Algorithm Design and Analysis
Homework #1 Hint

Hint instructions

這份提示集結了 Office Hour 中常被問到的問題與回應，內容有可能會破壞思考與學習的樂趣。請在嘗試思考後仍不得其解時再使用。作業中的 Reference 欄位中歡迎填本文件。

Problem 1 - Manhaha Problem (programming assignment)

1) 有些同學有學過另一個 closest pair 問題(找出一個歐氏距離最近的點對)，它的解法與這題的解法並不那麼相關(雖然都是 Divide & Conquer)，請先暫時把它忘記，不然可能一直往不對的方向想。
2) 絕對值的計算方式是一個很重要的關鍵，要怎麼把絕對值拆開？
3) 先試著想看看一維的情況怎麼做(通常遇到一個問題解不出來的時候可以先試著解解看它的簡化版，或許能有一些收穫)：用一次 sort 後只要看每個點的前一個跟後一個就夠了(為什麼？)，那這個方法能不能用在二維情況上？二維有哪幾個方向要看？
4) 把計算距離的式子實際寫出來，應該會有所發現。
5) 使用不同複雜度演算法各大概會得到的分數：

   O(n^2) – 3/10 分
   O(n*log(n)^2) – 8/10 分
   O(n*log(n)) – 10/10 分

Problem 2 - Super Safe

1) 彼得總得要有某一步打開第 n 個按鈕，此時前 $n-1$ 個按鈕的狀態一定要是「除了第 $n-1$ 個按鈕是被按住的，其他全部都被打開」。所以從這一步分開，這步之前的步驟與之後的步驟是否是兩個(或更多個)子問題？請記得子問題不一定要分成兩個喔。

   在敘述時，如果你需要逆著操作回去的話，請記得簡單提一下為什麼可以逆操作？譬如說打開 Type-3 保險箱需要 5 步的話，為什麼將 Type-3 保險箱從全打開變回全按住也是 5 步？
2) 第一小題會寫的話這題相信也沒問題！

   如果你的遞迴式很長很長，譬如說 $a_n = n + 2(a_{n-1} + a_{n-2} + \cdots + a_1)$，它當然是正確的遞迴式，不過在進入第三小題以前可能需要化簡一下，不然會很

痛苦。

3) 猜不到嗎？送你一個實用的網站：http://oeis.org/A000045
   你如果會直接解遞迴式當然最好囉！不然像這樣的題目其實只要猜到答案後再用數學歸納法就可以了。記得歸納基底要寫。
4) 若你第二小題的遞迴式只牽扯到 a_{n-1}, a_{n-2} 的話，你所困擾的應該是第二部份會有一個狀態（「除了第 n 個按鈕被打開，其他全部被按住」）是沒辦法確定一定要經過的。那你可以試試看先寫第五小題。
   若你第二小題的遞迴只牽扯到 a_{n-1} 的話，我想不出來這題該怎麼作，你可能一定得先寫第五題。
5) 奇怪的是這題的遞迴式比原題還乾淨，表達式比原題還漂亮，證明它是最佳解法比原題還簡單（數學歸納法？）。表達式有多漂亮？你如果沒感覺的話不如把 $n = 4$ 時的步驟畫下來觀察看看。

Problem 3 - Read Your Textbook

教授上課教了三種相關的方法：取代法、畫樹、大師定理。

取代法首先你必須先猜出他正確的複雜度，因為這題是上下界都要證明，所以如果只是猜一個上界是不夠的，要直接用它真正的複雜度當上界（與下界）。

畫樹基本上蠻好用的。畫出一棵樹後有三個步驟要做：算出每一層多出來的項、算出有幾個基底（base case, T(1)）、算出樹有幾層。如果是要準確一點的話，上面的「算出」其實是「估計出」。譬如說課堂上講的 $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$ 就必須要估計樹的層數界在 $\log_3 n$ 到 $\log_{1.5} n$ 之間作為上下界。

