**Programming Assignment#2 Report**

**NTUEE**

**B04501095**

**黃平瑋**

1. **演算法流程 (Algorithm Flow)**

以**MPS(i, j)**,表示點i 到點j 間的最大不相交弦數目

**V\_ij** 表示連接點i和點j之間的弦

**Trace(i, j)** 在backtrace時使用

* Recurrence:

MPS(i, j) = MPS(i, j-1)

(if k not in [i, j] for v\_jk or v\_kj)

= max( MPS(1, j-1)**,** MPS(i, k-1) + 1 + MPS(k+1, j-1) )

(if k in [i, j] for v\_jk or v\_kj)

* Pseudo code:

for j = 0 to n - 1 do:

k = the point that V\_jk or V\_kj is in the set

for i = 0 to j - 1 do:

if i <= k <= j-1 and MPS(i, k-1) + 1 + MPS(k+1, j-1) > MPS(1, j-1):

MPS(i, j) = MPS(i, k-1) + 1 + MPS(k+1, j-1)

Trace(i, j).left = Trace(i, k-1)

Trace(i, j).right = Trace(k+1, j-1)

else:

MPS(i, j) = MPS(i, j-1)

Trace(i, j).left = Trace(i, j-1)

MPS(i, n-1) 即為最大不相交弦的數目

* Finding all chords

由Trace(0, n-1)開始，用in order的方式traverse，若右邊的node 不為 null，則輸出

right->i-1 + ' ' + right->j+1

1. **資料結構 (Data Structure)**

這份程式我共使用的三個檔案實做 main.cpp maxPlanarSubset.h maxPlanarSubset.cpp

在maxPlanarSubset中 我宣告了class **MPS** 來處理這個問題 

* data memeber:

**\_table:** 紀錄MPS(i, j)的值

**\_trace:** 紀錄要連結到哪些子問題

**\_coord:** 給定一個點，若為某一弦之端點，回傳另一端之端點

* class function:

**parse:** 將input的檔案中含有所有chord 的set轉換成unordered\_map的形式

**assign:**利用dynamic programming 的方式將\_table的值填入

**traverse:**從MPS(0, n-1)開始，找出所有的max planar subset 中的 chords

**output:**輸出結果到檔案

1. **問題與討論 (Discussion)**

在這次的程式實做上沒有遇到什麼太大的問題，只有在一些比較特別的case要謹慎處

理。比如說剛好有一個弦V\_ij在[i, j]中，如果i = 0 時，依照前面的規則會變成要求 MPS(0, -1)

MPS(1, j-1)就會造成問題，需額外處理。

下表是不同test case的結果與時間

|  |  |  |
| --- | --- | --- |
| Number of nodes | Number of maximum chords | Runtime |
| 12 | 3 | 0.003 |
| 500 | 34 | 0.020 |
| 5000 | 120 | 0.802 |