**Programming Assignment 2 Report**

**ESOE**

**B05505053**

**柯哲邦**

1. **演算法流程 (Algorithm Flow) <解釋程式的運作>**

一共寫了3個函式：

1. Main：主程式，負責輸入輸出檔案，並將最後的結果以頂點之i點(i<j)由小到大排序。
2. MIS\_length：用來填表，將目前紀錄到的總弦數記錄下來。以及將k(也就是數字j連到的頂點)記錄下來。
3. trace\_k：以遞迴的方式將可以達成最多點的每一條線記錄下來。

(以下i,j 表示頂點之第一點及第二點，k表為與j之連線)

Main:

先將輸入的第一個值，也就是總列數讀進來後紀錄起來，並開啟一個以總列數為大小的array。此array用來記錄頂點的數字，且頂點都充當index以及value的角色ex: 1 3 的存放方式為record\_chord[1] = 3, record\_chord[3]=1，且另外以一個vector記錄所有頂點的i值。因為在答案回傳時，可能會回傳頂點的第二點，所以需要用一個vector將答案一致改為以i值表達。將答案一個個與所有i值比對，若沒有找到，代表答案一定是回傳j值(因為答案已確定為題目所給的弦)，再將其翻轉即可。最後將結果輸出至指定檔案中。

MIS\_length:

二維指標m為存放當下所可以包含得最多弦數。二維陣列record\_k則是用來填當下最多弦數的最後一條弦。沒有開一個完整的方陣而是直角三角形，因為j >= i 才會有意義。m[i][j] = m[i][j-1]; 先將表中每一點預設為上一列的值(因為上一點可能總弦數較多)。將題目以j之連線k分成二種情況進行填表：

1. i < k < j

如果上一列的值小於1 + m[i][k-1] + m[k+1][j-1]，就把該點存為後者。記得檢查上一列的點是否小於此值，這樣取的才會是到目前為止的最大值。k如果 = j – 1 也會需要一點些微調整，因為[k+1][j-1]會導致前者大於後者。且要以record\_k記錄該k值。

1. k = i

一樣如上一情形，也要先檢查上一列的值是否小於m[i+1][j-1] + 1，條件成立才會將該點存為後者，一樣要將該k點記錄。

trace\_k:

用遞迴的方式去trace剛剛記錄好的k值，不斷以上一function內提到的case去trace當中的弦。並將所有k值存放在一個vector中回傳。此vector只會存放一個頂點的其中一點，可能是i值可能是j值，到main當中再統一轉成i值後將其排列後與j值一併印出。

1. **資料結構 (Data Structure) <解釋”特別的”資料結構>**

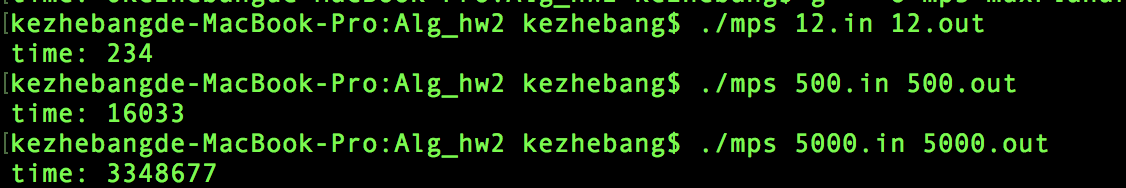
本次實作中沒有用到特殊的資料結構，都是用陣列和vector去存放所要記錄的弦。

1. **問題與討論 (Discussion) <討論實作中遇到的問題及疑問>**
2. 在記錄表格時，出現記錄變少的情況。也就是說，可能0,10之中已經記錄3條弦，可能0,11中只有2條。這種情況就是沒有檢查case中的情況得到的結果是否有大於前一列的值，若無，應該存的是上一列，也就是比較大的值。
3. 將k值trace回去放入vector當中時，存放的不一定是原來input中的第一項或第二項。如此輸出格式會錯誤。我的解決方法較為笨，我是開一個vector存放所有頂點中的第一項，並以k值iterate過這個vector。如果出現一樣的就代表此k值是頂點中的第一項，若沒有出現就代表k值為頂點中的第二項。
4. 將k值記錄下來，在trace回去的過程可能會記錄到錯的點，因為原本將record\_k這個表格初始化為每一個都為0，在trace的過程可能會導致0這個點取到的點會是錯的。
5. **時間分析**