大師定理有固定的形式，所以題目一看就知道能不能用大師定理。有些同學還沒理解課堂上的 ϵ。它不一定要是正整數，而且是只要存在一個 ϵ 可行就可以了。所以你如果取 $\epsilon = 1$ 不行的話，不代表大師定理不能用，你或許可以取 $\epsilon = 0.0001$。

另外也有同學可能忘記 $logn = O(n^{0.0001})$，也就是對數函數其實跑得比任何 n^k 都還要慢。

1) 大師定理？畫一棵樹？你或許會需要計算一些等比級數…
2) 大師定理？畫一棵樹？你或許會需要

$$ln(x+1) < 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{x} < 1 + ln(x)$$

或是比較複雜的

$$\ln(\mathrm{r}+\mathrm{d}+1) - \ln(\mathrm{r}) < \frac{1}{\mathrm{r}} + \frac{1}{r+1} \ldots \frac{1}{r+d} < ln(r+d) - \ln(r) + \frac{1}{r}$$

有沒有喚醒你遙遠的微積分記憶！？

3) 畫一棵樹？或是瀟灑地猜答案後用取代法。
4) 畫一棵樹？取代法？

Problem 4 - Dividing and Matching by Segments

1) 注意如果僅知道一個實數值會越來越小，只能知道它會趨近於某個數字，並不能保證最後它會變成定值。想想1, 0.1, 0.01, 0.001, ...就知道了。但是你其實知道 n 個點連 n 個點連法是有限的…
2) 這一題的 Divide and Conquer 主要的難點不在如何合併，而在於如何 Divide。想像如果有一條直線可以把這些點分成兩邊，並且每半邊的藍點跟琥珀點一樣多，那我們就可以把這兩半邊各自當作獨立的子問題 Conquer 下去，第一小題告訴我們只要點數一樣多就一定有解，所以這兩半邊一定辦法把點配對後使得線段不相交，又因為他們在直線兩側的關係，兩邊的解一定不會互相交叉，所以直接合併起來就是一組解。（如果你在答案中要講這些概念，請不要講得像上面那麼隨便。）當然這裡「一條直線把點分成兩邊」是指兩邊都要有點，子問題的 Size 才會越來越小。
   所以這條直線存在嗎？要怎麼找到這條直線呢？譬如說固定它的斜率，從無限大處慢慢平移到負無限大處，其中一定有某個剎那它就是我們要的直線？或是說假設這個直線固定通過一個點（可能是某個藍點、某個很遠的點、某個 x 座標最大的點、某個凸包上的點），讓它旋轉掃過去，其中一定有某個角度它就是我們要的直線？看凸包？找最小匹配？尋覓可愛數？切法很多種，請發揮創意，但記得複雜度不要飛向雲霄。

Problem 5 – A Lovely Problem (Uh, probably not)

1) 假如我們把 a_i 跟 b_i 一起排好序畫在數線上，然後對於任何實數 x ，定義一個值 $f(x)$ 是「小於 x 的 b_i 個數」減掉「小於 x 的 a_i 個數」。那麼 $f(-\infty) = ?, f(\infty) = ?$ ，$f(x)$ 會怎麼變化？$f(\text{可愛數})$ 在函數圖形上有什麼特徵？
2) a_i 從小搜到大，同時計算有幾個 b_j 比自己小。由於如果$b_1, b_2 < a_i$，那麼自然有$b_1, b_2 < a_{i+1}$，所以有些掃過的 b_i 不需要再掃一次。要解釋為什麼複雜度是 $O(n)$ 喔！
3) 勘根定理！還記得如果你知道$f(a) < 0, f(b) > 0,$ 要怎麼快速找到一個$f(x)$的根嗎？類似於第二小題的性質，若 $b_j < a_i$ 則 $b_j < a_{i+1}$，所以有些掃過的 b_i 也是不用再掃一遍，不過到底那些不用？Office Hour 中也有聽到有同學人是 binary search b_i 然後看 a_i 的，也是蠻酷。
4) 第 3)小題 + Call your MOM.