

程式設計大賽於 2004 年起第一次舉辦至今，已為 Openfind 年度重要的盛事及傳統之一。公司內部人員於比賽期間將打破原部門建置，重新編組，依據比賽題目內容，並在有限的時間及資源下，發揮最大的創意及團隊合作，努力達成目標。2013 年 Openfind 程式設計大賽花絮如下：

題目：Openfind 鐵工廠. 比賽時間：24 小時

競賽規則：[鐵工廠 比賽題目下載](#)

- (1) 給定一塊鐵板，切出給定長方體鐵盒子(長 X 寬 X 高)。
- (2) 一個鐵盒共需要六塊鐵板拼接起來。長 X 寬兩塊、長 X 高兩塊、寬 X 高兩塊。如何切出最多完整的鐵盒子
- (3) 總共 10 個關卡，切出鐵盒總數最多的組別獲勝。

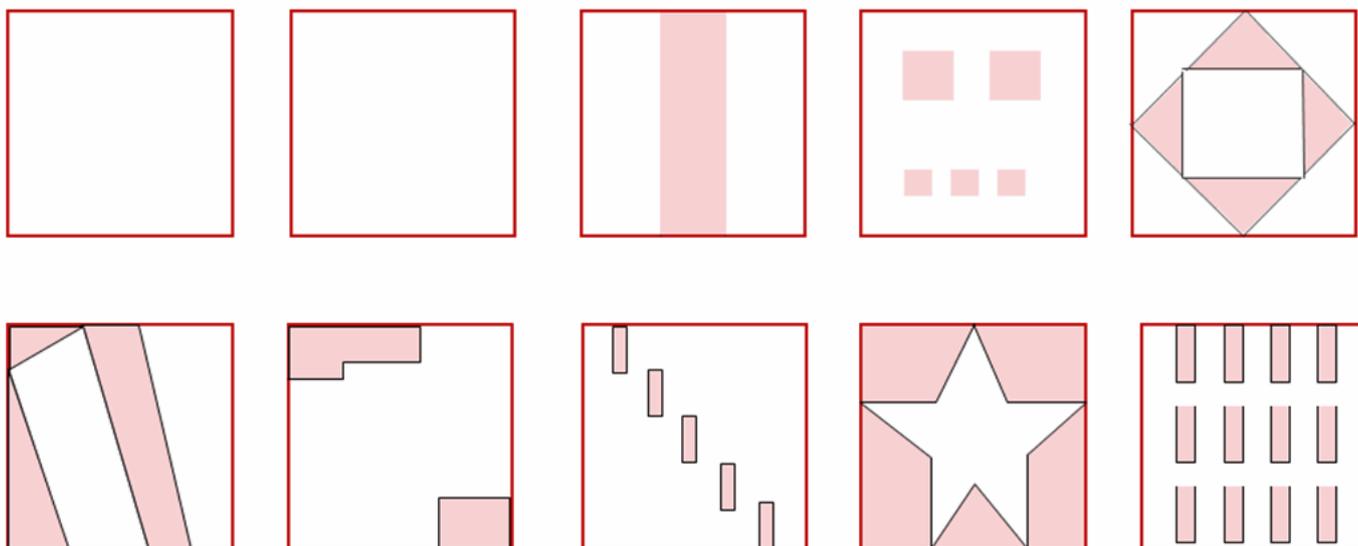
比賽結果名次公布：

- 第一名：Willie, Isaac, David Wu
- 第二名：Henry, Billy Liu, Iris
- 第三名：Jamei, Howard Lee, Odie

戰況剪影：

◎ 比賽地圖

10 關關卡 preview



◎ 比賽結果

	第一組	第二組	第三組	第四組	第五組	第六組	第七組
1	12	12	12	12	12	12	12
2	20	20	20	20	15	20	19
3	10	10	9	9	10	9	10
4	15	14	13	15	6	13	13
5	9	10	9	9	8	10	9
6	9	7	8	9	1	8	9
7	10	11	11	11	3	11	11
8	26	26	26	26	22	26	25
9	19	42	40	43	2	43	42
10	4	7	7	7	4	6	7
SUM	134	159	155	161	83	158	157
名次		2		1		3	4

