111 學年度成設班進階班規劃

進階教學組

August 6, 2022

課程規劃

1 課程目的

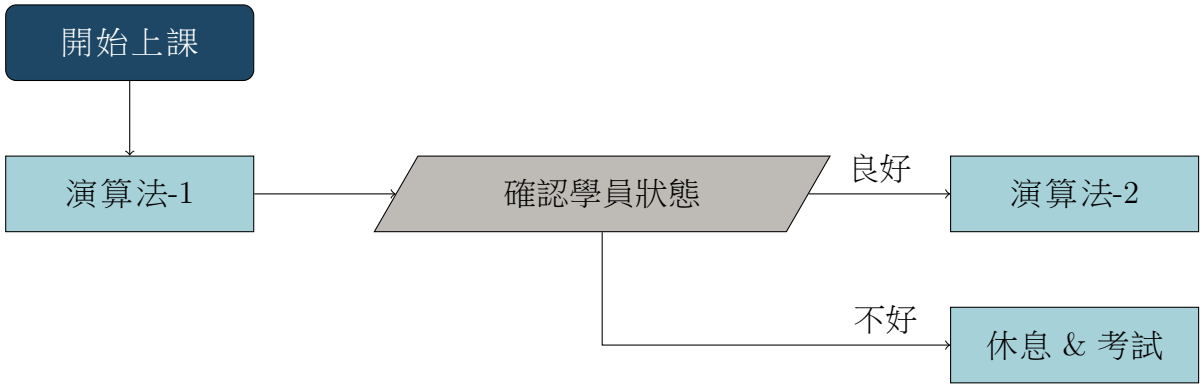
2 教學方式

2.1 課程時間安排

因為講義應該會提早生成出來，所以目前預計是能上就盡量上，如果有發現學員出現理解困難的時候，應該會讓大家休息一下，透過考試複習以前學過的東西。

;

整體流程大概如下



2.2 上課

上課目前還在思考要以哪一中方式來教學，目前有幾種想法。

1. 維持之前的教法，在課堂上把每一個觀念講清楚

   優點:

   (a) 可以最大程度的提供優質的教學

   (b) 可以讓較不主動的學員學到基本的演算法

   缺點:

   (a) 需要花的時間較多，能教的演算法較少

   (b) 因為需要在聽的時候理解，所以可能會導致學員因為無法理解而喪失自信心抑或是熱誠

2. 提前兩到三個禮拜把講義釋出，然後上課讓他們提問。

   優點:

   (a) 進度可以進展的比較快

   (b) 透過上課問問題的方式，可以叫清楚的了解學員們的問題並且提供適當的解釋

   缺點:

   (a) 學員可能害怕提問導致課程變得很尷尬

   (b) 可能沒有人會去讀講義，導致沒什麼學員學到東西

   (c) 很看中講義的詳細度，如果不夠詳細就會導致學員學習效率降低的行為

3. 在上課時直接解題，讓學員誰可以提問不會的語法，和不了解的部分。

   優點:

   (a) 可以讓學員熟悉實作方式和 debug 技巧

   (b) 透過邊實作邊講解的過程可以讓學員更了解每一區塊的程式碼所需要完成的事情，進一步了解原理

   缺點:

(a) 現場實作很考驗教學的功力，還有打字的速度

(b) 時間較難把控，因為 bug 可能會出現

(c) 可能會出現學員來不及理解 code 就打完的情況

現在一切都還沒有定論，總之還是需要看學員參與程度來決定使用哪一種教學方式。

2.3 考試

2.3.1 小考

小考應該會相對的較隨性一點，可能會定期，也可能會在讓學員們放鬆的練習一下的時候目前預計每次小考會有 3 至 4 題。分配預計會有

- 1 題水題
- 1 至 2 題模板題
- 1 題難題

就希望可以讓學員們有一定的成就感，也讓實力超群的人可以挑戰不同的題目。

2.3.2 期中考

期中考則是需要認真的去出題，希望每一題都是新的，也可以有幾題十分有挑戰性的題目，目前預計是 6 至 8 題，分配預計會有

- 2 至 3 題水題
- 4 至 5 題模板
- 1 至 2 題難題

難題配置較少的原因是因為下一屆的程度可能不太樂觀，所以就儘量的放在扎實觀念上面，當然，模板題也是會有變化的，可能還是需要一點時間去思考。

2.4 作業

基本上作業只會用來加分，因為進階班的作業都會有點難度，所以就留給想要拿到資優結業的人了。每個禮拜應該都會有作業，主要就是模板題，因為要在短短幾周內就學會一個演算法確實不太容易。

2.5 講義製作方式

主要是以 latex 為主，如果遇到需要當節課就產出來的話再使用 hackmd(譬如: 當節小考題解)，在講義的部分，我想信 latex 有無限的可能性，目前比較熟悉的可能是流程圖製作的部分。所以以後可能會在演算法的過程中放下流程圖，在 latex 的部分，受到了建中 2016 講義的啟發，我也想來做一個類似的東西，所以之後的講義可能也會分成很多個大觀念，之後再分成小觀念，讓之後的人可以比較方便的參考。

