

# HW5\_1 README

## heap.hpp

`int MaxHeapSwap(Heap * heap , int parent_pos) :`

傳入的參數是一個Heap物件(以下說明)，`parent_pos`指的是當前樹的節點的編號(`root` 是1，`parent = i` 則 `child = i * 2` or `i * 2 + 1`)。函數一開始會判斷`child`的大小，將最大的`child`選出再去和`parent`比，如果沒有`child`則回傳 `false`。如果`child`大則交換並回傳交換的`child`的編號，如果`child`小則回傳`false`。

總共有兩種回傳值：

1. 沒交換回傳`false`。
2. 有交換則回傳「被交換的`child`的編號」。

`int minHeapSwap(Heap * heap, int parent_pos) :`邏輯和`MaxHeapSwap`相同，只差大小於轉變而已。

## class Heap:

繼承自 `vector<int>`

## constructor:

**Heap() :** 一開始會初始化heap物件，將`RAND_MAX` push入Heap物件中當作陣列的第0號元素，並將`heap_method` 指向 `MaxHeapSwap`。

**Heap(vector<int>input) :** 會跟`Heap()` 做一樣的事情，差別在於會將`input`物件的內容全部push入Heap物件中。

## Data Member :

**(private) int (\*heap\_method)(Heap \*,int):** `heap_method` 是一個function pointer用來決定該是`MaxHeap`或是`minHeap`。

## Function Member :

**void InputElements(vector<int> input) :** 會將`input` 物件中的內容一一push入Heap 物件中。

**bool IsEmpty() :** 判斷Heap物件是否為空。

**int pop\_back() :** 回傳Heap物件中最後一個元素，並將它從物件中移除。

**Heap To\_Heap(int (\*method)(Heap \*,int)) :**

會從Heap物件的第一個元素開始一一scan並使用`method`將參數傳入做交換。回傳值有兩種型態(請參照`MaxHeapSwap`的說明)，三種情形：

1. 回傳值 `/ 2 == 1`代表是`root`與他的`child`交換。下次scan比須回到`root`開始。

2. 回傳值非零 代表有parent與child有交換，下次scan需回到parent的parent開始(因為child跟parent有交換，有可能child會比parent的parent大，所以下次scan須回到parent的parent開始)
3. 當回傳值 == 0時代表沒有做交換=>child和parent的關係符合規則，則下次scan從下一個node開始。

當所有的node都scan完後就代表所有的這個Tree已經是Max Heap or min Heap了。

**Heap Delete(int num) :** 將Heap物件中數字為num的第一個元素換成pop\_back()回傳的元素(呼叫reArrange函數)。並回到Heap的型態(MaxHeap or minHeap)

**Heap reArrange(int which, int num):**

which是Heap物件中哪一個元素，num是要將which替換的數。當which在Heap物件中不存在時會throw exception。函數會先將which 換成num後開始整理heap。

Scan Heap的位置從which的位置開始並使用heap\_method將參數傳入做交換。回傳值有兩種型態(請參照MaxHeapSwap的說明)，兩種情形：

1. 回傳值 / 2 == i代表它和它的child有做交換此時下次scan從交換後的child開始。
2. 如果回傳值==0 代表parent和child沒有做交換，但是有可能parent > 它的parent，所以下次scan從parent的parent開始。

**void Show() :** 將Heap中的內容output。

## hw5\_1.cpp

**vector<int> InputInstruction() :** 將使用者輸入的數字做處理並回傳。

**main() :**

宣告Heap物件，並呼叫InputInstuction將使用者輸入數字傳給Heap物件。利用heap.To\_Heap告訴heap物件現在要用哪種方式(MaxHeap or minHeap)做處理。使用heap.reArrange將heap內的元素做交換。