

系級：機械四 姓名：鄭承昀 學號：B03502106

一、資料結構的實作

1. 三種資料型態

I. Double Linked List :

Double linked list 是將每一份資料分別儲存在一個 node 中，跟 Single linked list 的差別在於每一個 node 都擁有兩個 pointer，分別指向他前面和後面的 node。也正是因為有兩個 pointer，期解決了 Single linked list 在 pop_back 的速度問題($O(n)$ 變 $O(1)$)。但是在 find 方面，因為要從頭開始找，較花時間($O(n)$)。

II. Array

將資料存在一連續記憶體中，當記憶體大小不夠時能重新拿取一個較大的連續記憶體，並將原本的資料複製過去。因為是連續存取的關係，array 支援 random access。再刪除資料時，array 必須做平移後續資料的動作($O(n)$)。

III. Binary Search Tree

BST 一樣是將每一份資料分別存在不同的 node 中，每一個 node 都擁有兩個 pointer，分別指向其右邊的子 node 和左邊的子 node。另外，在儲存資料時有一規則，左邊的子 node 的值小於等於其 parent 的值，相反的右邊子 node 的值大於其 parent 的值。正因為這個特性，BST 在搜尋時只需要 $O(\log n)$ ，且其資料已經經過排序。但是在 insert 方面卻必須花 $\log(n)$ 的時間，比較慢。

2. 使用程式實現

I. Double Linked List

分別使用兩個 class(DList 和 DListNode)。DListNode 裏面包含了_data 用來存取資料，兩個 pointer _next 和 _prev 分別指向其前後的 node。而 DList 裡面包含一個 DListNode 作為_head，還有其他會使用到的 function。另外，為了使用 iterator，加入了一個 dummy node，當作整個 list 的 end。

II. Array

其中包含了三個 data member，_data 為正在使用的連續記憶體，_size 為存進的資料數，_capacity 則代表現在使用的記憶體做多可儲存的量。每當記憶體空間不夠時，就會重新要一組記憶體，其大小為 $_capacity * 2$ 。另外，array 的 iterator 多了+和+=，可提供 random access 的功能。

III. Binary Search Tree

使用兩個 class(BSTree 和 BSTreeNode)。BSTreeNode 中除了儲存資料外，還包含了三的 pointer 分別指向_parent，_left，

_right，其中的_parent 的功能為方便 iterator 中++及-的撰寫。
而在 BSTree 中包含兩個 data member，分別為_root 和_size。
_root 為整個 BST 中的 root，在沒有任何資料時依然存在。而_size
則負責記錄總共有多少筆資料。

3. 實作優缺點

I. Double Linked List

加入_prev 使得在 push_back 時只需花 $O(1)$ 的時間，而加入 dummy node 則方便 iterator 的處理。

II. Array

其優點在於其記憶體大小可以隨需求做改變，在這次實作中只有當空間不夠時才會將記憶體空間變大，在資料被刪除後，並部會因為不需要這麼多記憶體而減少記憶體空間，可會發生浪費記憶體的情況。

III. Binary Search Tree

其中我加入了_parent，這個在實作過程中比較好操作，可以快速知道每一個 node 之間的關係。但是在當資料數擴大的時候，_parent 就會多占用較多記憶體空間，也會使執行速度變慢。

二、實驗比較

1. 實驗設計

Insert 和 sort 的比較。先 random 產生 50000 個資料和 100000 比資料，再進行排列，分別記錄其執行時間和記憶體使用量。

2. 實驗預期

I. 在 insert 方面，List 和 array 都是 $O(1)$ ，而 BST 則為 $O(\log n)$ 。

所以 List 和 array 應該時間相近，BST 最慢。

II. 在 sort 方面，BST 因為已經算是排列好了，所以應該是做快的。

而 array 使用的 quicksort($O(n \log n)$) 應該會比 list 使用得 bubblesort($O(n^2)$) 快。

而在記憶體使用量上則應該為 $BST > List > Array$ ，因為 BST 含有四個 data member，List 三個。

3. 結果比較和討論

I. 在執行時間上

50000

	insert	Sort
Dlist	0 second	23.04 second
Array	0.01 second	0.01 second
BST	0.02 second	0 second

100000

	insert	Sort
Dlist	0 second	91.58 second
Array	0.02 second	0.02 second
BST	0.05 second	0 second

跟預期的一樣，在 insert 時 BST 最慢，Dlist 跟 array 差不多。而在 sort 時則是 BST 最快，而 Bubblesort 跟 quicksort 的差異極大。

II. 在記憶體使用上

50000

Dlist	2.988M
Array	2.863M
BST	2.945M

100000

Dlist	6.004M
Array	5.875M
BST	6.012M

這裡跟預期的也差不多 BST 多了一點，而 array 最少。