

我使用一個 class，裏面有 constructor, destructor, MPSnum(用來計算 Max Planar Subset 的 chords 數量，採用 top-down 的方法), MPSsol(用來計算 Max Planar Subset 的 chords 是哪幾條，採用 top-down 的方法),還有一個一維 vector path_node(在 MPSsol 中把需要被儲存的 chord 的 node push_back 到 path_node 裏面)，最後在 main.cpp 中再寫入 output file 中。

另外還有一個在 main.cpp 中定義的一維動態陣列 chord_table，用來儲存 input 中每條 chord 的對應關係，在 class MPS 中有兩個 private 變數，其中一個是二維指標 M，用來作為二維動態陣列，當作 dp 中的 memorization 使用，另一個是 NN，則是在 class MPS 的 constructor 中把 NN 設為此次 input 的 N，也就是輸入的 node 數。

此次演算法採用 Top-down 的形式，相較 bottom-up 的方法而言，大幅降低計算時間，100000 的 case 也有在 10 分鐘內完成。