然後講義我想再暑假把他爆肝出來，要不然高二好像太忙了，所以之後可能會在上課前兩三個禮拜就有講義可以讓他們先行預習，也許可以增進理解的效率。

2.6 預計的課程風氣

我們希望進階班是可以讓學員盡情討論的地方，所以可能之後可能也會如同**資訊枝枒**一樣開 slido 匿名討論，遇到不懂的就問出來總有一個人可以幫你解決的，也希望課堂上能上學員發表他們的意見，這樣也許在意見的碰撞當中就會出現新的想法。

3 必教

3.1 分治

概念簡述

透過把問題不斷的拆分（通常是拆成兩半)，拆到不能再拆的時候再開始合併，再合併的同時處理左右兩邊的值，最後就可以得到答案。

重要觀念

1. 遞迴的觀念要講清楚，不然後面理解會困難
2. 通常是從 **左邊** $\Longrightarrow$ **右邊** $\Longrightarrow$ **處理**
   不過遇到 **左邊** $\Longrightarrow$ **處理** $\Longrightarrow$ **右邊**需要特別注意。

題目

- 逆序數對
- 最近點對

可延伸的觀念

1. CDQ 分治
2. 整體二分搜

3.2 DP

概念簡述

透過把小問題儲存起來，讓之後使用的時候可以快速的查詢。可以把許多暴搜 $O(2^N)$ 變成 $O(N^K)$ 且 K 為常數

重要觀念

1. 對於子問題的轉換需要清楚的解釋
2. 重點要放在轉移式上，只要轉移式寫的出來就幾乎解決一切了
3. 對於優化可以先不用那麼急

題目

- LIS　最長遞增子序列 $O(N^2)$
- LCS　最長共同子序列 $O(N^2)$
- 背包問題
  - 0/1 背包
  - 無限背包
  - 有限背包
- 費氏數列和其延伸

可延伸的觀念

1. 單調對列
2. 斜率優化
3. 狀態壓縮
4. 四邊形優化 (目前會: 50%)
5. Alien 優化 (50%)
6. SOS 優化 (0%)
7. 枚舉子集 (0%)
8. 多重背包 (0%)

3.3 greedy

概念簡述

透過每一次都選最好的可能性來大幅度的縮短演算法時間。

重要觀念

1. 要讓他們學會證明方式 (數學歸納法)，還有值觀的猜測
2. 讓他們學會如何去轉換問題 (雖然我好像也不太會)

題目

- 最多不相交線段
- 排程
- 和其他經典例題 (暫時沒想到)

可延伸的觀念

1. 最短路徑/最小生成樹
2. 有時候會配合二分搜使用

3.4 DFS/BFS

概念簡述

一種對於圖形或樹狀圖搜尋的方式，分別有深度優先 (DFS)，和廣度優先 (BFS)

重要觀念

1. BFS 通常使用 queue 實作
2. DFS 通常使用 **遞迴**實作
3. DFS 如果沒有加上 vis 判斷複雜度是以階乘成長的
4. 可能需要用類似動畫的方式來實際模擬一次

題目

- 數獨
- 方格圖上的最短路徑
- 圖的直徑

可以延伸的觀念

1. 拓鋪排序
2. 最大二分圖匹配
3. Tarjan

3.5 二分搜

概念簡述

對於有單調性的函數可以進行的操作，複雜度通常帶有 $O(logN)$

重要觀念

1. 函數需要有**單調性**
2. 需要注意二分的方式，不然很容易無限迴圈
3. 對於陣列已經有實作好的 lower_bound 了

可以延伸的觀念

1. 整體二分搜

3.6 應用

概念簡述

就純粹想分享我對於一些題目的看法

題目

1. 2021 年學科能力競賽
2. 2022 年學可能力競賽 (還不確定，可能我都不會)
3. Candle
4. 台大入營考 A

可以延伸的觀念

目前還無法評估

3.7 線段樹/BIT

概念簡述

線段樹

透過對每兩個小區塊分配一個父親，不斷的重複，對小區塊的值做處理，直到剩下一個父親 (根節點)。

BIT

透過數論的方式來達成近似於線段樹的操作，不過對於可以處理的方式有更多的限制。

重要觀念

1. BIT 是有限制的，不能一昧的套用
2. 也需要把遞迴的過程和概念講清楚，不然理解會十分困難

題目

1. 模板題 *N

可以延伸的觀念

1. 指標線段樹
2. 持久化線段樹/BIT
3. 李超線段樹
4. zkw 線段樹 (0%)
5. 時間線段數 (0%)
6. 線段數優化建圖 (0%)
7. 2D 線段樹/BIT
8. 樹套樹

