

競程觀念

進階教學組

August 14, 2022

目錄

1	賽制介紹	1
2	需掌握的知識	1
2.1	APCS	1
2.2	IOI/TOI	1
3	團體必賽的工作分配	2
3.1	打 code	2
3.2	算數學	2
3.3	通靈	2
4	如何 debug	3
4.1	黃色鴨鴨法	3
4.2	追蹤變數	3
5	引用文章	3

1 賽制介紹

特性 賽制	賽中評測	排名	懲罰 (penaty)	比賽、考試
ACM	✓	✓	✓	codeforces
OI	✗	✗	✗	APCS
IOI	✓	Δ	✗	TOI

2 需掌握的知識

2.1 APCS

1. 初階班的所有東西
2. 函數呼叫與遞迴
3. 基礎資料結構

- queues
- stacks
- tree
- graph

4. 基礎演算法

- sorting
- searching
- greedy
- dynamic programming

2.2 IOI/TOI

這個就什麼都不嫌多，因為可能有些人的目標不是這個，所以就有興趣的再來找我，我可以請學長教我們。

3 團體必賽的工作分配

如果是要比NPSC或著是YTP這類需要組隊的比賽時，分工就會變得很重要，借用前一屆的整理，通常會需要有

1. 一個打 code 的人
2. 一個算數學的人
3. 一個通靈的人

3.1 打 code

就需要把抽象的解題方式化為具體的程式碼，通常打字速度要足夠快，大概 6,70 左右應該就足以應付了。在打 code 的時候要盡量的讓其他人能夠看懂，所以對於一些地方可能需要著墨一下，像是函式化編程抑或是變數名稱還有宏樣式，都需要先跟隊友討論過，確保在比賽的時候不會出現問題。

3.2 算數學

對於每一個比賽，通常都會有數論題，算數學的就是需要把數論題解決，無論是觀察性質亦或是簡化計算，都可以由算數學的來完成，有時候也可以透過一些數學的公式抑或是證明來讓複雜度減少。

3.3 通靈

比賽時，常常會出現一些優美的做法，可能可以一下子讓複雜度下降一個量級，從 NA0% 直接變 AC100%，這就是通靈的工作，他需要想出題目的解法，然後把他表示給數學家，確認正確性以後再交給 coder 打出來。可能需要比較天馬行空一些，抑或著練過了許多題目，可以快速的抓出題目的特性。

4 如何 debug

4.1 黃色鴨鴨法

如同字面意思，你可以想像自己面前有一個黃色鴨鴨，就對鴨鴨講述你的解法和每個 code 區間的作用，不用擔心講的很爛，黃色鴨鴨很有包容心的。

使用黃色鴨鴨法可以讓你重新整理自己的思緒，跟費曼學習法有一曲同工之妙，只要熟練掌握，相信無論在哪裡都會很實用的。

4.2 追蹤變數

你可以把程式的變數輸出出來看是否符合正確的過程，如果發現錯誤你就可以開始二分搜錯誤的地方，這通常花費的時間比較多一些，

為了節省時間，你可以把輸出變成一個函式。不過這需要用到 template，所以就先放到以後了。

5 引用文章

[编程比赛三大赛制介绍（ACM 赛制、OI 赛制、IOI 赛制）](#) [演算法筆記](#)