資料結構與演算法：97

計算機結構與作業系統：68

英文：66

考試總分：171.6

網媒所錄取最低分：123.2

資工所錄取最底分：183.0

背景：台大資工系黑卷

服用必要前提：本科系或一百倍努力 & 英文查資料能力

服用適用範圍：僅限台大網媒所、資工所

【大心法】

**補習絕對不夠**：

* 補習班的教材適合台大以外的學校，但要考台大資工網媒絕對不夠。
* 補習班教材還是不錯，可以利用但千萬不要依賴。

資料結構與演算法：

* 不用補習，補習班的題目還是太少且變化少，且相同的題目一直刷會變笨。
* 請持續不斷的找新算法看、新題目做（下面的資源應該夠做），不要以考上台大為目標，請以成為一個真正合格能被認可的資工人為目標，不然你考進來還是被歧視。
* 同樣的題目做過兩三次就好，確保透徹理解就不會忘，會忘就是沒有透徹理解。
* 不要背code，背不完又沒意義，掌握算法直覺能夠描述即可。
* 學會證明正確性很重要，能說明正確性才代表透徹理解。很多時候你想不到怎麼修改經典算法來解題，就是因為對正確性不求甚解。那種稍難的經典算法變化題，通常修改個一兩行就能work，但要改哪裡、怎麼改，改完你能否確定他會對？還是憑直覺亂猜？就要靠你對該算法正確性的理解。

計算機結構：

* 我是以台大洪士灝教授的投影片為主軸抓重點，然後再去看白算盤原文。
* 白算盤真的是我見過最神的教課書，裡面每一段每一個elaboration每一個real case study都讓我功力大增，有更全面的認識。尤其是近年必考的平行化：第二（ll, sc）、三（SSE, AVX）、四（multiple issue, loop unrolling）、五（block multiplication）、六（threading）後面都有一節矩陣乘法加速，與第六章整章，真是太太太太精華了。你覺得平行化的技術太難掌握所以只專精pipeline和memory？那就考不上台大，尤其是洪士灝教授一定把這個考爆。
* 張凡老師那本完全都是白算盤的內容（和一點點點進階的另一本A Quantitative Approach），算是幫抓重點出來，但我覺得還是漏掉滿多我覺得重要的。因此我是讀完白算盤原文一到六章兩三遍，最後拿張凡老師那本快速複習。至於要不要讀A Quantitative Approach我只能說，讀了當然最好，但倒是不必要。你能認真看懂幹完白算盤就比一堆人有競爭力了。
* 刷題毫無意義，浪費時間台大又不考，請專注觀念上的理解，和與作業系統的整合。唸到最後一定要是作業系統和計算機結構一起念，互相補充。

作業系統：

* 我也不知道怎麼唸..................................................................
* 我就把郭大維教授的影片和投影片看完，買薛志文教授的那本作業系統看分散式那章節，做考古題倒是都能講出一點東西來。
* 但今年似乎投資報酬率真他媽太低了，考linux system programming的東西（對應到台大的System programming這門課，有修過就會），然後分散式那題作文我整個只能從薛志文教授的書中，拿些似乎相關的概念亂寫............
* 阿...........真是令人絕望的一科。
* 最後計系68分我覺得有很多分數掉在分散式那題。我認為計算機結構可念，但作業系統我就不知道了.......................很可能我的方法還是錯誤，需要再等大神分享。

【資源】

**演算法與資料結構：**

手寫筆記：

* [資料結構](https://1drv.ms/f/s!AvyMcQSCeyyEjH__bCUAn-OCfmwy)
* [演算法](https://1drv.ms/f/s!AvyMcQSCeyyEjUdIQ6rYDIkSL4nU)