本實作都運用兩個2個for迴圈來進行，推測時間複雜度應為O(n2)。

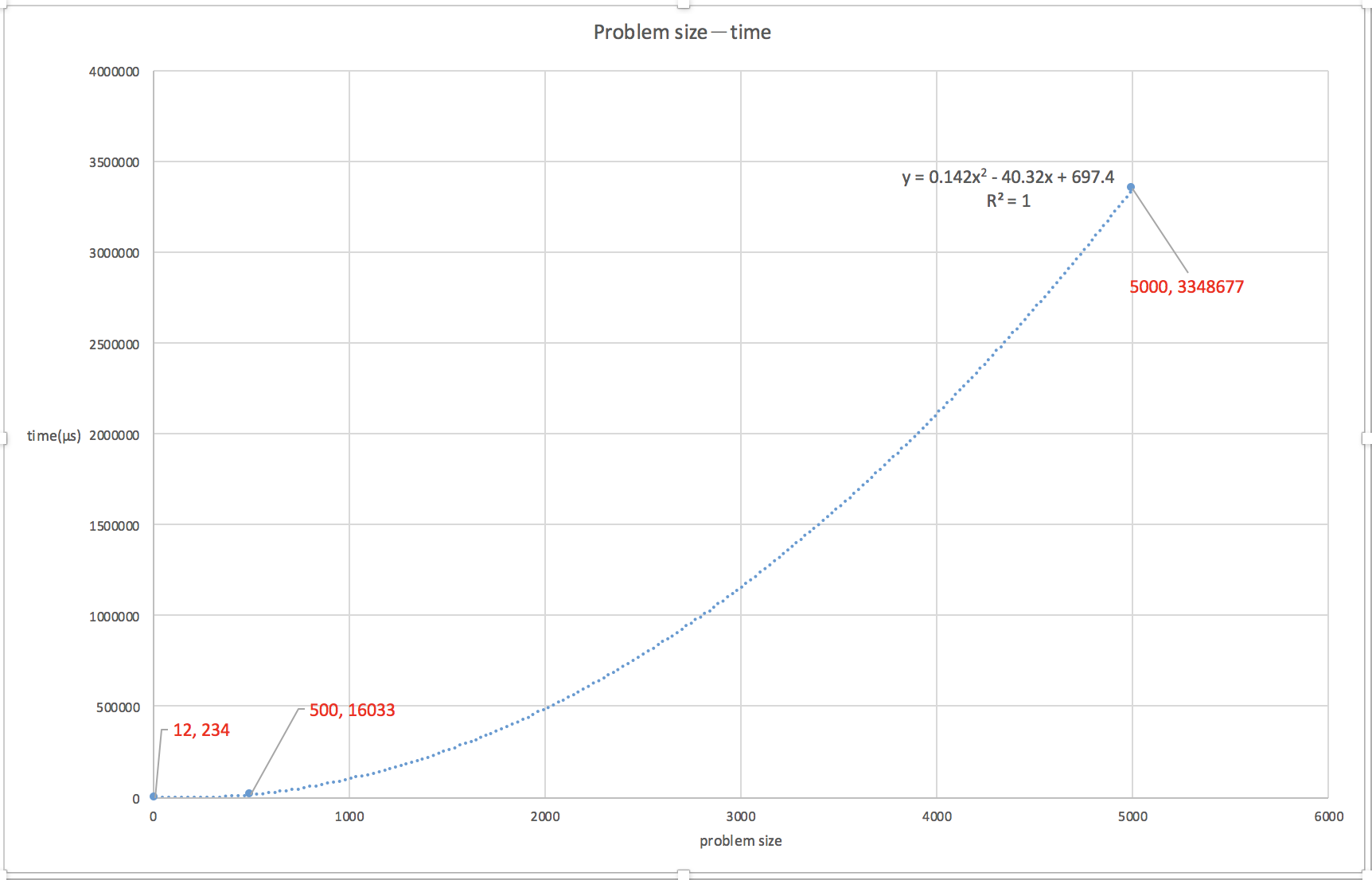
在程式中運用clock()來計算時間上的經過。運用題目所給之input來進行量測。

（本量測只截取中間實作DP的部分，沒有囊括讀檔的部分。）



附圖為運用此得到的時間（單位µs（以CLOCK\_PER\_SEC得1000000））

運用此數據畫出趨勢圖：



以此圖可以很明顯看出，此圖非常符合O(n2)的預測，R2達到1。

**Empirical Order of Growth:**

t ~ kna

* t: execution time
* k: coefficient not important
* n: input size
* a: coefficient we want to find

We can use log-log plot to measure “a” by using least square method in EXCEL.

|  |  |  |
| --- | --- | --- |
| Original value | | |
| Input size | Runtime (µs) | Memory (B) |
| 12 | 234 | 320000 |
| 500 | 16033 | 2372000 |
| 5000 | 3348677 | 16000000 |

|  |  |  |
| --- | --- | --- |
| Value after log operation | | |
| Input size | Runtime (µs) | Memory (B) |
| 3.584962501 | 7.87036472 | 18.28771238 |
| 8.965784285 | 13.96875678 | 21.17767258 |
| 12.28771238 | 21.6751598 | 23.93156857 |

Time complexity: O(n1.5429) Space Complexity: O(n0.6378)

