



Heap

課堂補充 by qazwsxedcrfvtg14

Sprout



課程影片

- 看了嗎?
- Q&A

Sprout



Heap 小知識

世界上的Heap有很多種

- 影片中介紹的Binary heap
- Binomial heap
- Pairing heap
- Thin heap
- Fibonacci heap
-

Sprout



Heap 小知識

- 為什麼會有這麼多種Heap?
- 有的常數比較小
- 有的可以 $O(1)$ 插入
- 有的可以 $O(1)$ 刪除
- 有的可以 $O(\log N)$ 合併??
- 有的可以 $O(1)$ 合併??!!

- 因為礙於篇幅，有興趣請自行上網查詢資料:P

Sprout



Heap 合併

I have a heap.

I have another heap.

Uhh!

A bigger heap!

Sprout



Heap 合併

- Heap合併是什麼?
- 簡單來說就是把兩個Heap變成一個Heap

Sprout



模擬情況

- 現在有 N 個Heap，有三種操作
- 把一個元素Push進其中一個Heap
- 從一個Heap中Pop元素
- 合併兩個Heap
- 假設總操作數最多 Q 次，總元素數最多為 N

Sprout



Heap 合併

- 先不管複雜度，來想個最天真的作法吧~
- Push和Pop就照一般的做法做
- 合併呢？
- 開一個新的Heap，把原先兩個Heap的元素一個一個的push進去
- 複雜度 $O(Q * N \log N)$

Sprout



Heap 合併

- 那我們換一種方式
- 把其中一個Heap的元素一個一個的push進另外一個Heap
- 複雜度?
- $O(Q * N \log N)$
- QQ

Sprout



Heap 合併

- 那...如果我們把比較小的Heap慢慢加到另外一個Heap中呢?
- 複雜度?
- ~~$O(Q * N \log N) \dots ?$~~
- $O(N \log^2(N) + Q \log N)$??!!
- 啟發式合併!
- 考慮對於每一個元素，它每次被併入一個新的集合，它所處集合的大小至少是原來的兩倍。所以對每個元素至多進行 $\log N$ 次合併操作，總複雜度 $N * \log N * \log N$



魔法

- 幾個在C++中關於Heap的魔法
- 白魔法
- `std::priority_queue`
- 黑魔法
- `__gnu_pbds::priority_queue`
- 能善用魔法是好事，但是要注意別走火入魔了！

Sprout



Flood fill

課堂補充 by qazwsxedcrfvtg14

Sprout



課程影片

- Q&A

Sprout



淹水

- 模擬「淹水」的過程很麻煩，幾個方向就要寫幾個很大串的 `if`，怎麼辦？
- 提示：有沒有發現所有的 `if` 其實都長得很像？

Sprout



淹水

```
int dx[4]=[-1,0,0,1];  
int dy[4]=[0,-1,1,0];  
  
for(int i=0;i<4;i++)  
    if(check(x+dx[i],y+dy[i]))  
        queue.push(x+dx[i],y+dy[i]);
```

Sprout



想一想

- 可不可以用 DFS 來做例題一(染色問題)？
- 可不可以用 DFS 來做例題二(喵喵抓老鼠)？
 - 有沒有什麼壞處？
 - 複雜度是多少呢？
- 如果題目是，每個格子都有一個高度值，高度值低於一定程度就會淹水，請問到最後會有幾塊水窪？
 - 用 BFS 可以做嗎？
 - 用 DFS 可以做嗎？

Sprout



Graph

課堂補充 by qazwsxedcrfvtg14

Sprout



課程影片

- Q&A

Sprout



存圖

- 當我們需要動態加邊的時候，該怎么做？
- 當我們需要刪邊的時候，該怎么做？
- 還有很多有趣的存圖方式！
 - 其實應該算是原本那兩種的變種

Sprout



存圖

- 相鄰串列
- 相鄰串列 with 平衡二元搜尋樹
- 前向星(forward star representation)???
- 相鄰矩陣
- 相鄰矩陣 with 串列(路徑壓縮)???

Sprout



比較

- 相鄰矩陣
 - 空間複雜度： $O(V^2)$
 - 查詢兩個點之間是否有邊： $O(1)$
 - 遍歷一個點 v 周圍的邊： $O(V)$
 - 增加一條邊： $O(1)$
 - 刪除一條邊： $O(1)$
- 相鄰串列、前向星
 - 空間複雜度： $O(E)$ or $O(\text{degree}(E \text{ total}))$
 - 查詢兩個點之間是否有邊： $O(\text{degree}(V))$ or $O(\text{degree}(V \text{ total}))$
 - 遍歷一個點 v 周圍的邊： $O(\text{degree}(V))$ or $O(\text{degree}(V \text{ total}))$
 - 增加一條邊： $O(1)$
 - 刪除一條邊： $O(V)$ or $O(1)$

Sprout



比較

- 相鄰矩陣 with 串列(路徑壓縮)
 - 空間複雜度： $O(V^2)$
 - 查詢兩個點之間是否有邊： $O(1)$
 - 遍歷一個點 v 周圍的邊： $O(\text{degree}(V)\log(V))$
 - 刪除一條邊： $O(\log(V))$
 - 增加一條邊： $O(V)$
- 相鄰串列 with 平衡二元搜尋樹
 - 空間複雜度： $O(E)$
 - 查詢兩個點之間是否有邊： $O(\log(V))$
 - 遍歷一個點 v 周圍的邊： $O(\text{degree}(V))$
 - 刪除一條邊： $O(\log(V))$
 - 增加一條邊： $O(\log(V))$

Sprout



想想看

- 給你一張有向圖，要怎麼知道這張圖上面有沒有環？
- DFS?
- BFS?

Sprout



想想看

- 現在有一個工廠，裡面有很多台機器，有些機器所生產的東西可能會依賴於其他機器，但是保證不會循環。
- 問：找出一個機器的執行順序使得過程中不會有機器依賴於還沒執行過的機器。
- 拓撲排序！
- 有向無環圖 (DAG)

Sprout



Game

- 有 N 個城市(編號為 $0, \dots, n-1$)，而有些城市間存在飛機航線。每個航線連接兩個城市，而且是雙向的。
- 大毛想知道是不是所有城市都能直接或間接地搭飛機往來。你不想直接說出答案，所以提議玩個猜謎遊戲。
- 大毛可以問你「城市A和B是否有『直接』的航線?」，而你需要立刻回答是或不是。大毛會試圖詢問每一組城市是否存在直接航線，總共問 $r=n(n-1)/2$ 個問題。如果大毛問到第 $i(i < r)$ 個問題就能推論出網路是否為連通(即「是否所有城市都能直接或間接搭機往來」)，大毛就贏了。反過來，如果大毛需要問完所有 r 個問題才能知道答案，那麼你就贏了。
- 為了讓遊戲更有趣點，你不需在意真實的飛航網路，且你可以在過程中隨意建構網路，依據大毛之前的提問來決定後面的回答。
- 你的目標是決定該怎麼回答，才能贏得遊戲。



Sprout