網路資源：

* [Geeksforgeeks](https://www.geeksforgeeks.org/)（重要）：真他媽寶庫，一堆變化相關題型
* [MIT midterm / final](https://courses.csail.mit.edu/6.006/oldquizzes/)（重要）：功力指數起飛
* [Stanford data structure](http://web.stanford.edu/class/cs166/)（選讀）：通體舒暢，不過要挑著看有些太偏激，未來有空會詳列哪些值得看。
* [Stanford second course in algorithm](http://timroughgarden.org/w16/w16.html)（選讀）：我只有看max-flow相關的前三四堂課，講得非常棒，尤其證明部分。看完這邊的課程就能再去重看一次維基百科的push relabel algorithm，懂的更全面。
* CLRS後面題目（選讀）：因為題目實在太多了，所以當選讀。我只有寫hashing那章節的題目，因為我發現近三年台大開始愛考hashing。需要解答的話我有個朋友有架一個網頁，上面整理了滿多詳解的：<https://walkccc.github.io/CLRS/>
* 不管是MIT還是台大，都做期中期末就好，除非時間爆多不然不用做他們的作業。作業的難度通常是設計給時間兩個禮拜以上寫的，跟考試題目有本質上的不同。
* 正八整理的很棒的題型彙整（不適合一開始看，而是基本的都熟了之後）：<https://hackmd.io/09d0YvUHQSOH82fGsDaLjA?view&fbclid=IwAR3CahBGBg2Gu3WGvHuX7rb-RayGTw41EiBm3QNs2ZXWK2JZGhXkQKqwTeU>

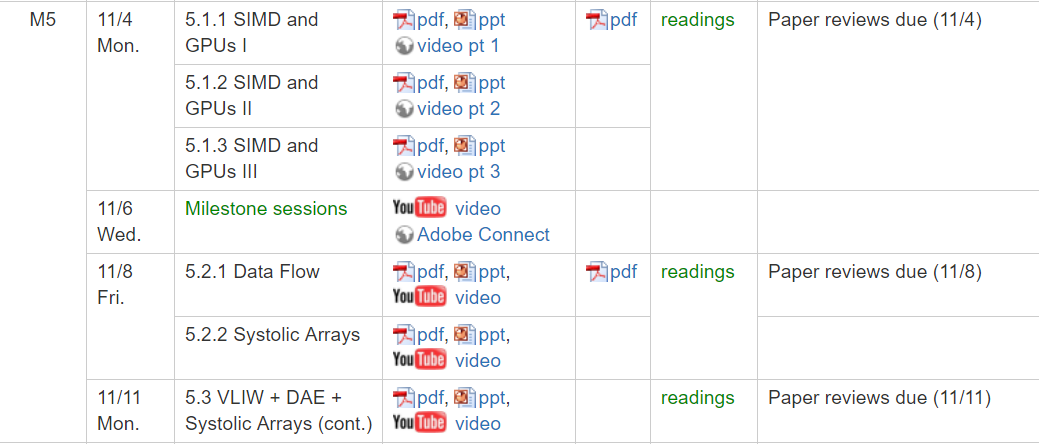
**計算機結構與作業系統：**

手寫筆記：

* [計算機結構](https://1drv.ms/f/s!AvyMcQSCeyyEjlG7Eyd7q3WoWNSp)
* [作業系統](https://1drv.ms/f/s!AvyMcQSCeyyEjj_JRE83cDEBMJbL)

網路資源：

* 主要都是念課本，看不懂的查
* [CMU Computer architecture](http://www.archive.ece.cmu.edu/~ece740/f13/doku.php?id=schedule)（選讀）：這是進階的計算機結構，我主要看的是截圖的部分，因為白算盤這部分沒有講太細，所以覺得不舒服的話可以服用：



累了............有空再繼續打，之後會想再加上我自己的手寫筆記和一個念過的所有經典演算法清單。不好意思有點亂QQQ 但分享這些資源的用意並不是要給出整理精美的教材讓大家可以馬上讀馬上上榜，而是分享一個準備的過程，真正重要的還是要靠自己自學。畢竟跟著補習班教材刷刷題、規劃時程這種心得文版上有很多了，我再寫的話也不會差多少。總歸一句話，自己要努力，努力不是你花了多少時間自己讀了多久，而是你為了真正讀懂他下了多少功夫，找多少資源，問多少人。所以有人可以準備一年，有人準備半年，有人覺得兩個禮拜就夠。念多久從來就不是重點，念多深刻才是決勝的關鍵。

最後還是跟打氣一下，現在開始準備就還有一年！一年真的很久的！

Fight！！

聯繫我：leo19941227@gmail.com