幕後花絮

<p>Billy Liu</p>	<p>一看到看到這個題目，想了幾個基本策略 但是寫下去才發現問題在要跑很久 有再多策略也沒辦法用(不愧是書安也解不出來的題目) 所以乖乖用基本的方式分割 每個版本使用不同的小策略來比較哪個版本的 output 比較好</p> <p>可惜在有斜邊的第 6 題和第 9 題吃了點虧 還好第 3 題和第 5 題有花時間手工刻 output 還能有個第二名 這種問題的結果差距本來就不大，還要三種板子各兩組才能多切一個 反倒是第 9 題的測資居然設計成能切出 40 幾個 這題的結果幾乎就能決定勝負 這是我認為不太合理的地方 而我認為出的最好的是第 5 題 要是真的有這個圖手工可以切出 11 個，一般用程式跑就是 9 個 可惜測資差了一行，也只能手工刻出 10 個</p> <p>另外不得不佩服 Ansel 一個人就有這樣的成績</p>
<p>Jamei</p>	<p>隊友介紹： Odie：Jamei 跟他借過 library 使用在客製裡。 Howard：Jamei 跟他打過一次羽球，才來 Openfind 一個月還在 pool 被烹煮的新人。</p> <p>就是這樣，看起來一點默契都沒有的隊伍（誤），一進入會議室，馬上就擦出火花啦！ 經過一個多小時熱烈的討論（哦好有參與感好開心喔！）擬出了策略， 分工時我就知道接下來就是讓兩位邏輯清楚的帥哥上場，然後我盡量不要干擾他們這樣（嘆）。</p> <p>晚上聽說第五題測資可以用手刻，於是第二天上午我們也在刻， 但是居然沒意識到第三題也要刻，嗚嗚嗚～～～ 另外第九題其實可以跑出 44 個的，可惜時間不夠，當天我的測試機中間還搞 disk full，殘念！Q_Q 有股「後端產品做得破兒棒，業績卻賣不好」的惆悵惹..... 下次一定會更好的！</p>
<p>Henry</p>	<p>第一次程式設計比賽全程留在公司，清晨日出的景色不錯</p> <p>其實寫完可以跑的第一版大概也才晚上七八點而已，雖然數量還有待加強</p> <p>在沒有圖形介面可以看結果的情況下，要 Debug 難度還不小（幾何苦手，一開始就放棄斜的方塊 XD)</p> <p>跑了一堆測試之後，發現策略不同的結果數量差距有限，最後決戰可能每題只會差一兩個 於是決定先手刻出例題的 output，用人腦來戰勝電腦（多一個就贏了 XD)</p> <p>沒想到最後刻完的時間是 AM03:XX ... (體力用盡)</p> <p>第二天早上再用暴力法生出六種排列組合來操 CPU ... 來多湊出一兩個鐵盒</p> <p>最後跑的結果還算差強人意（很擔心熬夜改的程式會跑很久或跑不出來）</p>

	<p>靠著隊友 Iris 成功產出剩下七題的圖形測試資料 (2001 x 2001 個 0 和 1 看到眼花)</p> <p>Billy 在凌晨三點仍然可以用人腦算出斜的長方形坐標 (我已經快進入昏迷狀態...)</p> <p>最後以第二名作收 (險勝第三名一個，果然手刻才是王道(?))</p> <p>這次印象最深的是 Ansel 在最後的第九題嚇了大家一大跳 (一口氣贏了十幾個，所有人都傻了)</p> <p>不過也證明用斜的擺果然很容易出問題 (雖然問題是出在 checker XDDDD)</p> <p>以上報告完畢 (好像完全沒寫到程式的演算法 XD)</p>
Willie	<p>這是第三次參加程式設計大賽 算是把前幾次的經驗都用上了</p> <ol style="list-style-type: none"> 1. 先做出簡單且會動的版本，再根據輸出和 test case 做調整 2. 每一個版本都要留著，寫一個自動算最佳 output 的 script 3. 要好好睡覺 4. 要好好睡覺，因為很重要所以要說兩次 <p>上次就是沒睡午覺又熬夜 結果寫出一堆垃圾 還把分數比較高的版本改爛 這次睡完午覺，簡單的討論一下決定一人寫一個策略 兩點多開始動工，快五點就有第一個版本了 回去睡覺的時候想了 2 個版本 隔天睡到 9 點，10 點到公司 兩小時內就改了 2 個版本跟 1 個 bug</p> <p>最後總共是 11 個版本取最佳 每個版本的差異就只有切的順序不一樣 有 bug 的版本會多算 1 pixel 不過多那 1 pixel 反而有些關分數更高</p> <p>這次要感謝 david 幫忙產生 test case 協助新版本的產生 isaac 的手刻版幫忙找到那 1 pixel bug 還有 Phantom 的 checker bug 讓我們可以僥倖贏得比賽 XD</p>
Bart	<p>這次比賽就輸在一題的結果上面，有點可惜，可是也只能說在一開始沒想到這種排列方式，經過幾次比賽，已經發現沒幾組在熬夜了，最後我們熬夜看來也沒得到理想的結果，哈哈下次有在比賽機會就是跟大家說不要熬夜才能頭腦清晰，正確地完成比賽!!!!!!</p>
Iris	<p>第一次參加程式設計比賽，當聽到分組後的第一個感想就是「好的隊友帶你天堂」。大家都和我說跟到 Billy, Henry 就完全不用發愁了，但其實我滿緊張的，很怕成為這組的負擔。</p> <p>老實說比賽開始後的一段時間裡我都在狀況外，被排定的工作是用 1 跟 0 排出和題目型狀相近的鐵板，卻不知道該如何下手，尤其是當每次一開範例檔電腦就開始變慢時就更急躁了.....最後選擇了土法煉鋼，自己手動排。但果然作事要找方法的，當發現有現成的工具可以加快完成事情的速度時，頓時感到得救了。</p>

