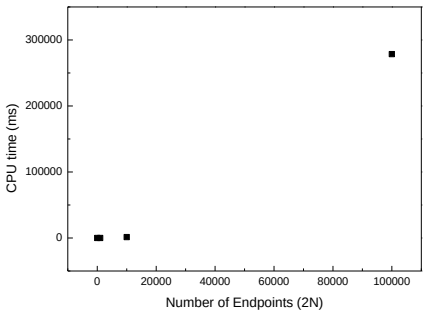
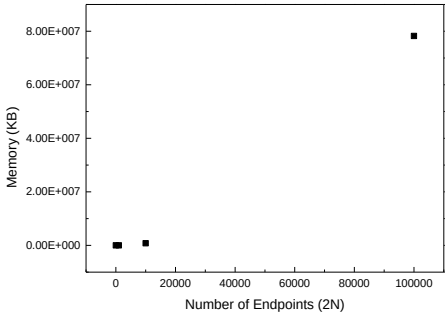
PA2 Homework 陳韋辰 r08222017

1. Complexity Analysis

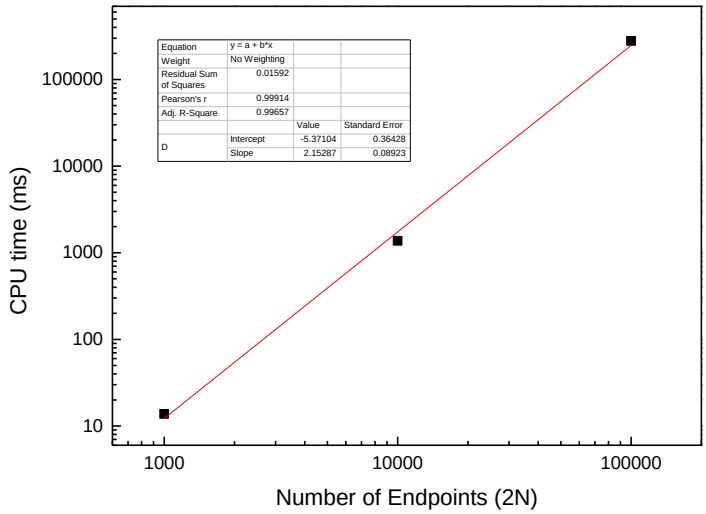
2N	The total CPU time (ms)	Memory (KB)
12	0.593	5892
1000	13.775	13812
10000	1372.19	788092
100000	278506	78229048

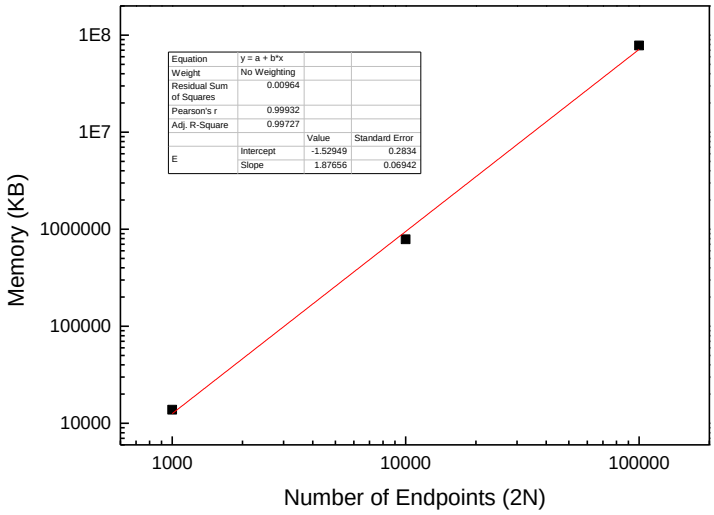
下圖左為 data size 跟運算時間的關係圖，下圖右為記憶體用量跟運算時間的關係圖：





將 1000、10000 和 100000 的 case 取對數並用線性 fitting，如下圖：





可以發現所花的時間(~2.15)跟記憶體用量(~1.88)皆跟 data size 大約成二次關係，符合時間複雜度 $O(N^2)$和空間複雜度 $O(N^2)$的預期，而 case 12 大部分的運算不在算 mps，所以時間跟記憶體用量會比趨勢線預期高。

2. Data Structure and Algorithm

使用 bottom-up 的 DP

(1) 將 input data 存入 data array 中，每個 index 對應的值，都是他 index 對應的 chord 另一端，所以 data 總共有 2N 個

Time complexity: O(N)

(2) 找 mps 中 chords 的數量：用 HW2 problem8 的方法，M 跟 N 是 2D-array，其 index 代表所選 endpoints 的區間，M 是紀錄 mps 的值，N 是分出三個 case。把 mps 存入 M[i,j]中；同時間，如果遇到 k=i 那 N[i,j]設為 1，如果 k 在[i,j]中，且要選，則設 N[i,j]為 2，其他為 0。此時 M[0][2N-1]即為 chord 的最大數量

Time complexity: $O(N^2)$

(3) 找 mps 被選到的 chords：從 N[0,2N-1]開始，如果 N[i,j]=0 就 j=j-1；如果 N[i,j]=1 就記錄當下的 k 到 Choosed_Chord 的 array 中然後 j=j-1；如果 N[i,j]=2 就記錄當下的 k 到 Choosed_Chord 中然後再從 N[i,k-1]和 N[k+1,j-1]中尋找其他的 Choosed_Chord，直到 i<j

Time complexity: O(N)

(4) 將找到的 Choosed_Chord 用 heap sort 排好，並用 data array 找對應的 chord 另一端

Time complexity: Worst case(所有 chords 不重疊): O(NlgN) Best case(只選一條 chord): O(1)

Overall time complexity: $O(N^2)$