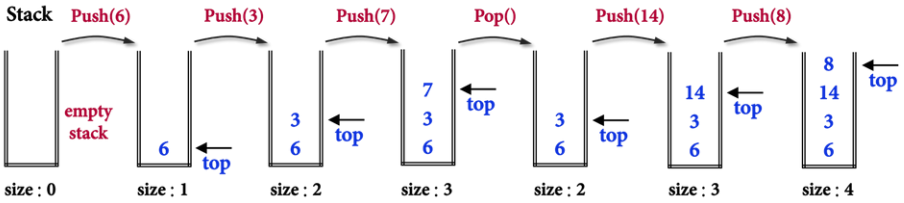
Stack 說明

Stack 是具有「Last-In-First-Out」的資料結構(可以想像成一種裝資料的容器),「最晚進入 Stack」的資料會「最先被取出」,「最早進入 Stack」的資料則「最晚被取出」。

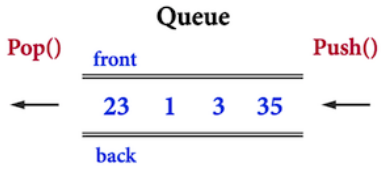
一般的 Stack，會有以下幾個處理資料結構的功能：
Push(data)：把資料放進 Stack。
把書放進箱子。
Pop：把「最上面」的資料從 Stack 中移除。
把位於箱子「最上面」的書拿出來。
Top：回傳「最上面」的資料，不影響資料結構本身。
確認目前位於箱子「最上面」的是哪本書。
IsEmpty：確認 Stack 裡是否有資料，不影響資料結構本身。
確認箱子裡面有沒有書。
getSize：回傳 Stack 裡的資料個數，不影響資料結構本身。
記錄目前箱子已經裝了多少本書。



Queue 說明

Queue 是具有「First-In-First-Out」的資料結構，如同排隊買車票的隊伍即可視為 Queue，先進入隊伍的人，可以優先離開隊伍，走向售票窗口買票，而後到的人，就需要等隊伍前面的人都買完票後才能買。

如同普遍認知的排隊隊伍，Queue 也具有以下特徵：
隊伍有前方(以 front 表示)以及後方(以 back 表示)之分。
若要進入隊伍(Push)，一定是從 back 進入。
若要離開隊伍(Pop)，一定是從 front 離開。
以圖一為例，由 front(隊伍前方)和 back(隊伍後方)可以判斷，進入隊伍的順序應該是 23、1、3、35。



一般的 Queue，會有以下幾個處理資料結構的功能，配合圖二：
Push(data)：把資料從 Queue 的「後面」放進 Queue，並更新成新的 back。
在 Queue 中新增資料又稱為 enqueue。
Pop：把 front 所指向的資料從 Queue 中移除，並更新 front。
從 Queue 刪除資料又稱為 dequeue。
getFront：回傳 front 所指向的資料。
getBack：回傳 back 所指向的資料。
IsEmpty：確認 Queue 裡是否有資料。
getSize：回傳 Queue 裡的資料個數。

參考資料

1. http://alrightchiu.github.io/SecondRound/stack-introjian-jie.html
2. http://alrightchiu.github.io/SecondRound/queue-introjian-jie-bing-yi-linked-lists hi-zuo.html