	<p>直到第二天有東西能跑時，才進入狀況知道自己昨天作的事情功用何在。 感謝兩位優秀的隊友，你們太棒了！！</p>
David Wang	<p>這是我第一次參加公司為期 24 小時的程式比賽， 原本想說下午和隊友們討論一下之後，就開始寫 code， 沒想到一討論花掉了一整個下午，到了傍晚才開始動程式的部分， 到了晚上有了初步的版本，接著深夜繼續挑燈夜戰來改版； 不過漸漸的，體力越來越不支，最後只剩眼皮撐著，在 coding 方面幾乎無效益可言， 隔天早晨就算偷閒趴了幾十分鐘，精神還是沒有多好，只能設法修改先前的演算法，來求出最佳解。 就這樣持續的修修改改到了比賽終點，用功能未齊的最終版進行挑戰， 結果果然差強人意，沒有進入得獎名次； 雖然如此，但也是一次有趣的經驗， 除了在公司看日出外，更體會到睡眠的重要阿～</p>
Odie	<p>聽完題目時覺得好像沒有很難，但又不是很容易。然後我們這組感覺起來又好像比較弱， 所以一開始就盡量採取單純的作法，先求能跑出結果，在討論的時候，只要提出太複雜的想法一律砍掉。我們的想法其實還蠻容易的，就是從最大尺寸的盒面開始擺， 並盡量讓三種尺寸的板子的數量相同。為了讓實作更容易，我們一次只擺一種尺寸的板子， 而且也不考慮歪斜的擺法，讓盒面沿著鐵板的邊界一塊接著一塊。 (但最後看起來好像除了 Ansel 外也沒有人是擺斜的。)</p> <p>延續我們的想法，將工作分成三部分，一位負責寫輸入輸出，一位負責整個演算法流程， 一位負責將盒面擺在鐵板上的函式。因為 Phantom 很貼心地提供許多 API (雖然有很多都沒用到)， 所以寫起來還算順利。我們大概從四點多開始寫，八點左右把三個人的部分接起來， 不過可能是經驗不夠，接起來的地方有些 bug，大概又花了兩個小時 debug。很幸運地， 測試結果好像還可以 (至少和手算沒有相差太遠)，所以我們就很(ㄉㄨ)開(ㄉㄩ)心(ㄌ)的在十一點時回家了。 隔天早上我們又稍微調了一下程式，Jamei 和 Howard 兩人很努力地手刻了第五題的答案， 成為我們致勝的關鍵。</p> <p>然後是關於這次比賽我們這組比較驚險的地方。Phantom 事先給的三題測資， 我們在十分鐘之內都能跑完，但是在跑剩下的題目時，有好幾題在我們自己程式的某些參數設定下跑了非常久， 幸好我們另外使用比較安全的設定，才能在時限內跑完。之後參賽得更注意效率的問題， 畢竟結果再好，不能在時限內跑完也是沒用的。最後還是要感謝我的隊友們， 能和我一起堅持用最簡單的方式來解這個問題(XD)，希望以後還有機會一起參賽。</p>
Ansel	<p>這次一個人在日本出差 所以只好自己一個人一隊參加 缺點是第一個是沒有隊友可以討論 第二個是少很多小道消息 所以寫完 greedy 的基本版本以後就沒什麼方向 剩下就是調 performance 讓之後的 bruce force 在題目公布後可以跑更多組參數 調一調之後就不太知道要幹嘛了 第二天早早起來 修修改改又加了一些不同的參數 但是好像都沒什麼效果</p>

