Programming Assignment #2
Maximum Planar Subset

B11901164 陳秉緯

1. Preprocessing

Transform each input data into a pair and store them into a vector. For example, 0 4 -> (0, 4), 1 9 -> (1, 9), and so on.

Sample Input
12
0 4
1 9
2 6
3 10
5 7
8 11
0

Storing both directions enables us to get the other endpoint from one known endpoint without additional computations.

2. Construct Tables

A function called ConstructTable is the core of the algorithm. Table M stores the maximum chords and Table C stores what case it is. And call the TraceBack function to trace back the result with Table C.

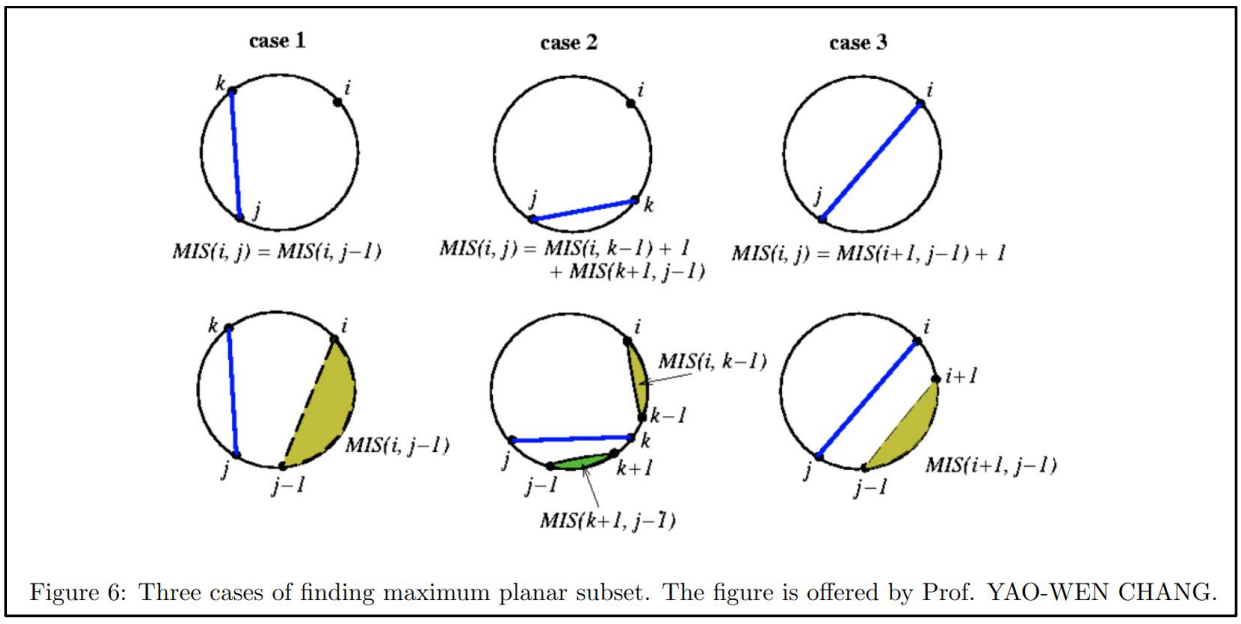

Figure 6: Three cases of finding maximum planar subset. The figure is offered by Prof. YAO-WEN CHANG.

3. Trace Back

A function called TraceBack recursively calls itself to save the start point of the chord into the result.

4. Time Complexity

In the function ConstructTable, we have to fill up 2 n by n tables M and C, so it costs O(n^2)-time

/* time dominant part

```
for (int d = 1; d < n; d++) {
   for (int i = 0; i < n - d; i++) {
     int j = i + d;
     int k = data[j].second;
     if (k == i) { // chord (i, j) is a part of the maximum planar subset
        M[i][j] = M[i + 1][j - 1] + 1;
        C[i][j] = 2;
     } else if (k > i &&
        k <= j) { // chord (i, j) intersects with some other chord
        int temp = M[i][k - 1] + 1 + M[k + 1][j - 1];
           if (M[i][j - 1] <= temp) {
              M[i][j] = temp;
              C[i][j] = 3;
           } else {
              M[i][j] = M[i][j - 1];
           }
     } else { // chord (i, j) does not intersect with any other chord
         M[i][j] = M[i][j - 1];
     }
   }
}
*/
```

5. Result of public test cases

Run on EDA union lab machines

Input size	CPU time (ms)	Memory (KB)
12.in	0.139	5912
1000.in	10.579	8948
10000.in	1075.92	299852
100000.in	272498	29355992