

## DS project2

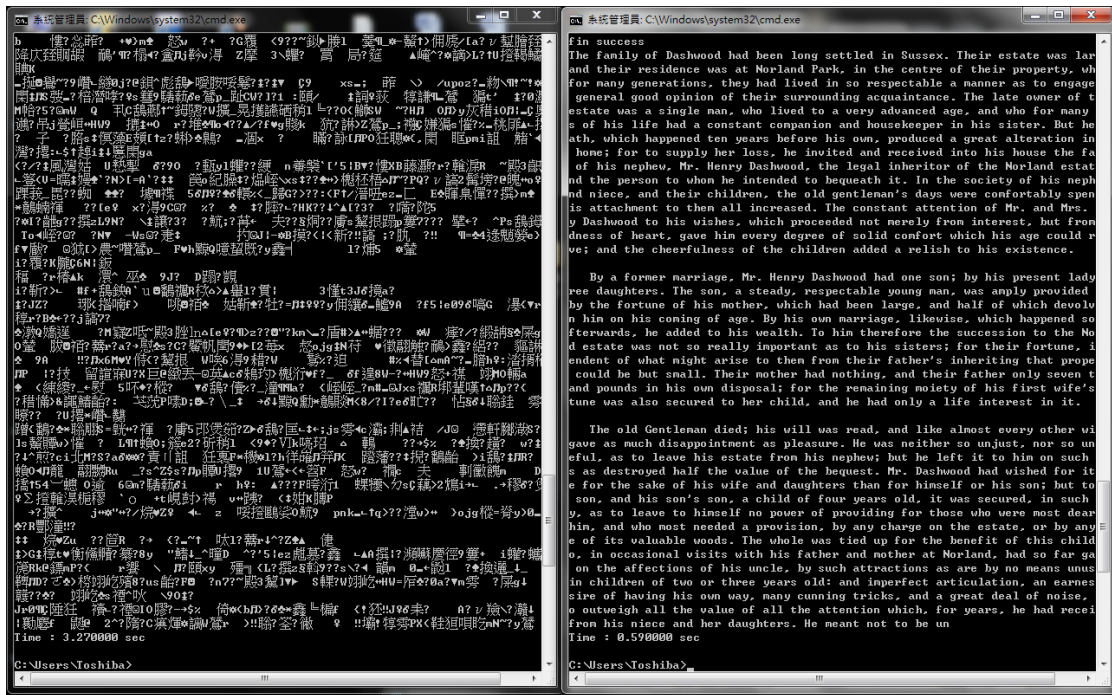
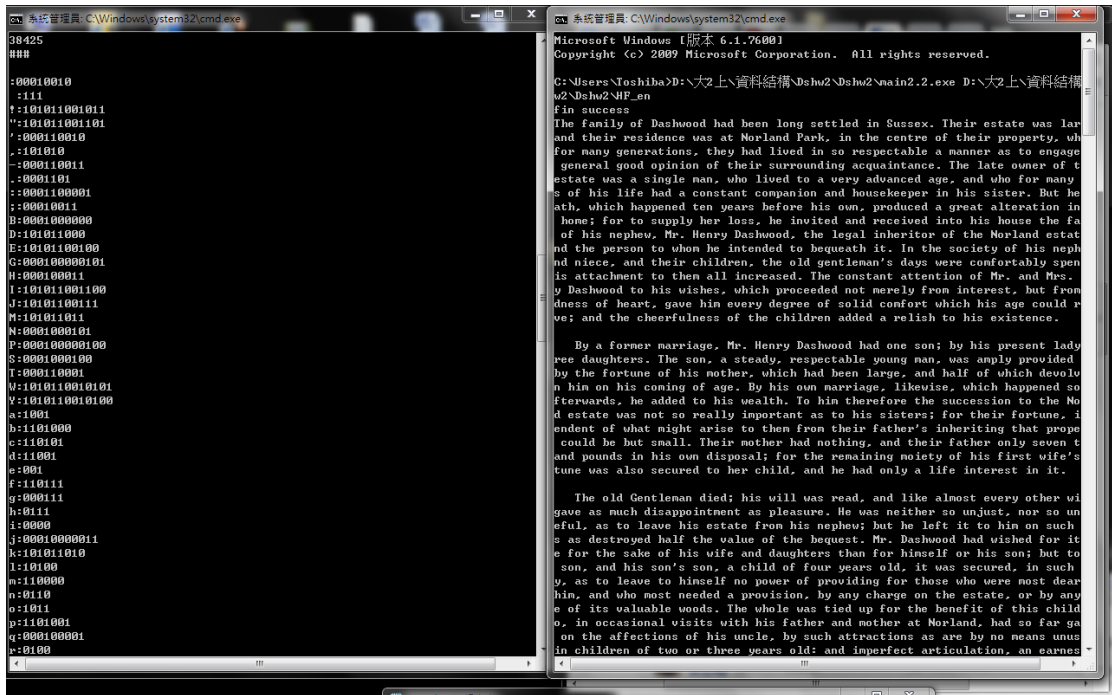
Encode 做法:

1. 定義一個型態為 data 的 struct 內容有: int count, char asc, bool flag, data left, data right 宣告 data list[128]
2. list[i].asc=i 讀進來的字對照 asc 碼將 list[i].count++, 讀完之後再用 qsort 把 list 由小到大排列, 如果 list[i].count==0 則將他的 flag 標為 false 不使用
3. 另建一個 data node[] 把 count 最小的兩個 list(i,i+1)複製給 node[k] node[k+1]將較小的 list[i].flag 設成 false, 較大的 list[i+1].left 指向 node[k], list[i+1].right 指向 node[k] 重新用 qsort 排列一次, 重複上述步驟, 直到剩下一個 list, 把最後一個 list 複製給 node[k], 樹就建好了, root 是 node[k]
4. 再來是樹的編碼, 建一個函式叫 traverse, 從 root 出發, 走左邊就給一個 0 走右邊就給一個 1 直到樹的盡頭(左右都為 NULL)再把編碼複製到 table
5. 把讀進來的字對應 table 換成編碼並存在一個 string 裡面, 把 string 裡的 01011.. 用 &,| 等做 bitwise 運並存到 char, 滿 8 個 bits 就 cout, 印出一串亂碼後, encode 就完成了。

Decode 做法:

1. 用一個 buffer 把亂碼變回來, 讀取一個字元 c: string t; if((c&1)==0){c=>>1;t+='0'} c 右移一個字 bit 並將 '0' 放入 t 裡, ex: 00001110-> X0000111 t=0; XX000111 t=01 最後 XXXXXXXX t=01110000 因為 t 會和本來的 bits 排列相反, 所以寫一個 reverse 函式將他 reverse 回來->t=00001110, 將 t 存入一個 string cc, cc+=t; cc 存的東西就是所有的 bits
2. 把樹建回去, table 是一個 struct, 存字元和編碼, 先假設讀 table[i], 指標 \*root=new node, 再宣告一個指標 \*p, if(讀到'0'){if(p->left!=NULL)p=p->left; else if (p->left==NULL){p->left=new node; p->left=p->right=NULL;}}讀到'1'的時候情況也依樣, 直到讀到'\0' p->asc='X' 在繼續讀下一個 table[i+1], 直到 table 讀完了, 樹就建好了!
3. 再做 tree 的走訪 如果是 0 走左邊 1 走右邊, 直到盡頭把字元印出來就完成了

我的 decode 在自己的電腦裡用 cmd 開會成功 可是工作站跑出來的結果不大一樣 找不出原因...



3.27s

0.59s