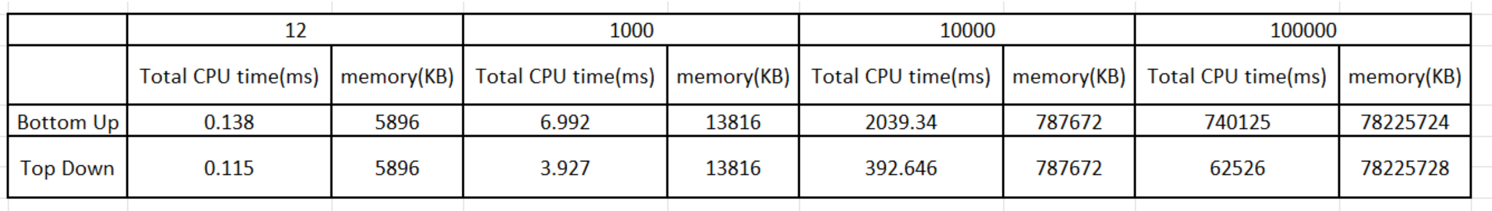
姓名：金家逸　　　學號：B10502076　　　系級：電機三

PA2 report

Running time for each case:



	12		1000		10000		100000	
	Total CPU time(ms)	memory(KB)	Total CPU time(ms)	memory(KB)	Total CPU time(ms)	memory(KB)	Total CPU time(ms)	memory(KB)
Bottom Up	0.138	5896	6.992	13816	2039.34	787672	740125	78225724
Top Down	0.115	5896	3.927	13816	392.646	787672	62526	78225728

1. Data structures: (all of the following are implemented using dynamic arrays)

(1) chords:

a one-dimensional array with a size of 2n, used to record the connectivity between each vertex on the circle.

(2) MPS:

a two-dimensional array with a size of 2n x 2n, which is used to record the maximum planar subset of M(i, j).

(3) CASE:

a two-dimensional array with a size of 2n x 2n, used to trace which chords compose the maximum planar subset. Every element in CASE[i][j] denotes the case in which M(i, j) is obtained.

(4) max_chords:

a one-dimensional array with a size of 2n, where each element denotes whether the chord is in the maximum planar subset. If the chord is in the subset, then the two points of the chord are marked as true; otherwise, they are marked as false.

2. Subproblems and recurrence formulas:

Define the point to which j is connected as point k.

(1) **Case 1: if k is equal to i**

M(i, j) = M(i+1, j-1) + 1

(2) **Case 2 and Case 3: if k is in the range i < k < j**

Case 2 : M(i, k-1) + M(k+1, j-1) + 1 is larger, then M(i, j) = M(i, k-1) + M(k+1, j-1) + 1

Case 3 : M(i, j-1) is larger, then M(i, j) = M(i, j-1)

(3) **Case 4: if k is out of the range i~j, that is k > j or k < i**

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

3. Time complexity analysis:

For the Top-Down method, although we need to fill the tables for MPS and CASE, not all elements are required, hence the time complexity is O(n^2), which is better than the Bottom-Up method.

4. Space complexity analysis:

(1) For chords and max_chords, which have an array size of 2n: space complexity is Θ(n).

(2) For MPS and CASE, which have an array size of 2n x 2n: space complexity is Θ(n^2).

⇨ The total space complexity is Θ(n^2) for Top-Down methods.