

EDA union lab – 40056

1. 為了找出 maximum planer subset，我使用了 dynamic programming 這個 algorithm，而我使用的是 bottom-up。Recurrence equation 如下：

$$M[i][j] = \begin{cases} M[i][j-1], & \text{if } k \notin [i, j] \\ \max(M[i][j-1], M[k+1, j-1] + 1 + M[i][k-1]), & \text{if } k \in (i, j) \\ M[i+1][j-1] + 1, & \text{if } k = i \end{cases}$$

其中  $k, j \in C$ 。

這樣我們只要輸出  $M[0][n-1]$  即可知道 maximum planer subset 中有多少條 chords。這部分我參考了 hw2 的 Q10。

至於要如何知道是由哪些 chords 組成，我是使用 top-down 的想法，搭配遞迴的方式，分析  $M$ 。如果  $M[i][j] = M[i][j-1]$ ，那我們就去看  $M[i][j-1]$ 。如果  $M[i][j] = M[k+1, j-1] + 1 + M[i][k-1]$ ，那我們就把  $j$  放進 results，然後看  $M[k+1, j-1]$  跟  $M[i][k-1]$ 。如果  $M[i][j] = M[i+1][j-1] + 1$ ，同樣把  $j$  放進 results，然後看  $M[i+1][j-1]$ 。就這樣跑完後我們的 results 裡面就有 maximum planer subset 的所有 chords 的一個端點。

最後因為輸出的點要 sort 過，所以我使用了 merge sort，因為在 pa1 中我們發現他的綜合表現是相對好的。

2. 我們主要需要分析三個步驟的 time complexity。Maximum planer subset 需要  $O(n^2)$  time，得到 maximum planer subset 所組成的 chords 需要  $O(n)$  time，最後 merge sort 需要  $O(n \lg n)$  time。總的來說，total time complexity 為  $O(n^2)$ 。