i<<'\n';
```

陳柏安 程式競賽導論 2023-03-17 37 / 42

■ i = 5 時是 O(n), 其他狀態都是 O(1)

< ロト < 個 ト < 重 ト < 重 ト 三 重 ・ の Q ()

陳柏安 程式競賽導論 2023-03-17 38 / 42

- i = 5 時是 O(n),其他狀態都是 O(1)
- 這種狀況時 i = 5 的複雜度會被其他的均攤掉

陳柏安 2023-03-17 38 / 42

- i = 5 時是 O(n),其他狀態都是 O(1)
- 這種狀況時 i = 5 的複雜度會被其他的均攤掉
- **■** *O*(*n*)

陳柏安 程式競賽導論 2023-03-17 38 / 42

## 量級比較

$$O(1) < O(logn) < O(\sqrt{n}) < O(nlogn) < O(n^2) < O(n^3) < O(2^N) < O(n!) < O(n^n)$$

◆ロト ◆個ト ◆園ト ◆園ト ■ りへ○

陳柏安 程式競賽導論 2023-03-17 39 / 42

## 時間複雜度的標準

■ 大家一定很疑惑到底怎樣的時間複雜度才是可行的

陳柏安 程式競賽導論 2023-03-17 40/42

## 時間複雜度的標準

- 大家一定很疑惑到底怎樣的時間複雜度才是可行的
- Judge 每秒鐘可執行約 10<sup>8</sup> 的「簡單運算」

陳柏安 程式競賽導論 2023-03-17 40 / 42

## 時間複雜度的標準

- 大家一定很疑惑到底怎樣的時間複雜度才是可行的
- Judge 每秒鐘可執行約 10<sup>8</sup> 的「簡單運算」
- 所以當你的程式的 Big O 估出來超過  $10^8$  便會卡 TLE

陳柏安 2023-03-17 40 / 42

## 來點例題

### no judge

有一個長度為 n 的序列 a,請找出兩個數字相加 x。 測資範圍

$$1 <= n <= 10^9$$

$$1 <= a_i <= 10^9$$



41 / 42

陳柏安 程式競賽導論 2023-03-17

謝謝大家