

宣告一個 char a

宣告一個 string s

宣告一個存放 string 的 deque d

宣告一個存放 deque(內部是 string)的 stack stk

宣告一個存放 deque(內部是 string)的 queue q

While(從檔案得到一個 char 給 a)

{

    如果 a 是#的話，跳出迴圈。

    如果 a 是換行，把 d push 到 stk 裡面，並清空 d。

    如果 a 是空白，繼續下一輪迴圈。

    如果 a 是其他(也就是數字或花色)，把 a 接在 s 後面，再看檔案的下一個字元是不是空白，若是，代表現在的字元是花色，把 s 放到 d 前面，清空 s。

}

宣告 2 個 int num1 num2

宣告 3 個存放 string 的 deque d1,d2,d\_temp

While(從檔案得到一個 char 給 a)

{

    如果 a 是換行，繼續下一輪迴圈。

    如果 a 是空白，再從檔案得到一個 char，把它轉成 int 並給 num2，接著比較 num1 跟 num2 的大小，若 num1 小於 num2，從 stk 中 pop 出小於 num1 數量的 deque 放到 queue 之中，下一個 pop 出的 deque 則給 d1，接著 pop 出 num2-num1 的 deque 到 queue 中，下一個 deque 給 d2，num2 小於 num1 則把 num1 num2 對調，d1,d2 對調。

    接著比較 d1 第一個 string 跟 d2 的每個 string(從前面開始，若有其中一個 char 一樣把比 d2 的第一個放到 d1 前面，否則放到後面，直到 d2 為空，這邊用一個 flag 紀錄卡是放到前面還後面，，最後，若 flag 是 0，先把 d1 push 到 stk 中，再把 q 中所有 deque push 進去，否則先 push q 中所有 deque 再 push d1。

    其他其況代表 a 是 num1，把 a 轉成 int 給 num1。

}

最後拿出 stk 中剩下的那個 deque 把裡面每個 string 印到 fout 當中。

時間複雜度分析:

我的整個程式有 2 重回圈以上的地方在於把 2 個 deque 從 stack 拿出、結合然後放回，所以時間大都花在這裡。

第一階段的拿出 2 個 deque，若一開始總 deque 數為  $n$ ，第一次結合最多需拿出  $n$  個，第二次最多拿出  $n-1$  個，以此類推，最後一次拿 2 個，所以拿出的總數最多為  $n(n+1)/2$  再減 1，第二階段的結合，若平均每個 deque 有  $m$  張卡，則總共需比較  $(n-1)m$  次，第三階段放回則是每輪比拿出來時少 1，所以是  $n(n-1)/2$ 。

整體時間約為  $O(n^2 + nm)$