	<p>這次好像是有比上次比賽進步一點了 上次 coding 太複雜 最後連 bug 都抓不出來 這次寫的就很簡單 結果看起來沒有甚麼問題 只是感覺還抓不到比賽的訣竅....</p> <p>原本聽說會大量用到數學，三角函數，矩陣相乘，多元聯立方程式，幾何圖形..... 但是結果好像也沒用到 如果題目改成向量圖的話 可能就真的會用到了</p> <p>另外建議除了分享心得之外 也應該分享一下比較 detail 的演算法或是想法 不然也不知道有什麼差異</p> <p>沒有隊友可以謝 不過很謝謝過程中願意跟我討論的人和裁判</p>
David Wu	<p>第一次參加程式設計大賽， 很高興能與兩位經驗豐富的 RD 一隊， 在此次比賽中， 我的項獻實在不大， 希望以後有機會能加強這部份的能力， 以提昇戰力，幫助隊友。</p>
Howard Lee	<p>首先，這次的程式設計大賽要感謝 Odie 和 Jamei 的幫忙，讓剛進 Openfind 的自己，能夠分到一些獎金 XDDD</p> <p>在比賽前，自己其實是很期待又相當緊張，因為其實自己和別人一起寫程式的經驗不多，怕會拖累到隊友，但又在比賽前聽到題目跟數學有關，所以對數學還算喜歡的自己，其實還是相當躍躍欲試的。</p> <p>比賽當天，題目公布後，我們就先利用中午時間自己看看題目，下午起床後才開始討論，多虧 Jamei 預訂了會議室，讓我們有空間進行討論。</p> <p>在會議室中討論一兩個小時中，自己雖然提出一堆想法，但因為要在 24 小時內完成，似乎很多東西都會實作困難，所以被打槍了好幾次 XDDD。最後，我們決定先做出簡單能 work 的版本， 簡單來說就是以大中小順序開始擺，利用面積計算最大個數，訂出可以放的最大可能，完全不考慮斜的，利用以立和倒的方式直接看可不可以放，可以放就先放，做出固定排列下可能的最好結果。</p> <p>在方法訂定後，靠著簡單的設計，很清楚的切出三個人要做的部分，且銜接的變數和參數也都相當清楚訂定，這也是令我相當驚豔，簡單清楚的設計，讓我們之後三人程式兜在一起時沒有遇到太大的困難(不過還是有 BUG 啦 XDDDD)， 也讓我們可以順利回家睡覺，準備隔天再奮鬥。</p> <p>回家後，因為自己已經寫到有點亢奮的狀態，洗完澡腦袋都還是一直轉，</p>

	<p>沒想到也就這樣剛好發現到程式實作和設計似乎有些不同，所以躺上床後腦筋還是一直動， 也沒休息到多少就天亮了。</p> <p>隔天早上，沒想到三人都不約而同提早到，所以又有多餘時間修改、 做不同的版本和刻第五題(不過忘記刻第三題真是殘念...)，這邊真的要再提一下， 多虧 Odie 的修改版本(位移查找適合位置)，讓我們程式效果變好很多。</p> <p>跑結果的時候，我們三人各跑一個版本(位移大小和開始方向) 且分工跑各題(因有些位移設太小會造成速度較慢)，也順利在時間到前把結果跑出來 (其實有些緊張，因為第九題實在算太久了...)</p> <p>最後，比賽公布後一開始是殘念的第四名，不過因為其實也沒有得失心太重 (讓賭盤最後一名真是對不起他們兩位了.....)，但就是有點可惜了， 沒想到隔天看到名次異動，讓我們撿到第三名!!!</p> <p>其實加入 Openfind 時，自己就覺得公司有這個比賽真是太酷了!!! 而這次也多虧了兩位隊友，也讓自己學到很多合作和設計的觀念和方式。</p>
Jamie Hsu	<p>一聽完題目就覺得這是 NP complete 問題，試想辦法是否能用 DP 方法來求得最佳解。</p> <p>組員 Hunter 一開始就被困在新竹，等到快三點終於全員到齊 (與 Miyako)， 由於苦思無解，與大夥兒一起討論出一個簡易的演算法並分工。</p> <p>工作分為三部分 (演算法主幹、幾個 function、算數學與 QA)， 由於大家極度不想算數學且因為我出身為數學系，所以算數學與 QA 的工作較落在我身上了 (疑?!) 負責演算法主幹的是 Hunter，而 Miyako 負責幾個 function 撰寫。</p> <p>大家回座位努力完成各自的部分，而我真的開始看測資並開始手刻鐵板塊與產測資..... 現在想起來真的很不符合成本效益原則。</p> <p>九點左右，開始合併各自負責的部分，卻出現一堆問題，core dump、segmentation fault、 xy 弄反.....等， 可能發生的都發生了 (囧)</p> <p>好不容易完成預想中的演算法時，卻因效能太差而遲遲跑不出結果。 而 Hunter 在試著改善 performance 遇到了困難，直至十一點，我放下手邊的工作， 開始試著理解 Hunter 想的改善的方法。好不容易理解方法並且可以 work。 改善方法的確是可以讓結果快速產生，但卻會讓產生出來的鐵盒子數量大幅降低 (這不是 比賽的目標阿!!!</p> <p>似乎只有 Miyako 頭腦最清楚，我開始想回到一開始的版本並試著重新修改， 但非但沒有保留一開始的版本且拿到了合併版本 Hunter 版與 Miyako 版 XD 最後大家仍然無力回天，失敗原因大概有浪費太多時間算數學、缺乏一個好的領導、 想做的是 algo. A, 已改善效能的 algo. B, 最後卻做出 algo. C、努力目標與遊戲目標相 反、 版本控制方法錯誤，各自把對方寫的部分加到自己的 code 之外還自己修改成了自己的 版本、 沒有對每個版本做備份.....等。</p>

