

# 演算法 作業二 報告

b03208022 李友岐

(1) 12.in 之結果: (也可見 12.output)

```
[Jamesde-Mac-mini:001 JamesLee$ g++ -o bin main.cpp  
[Jamesde-Mac-mini:001 JamesLee$ ./bin 12.in 12.output  
3  
0 4  
5 7  
8 11
```

(2) 500.in 之結果: (也可見 500.output)

```
[Jamesde-Mac-mini:001 JamesLee$ ./bin 500.in 500.output  
34  
19 474  
32 465  
41 363  
51 83  
78 66  
91 359  
114 136  
115 126  
138 139  
161 166  
174 346  
183 203  
218 271  
240 225  
253 250  
268 244  
290 277  
307 296  
313 294  
318 215  
321 322  
325 206  
329 176  
333 344  
353 100  
356 98  
360 86  
367 33  
377 413  
380 385  
459 418  
464 417  
468 24  
479 478
```

(3) 5000.in 之結果:

```
[Jamesde-Mac-mini:001 JamesLee$ ./bin 5000.in
```

(4) 程式說明:

我的程式主要是依照 **matrix chain multiplication** 的演算法來修改

一樣先建造 **m** 陣列和 **s** 陣列，之後再藉著幾個 **for loop**，

一層一層將 **m** 陣列和 **s** 陣列完成，再運用遞迴將弦找出來。

程式的空間複雜度為  $O(n^2)$ ，時間複雜度為  $O(i*n^3)$ ，

**n** 為點的數量，**i** 為弦的數量。

當 **n** 為 12 和 500 時，程式皆能跑出正確的結果。

當 **n** 變成 5000 時，建造 **m** 和 **s** 陣列所需的空間太大，所以記憶體直接不足。

因此我改用動態配置記憶體，也成功建造了 **m** 和 **s** 陣列。

但是由於時間複雜度高達  $O(i*n^3)$ ，所以等了一陣子程式依舊跑不出結果，

應該是卡在逐層填滿 **m** 陣列和 **s** 陣列那邊。

希望下次我能想出時間複雜度更小的寫法。