

# 資料結構hw1 報告

For a linked list of two-dimensional points (i.e. (x, y)) create a user-friendly, menu-driven program

that performs the following operations. Please discuss the complexity for each operation.

- (a) Create a list of points.
- (b) Determine if a list is empty.
- (c) Insert a point at the end of a list.
- (d) Search for a point.
- (e) Delete a point.
- (f) Print out a list.
- (g) Count the number of points in a list.

## 功能實做說明：

- (a) create的複雜性不高，他只會開在開頭位置malloc一次。
- (b) 是否為空LIST也不複雜，他也只會檢查開頭位置是否為NULL，是則為空LIST。
- (c) 在LIST最後面insert一個點，需要先從開頭位置進入，依序檢查每一個node是否為NULL。檢查到NULL則為end，然後就在end of list 插入一個點。
- (d) search 的部分會從開頭依序尋找，找到最後一個位置為止，但search功能只會找到第一個點，然後回傳是否找到那個點。如果你的LIST上面有超過兩個一樣的點，他只會找到最靠近開頭的點。
- (e) delete一樣也是從頭開始找，如果你相同的點超過兩個，也只會找到第一個要刪除的點而進行刪除。
- (f) 從頭到尾把一整個LIST顯示出來。
- (g) 從頭到尾計算LIST的長度

## 複雜度：

假設LIST長度為n，step為一個步驟。

- (a) 只會做一次malloc。 1 step.
- (b) 只會檢查開頭是否為NULL。 1 step.
- (c) 會檢查到end再insert。 檢查為n steps，insert為1 step，全部 n+1 steps。
- (d) search 最差的情況就是檢查到最後沒有結果，n step。  
否則"檢查的次數 k次"就是step的次數。k step。
- (e) delete 最差的情況就是檢查到最後沒有要刪除的點，n step。  
否則就是"檢查的次數 k次"加上"刪除的步驟1次"。k+1 step。
- (f) 顯示的話就是每個位置都訪問，每個位置都print。2n steps。
- (g) 計數的話就是每個位置訪問，然後計數器每次加一，最後再print出次數。2n+1 steps。  
訪問n次，計數n次，print1次。

操作教學：<https://github.com/yoxisem544/datastructurehw1>  
在git上面。