	<p>雖然結果不如一開始預期，但能跟 Miyako 與 Hunter 合作，獲得了難得的 team work 機會與經驗， 相信下次會更得心應手。</p> <p>感謝 Miyako & Hunter. :)</p>
Ian	<p>剛進 Openfind 的時候就一直聽過程式設計大賽的許多傳聞與故事， 不過因為進來的時間點好剛是上一屆結束不久， 所以過了快一年終於出現！ 我的隊友是 Joseph & David， 雖然聽說團隊分工合作通常不會有好下場， 但想說既然都分組了，就應該要團隊合作！ 原本以為應該不會討論太久， 沒想到這個題目看似普通， 真的要細想起來，也是挺複雜的， 經過熱烈的討論， 時間也來到晚上 7 點， 發現再不開工不行了！ 於是照著分工的項目， 大家各自下去奮鬥， 一轉眼來到 12 點， 終於要來整合了，真是緊張又刺激， 果然看起來就是 bug 百出， 跟前輩講得一樣，分工不是這麼容易的， 隨時時間愈晚，腦袋已經無法正常思考... 但為了對應 Phantom 為我們這組下的標語， 眼睛絕對不能闔上！ 時間來到早上 5 點， 經過探子的回報，發現好像每一組成績差不多， 於是我們決定加入旋轉！ 想像著每題只要被我們多矇 1、2 個，就勝券在握了！ 經過了數個小時的努力， 發現不知道是數學太差還是腦袋快當機， 跑出來的結果都不像我們所想像... 終於，各組要來比結果了！ 原本以為會被其它人慘電， 沒想到原來真的大家都差不多， 多 1 個，可能就會多差一個名次！ 而且大家好像也都沒有旋轉， 唯一有旋轉的帥哥 - Ansel，抓到了 Phantom 工具的 bug，超強！XDD 一天真的好快喔~ 下次還是走效率路線好了，回家睡個覺起床再戰！</p>
Phantom	<p>這次的題目是從 NP-Complete 的歷史名題 Knapsack 所演變出來的， 題目的原型由書安所提供（他過去接案子所遇到的實際案例），本來想拿來當成 Senior Engineer Training 的題目，但是被我拿來用在程式設計大賽上。</p> <p>題目其實不難理解，但是要做到 perfect 幾乎不可能。為了怕電梯那年的慘劇再現， 依照往例我照樣寫了一堆 API 給參賽者呼叫，結果這些 API 一寫寫了四個晚上，還把</p>

三角函數、圖學的矩陣相乘、演算法的線段相交、演算法的 point、line、polygon 等等幾乎重新複習了一輪，連 Convex Hull 都看了，只差沒用到 Voronoi Diagram (註 1)

出題上我跟書安各出了幾題，還有一些很變態的題目被書安所勸阻，不然各位拿到的 10 個考題就不是這樣的了。@_@

但是很不幸的 API 還是出了問題，在算斜率的近似值上我用的規則太寬鬆了，導致旋準的矩形跟點之間的關係判定會出錯。這邊要感謝 Ansel 的體諒。

說真的這題我其實也想不到什麼好解法，要是想得到好解法證明 $P=NP$ ，早就可以去拿 Turing Award 了。

(註 1:) Voronoi Diagram 是我大三的演算法最後一個作業，當年的聖誕夜全班 50 個人有將近 30 個是在系館的 PC lab 度過的。演算法的陳文進老師一點多還跑來系館看我們，

看到很多人後才很滿足的離開。後來清晨四點大家決定去唱歌，聽到價格要 double，馬上有人罵出口 => 「我已經用了好幾天的 double，不想再聽到這個字。」

當場全部人解散回家睡覺。