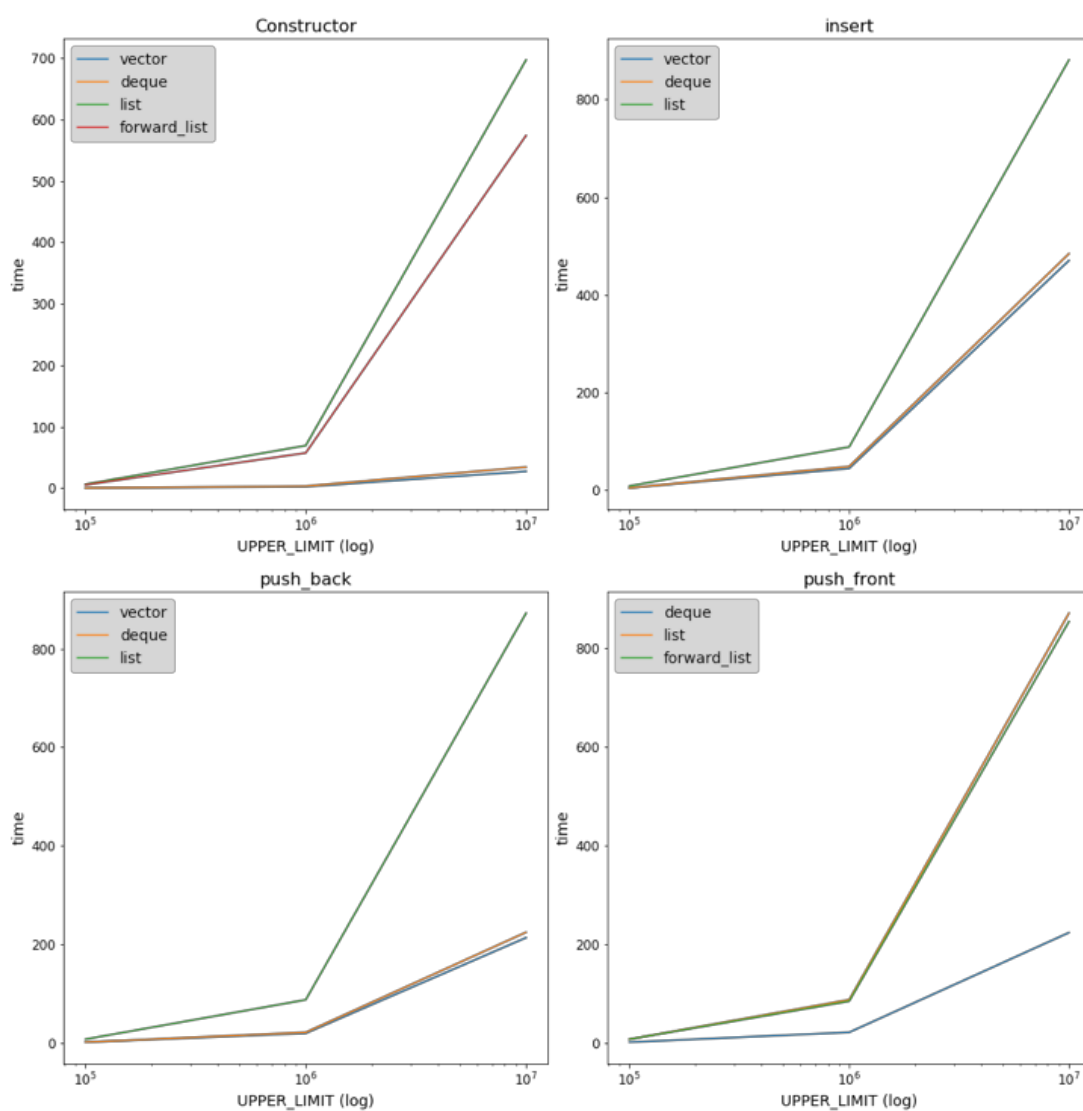


## HW9 Report

R07521603 蔡松霖

由圖一可以觀察到，當 UPPER\_LIMIT 越大時，不同容器執行上所需花費時間的差異越趨明顯。這表示當我們處理較大的程式時，用對的容器以及執行方法，可以使我們的程式更有效率。



圖一 數量級所需時間比較圖

圖二中的每個數據均是運行十次取平均的結果，其中-1項代表該容器沒有

支援對應的 insert 方法。總的來看，在 UPPER\_LIMIT 為 10,000,000 的設定下，執行容器的初始化 constructor，使用 vector 最快，其次是 deque，剩下兩個 forward\_list 及 list 則慢上許多，兩個都超過 vector 所需時間的 20 倍。比較 vector 執行 push\_back 以及 insert 所需的時間，我們可以觀察到當在 vector 最尾端 insert 新元素的時候耗時最少。同樣觀察 deque 的結果，也可以推得同樣類似的結果，在最頭或最尾端 insert 的時候最有效率。List 比起 forward\_list 要建立雙向的關係，因此在執行 constructor 上就慢上許多。

### Comparison between sequential containers ###				
	constructor	push_back	insert	push_front
vector	25.9359	211.701	469.664	-1
deque	34.7105	225.833	485.191	225.24
list	702.84	867.926	932.188	871.956
forward_list	537.308	-1	-1	810.04

圖 二 綜合比較表格(UPPER\_LIMIT: 10,000,000; 10 cases per test)