

11027127 林肇昌

ch5 堆疊

線性結構

LIFO (最後進, 最先出)

ADT List {

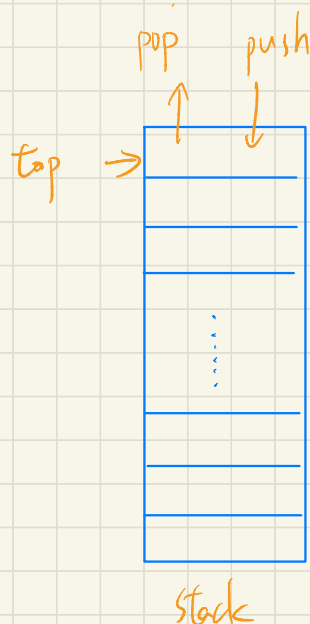
push (item); 添加元素

pop (top); 移除最上層

top (top); 找到最上層

isEmpty (); 檢查是否為空

} // stack



實例: 迴文判斷、括號配對、河內塔

ch6 佇列

與 stack 相反 (FIFO) 先進先出

API List {

enqueue(item); 添加元素

dequeue(); 移除 front

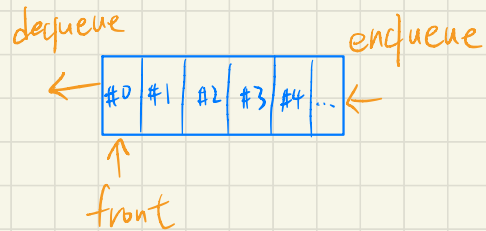
front(); 找到 front

isEmpty(); 檢查是否為空

} // Queue

實例

排隊、接訂單



stack vs Queue
LIFO FIFO

兩者差異在資料處理上，
但兩者可以互相實現、轉換

ch1 演算法效率

時間
空間 效率

變因 { 實作
電腦
資料

大小位階

$$2.5n^2 - 2.5n \quad | \quad n_0?$$

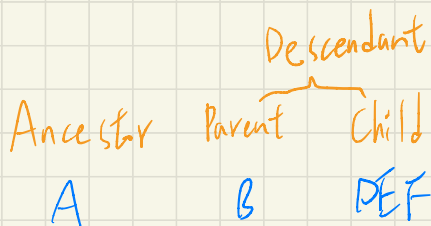
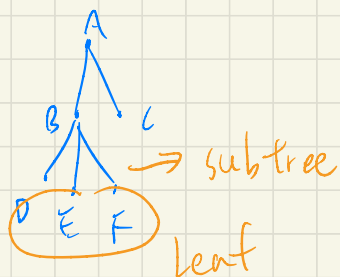
找到最差的時間

成長函數

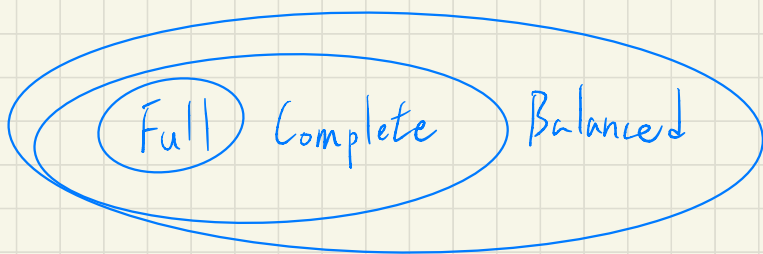
忽略低位階、常數

ch 8 樹

二維資料結構



階級越多越慢
(樹高)

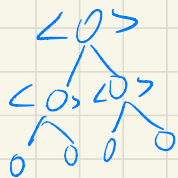


完整樹

1. h-1 全滿

2. 底層往左靠

原理



好找資料

Delete 父節點由右邊 child 繼承
(> 左 child)