3.8 最短路

概念簡述

在一個有向圖或無向途中找出起始點到終點的最近距離分別有三種演算法和一種優化法，複雜度分別為 $O(V^3), O(VE), O(E \sim VE), O(VlogE)$

重要觀念

1. 需要解釋清楚每一個演算法的作用過程
2. 需要讓學員了解每一個演算法的適用範圍

題目

1. 帶負環的最短路徑
2. 需同時從兩邊開始的最短路徑
3. 模板題 *N

可以延伸的觀念

1. Dinic(20%)

4 選修

4.1 矩陣

概念簡述

原本為高二的課程，但是進階班會稍微提到。主要會放在矩陣快速冪上。

重要觀念

1. 矩陣乘法
2. 矩陣的交換律可能不行
3. 快速冪

題目

1. 費氏數列 (矩陣快速冪)

4.2 BST

概念簡述

BST 是一個很實用的演算法，除了可以練習指標的使用之外，也可以訓練思考遞迴的能力。

重要觀念

1. 遞迴過程需仔細想過
2. 在刪除節點的時候需要先把子樹的刪掉

可延伸觀念

1. 逆序數對
2. 線段樹
3. Treap
4. map/set

4.3 字串算法

概念簡述

字串算法十分的繁雜

1. KMP
2. Trie
3. Zvalue
4. 回文
5. AC 自動機 (0%)
6. 回文自動機
7. 後綴數組 (0%)

4.4 LCA/倍增法

概念簡述

可以在 $O(nlogn)$ 時間預處理，然後 $O(logn)$ 的時間獲得 LCA。

通常會先使用 DFS 來獲得每個節點的進入時間和離開時間 (T_in, T_out)，之後就可以 $O(1)$ 判斷兩個節點的祖先關係。

重要觀念

1. 對於次方的處理需小心
2. 如何判斷是否為子孫關係

4.5 Treap

概念簡述

是一個實務上不實用但在競賽上很好用的東西，透過一些隨機的方式來讓二元樹的操作複雜度為 $logn$ 透過 fhq treap 也可以做到許多線段樹做不到的事情 (e.g. 區間翻轉)

重要觀念

1. 對於複雜度的計算十分困難
2. merge、split 函式一定需要仔細思考
3. 需要非常精熟指標，要不然很容易以為他每個操作為 $O(N)$
4. 在 node 為 null 的時候不能進行操作

4.6 DSU

概念簡述

可以動態判斷分組結果的演算法，透過維護 father 陣列來獲得較好的複雜度。使用路徑壓縮或啟發式合併可以得到 $O(logn)$ 的搜尋時間如果同時使用可以得到偽線性複雜度。

重要觀念

1. 啟發式合併的原理
2. 路徑壓縮的正確性
3. 持久化時需要注意的事項

4.7 最小生成樹

概念簡述

找到一個連接所有點的樹且讓邊的權重最小。

重要觀念

1. 使用 DSU 的方式
2. 和最短路徑的差別
3. 次小生成樹如何做

4.8 單調隊列

概念簡述

一種 DP 的優化方式，相較於斜率優化、四邊形優化、Aliens 優化簡單了需多，只要會概念，實作起來並不困難。

重要觀念

1. 如何維護好單調性
2. 如何判斷出單調性
3. 實作需要注意的小細節

4.9 向量/凸包

概念簡述

計算幾何，就滿滿的數學。

重要觀念

1. 對於向量運算的熟悉度

4.10 莫隊

概念簡述

一種離線的演算法，可以透過對詢問進行分塊跟排序來得到較好的複雜度，也可以處理帶修改的情況

重要觀念

1. 對於複雜度的證明
2. 需要注意的事情（e.g. 要把詢問排序回來)
3. 在轉移的時候的先後順序

4.11 整體二分搜 (毒)

概念簡述

和 CDQ 分治十分類似，是比較難解釋清楚的演算法，因為已經被基礎的二分搜定下了觀念，對於整體不太能想像。

也是一個離線算法，對於帶修改的資料也比較能夠去應付

重要觀念

1. 是如何把詢問分成左右兩遍的，和如何維持左右兩區間的順序。
2. 對於修改操作如何實作

4.12 CDQ 分治

概念簡述

分治的一些特殊用法。通常用於離線算法。有時也可以把 $O(N^2)$ 的 DP 變成 $O(Nlogn)$

重要觀念

1. 對於分治的基礎理解
2. 處理左右區間的順序

4.13 神奇的類二元樹

概念簡述

這是我某天突發起想出來的資料結構，主要透過位元運算來時做複雜度為 $O(nlog^2C)$ 或著 $O(nlogCloglogC)$

重要觀念

1. 對於位元運算的觀念
2. 對於完美二元樹的理解
3.「億」點點的數論

4.14 SCC/BCC

分別為點雙連通分量和邊雙連通分量。

概念簡述

可能不太能簡述，因為需要涉及的知識有些多

重要觀念

1. DFS tree
2. 圖的連通性
3. 點雙連通分量，邊雙連通分量
4. Tarjan