----- DS_Final_Project Report -----

107062313 黃寶萱

Q1) How you implement your code

1. For K-core (find coreness)

   <方法一> 利用兩個 list 的二維陣列，一個的 index 為 vertex ID，每個 ID 的 list 裡面存和那個 vertex 相鄰的點，另一個陣列的 index 為 degree，即將每個點放在其對應的 degree list 裡，接著從 degree=1 開始，依序將的 list 裡的所有點刪除，同時也要刪掉其鄰居，接著再更新每個點其所在的 degree list，過程中要同時記錄每個點最後屬於哪個 degree，其值就是那個點的 coreness，便將它存在一維陣列中。

   <方法二>首先，先用 set 的二維陣列記錄每個點的 neighbor，用一維陣列紀錄每個點的 degree，從 degree 最小(min)的點開始移除，將 degree<=min 的點放進 set 裡，接著跑過 set 裡的所有點，同時將其和 neighbor 的 edge 移除，重複直到找到所有的 coreness。

2. For clique

   接續上方找 coreness 的結果，因為已將每個點的 coreness 存進一維陣列，再利用網路上找到的演算法求 maximum clique，分成兩個部

分，With vertex ordering(左圖)以及 With pivoting(右圖)，

```
BronKerbosch3(G):
    P = V(G)
    R = X = empty
    for each vertex v in a degeneracy ordering of G:
        BronKerbosch2({v}, P ∩ N(v), X ∩ N(v))
        P := P \ {v}
        X := X ∪ {v}
```

```
BronKerbosch2(R,P,X):
    if P and X are both empty:
        report R as a maximal clique
    choose a pivot vertex u in P ∪ X
    for each vertex v in P \ N(u):
        BronKerbosch2(R ∪ {v}, P ∩ N(v), X ∩ N(v))
        P := P \ {v}
        X := X ∪ {v}
```

R: 目前的 clique。

P: 記錄可以增大目前 clique 的點 set，即接下來要列舉的點

（與目前 clique 上所有點皆相鄰的點所構成的集合）

X: 可以增大目前 clique 的點集合，但是為已經列舉過的點。用來避免重複取到相同的點。

其餘的部分只需利用 set 交集、聯集、差集的用法即可以完成。

Q2) Challenge I encounter in this project

一開始是用較直接的方法來找 coreness，但因為最初版本的寫法需要跑多層 for(int i=0; i<MAX; i++)的迴圈才能找到正確的 coreness，跑完 open_testcase_ds.txt 耗時大約介於 100~120 秒之間，找完 coreness 後再用 Bron–Kerbosch algorithm 來找出最大的 clique，此情況能符合 final project(basic)的時間限制(180 seconds)，但為了完成 bonus 將時間壓在 30 秒內，我修正了找 coreness 的算法，避免跑多層迴圈，新的算法加快了許多，同樣再利用 Bron–Kerbosch algorithm，便能在 30 內找完 coreness 和最大的 clique。除了時間限制上的問題，另一個遇到困難是關於 Bron–

Kerbosch 演算法，執行過程會有遞迴，但因整個遞迴完成的時間會超過 30 秒，而超過 30 秒會終止程式，因此無法在 main function 中將目前找到的 maximum clique 寫入 output 檔(clique.txt)，我一開始想到的解決方法是除了在 main function 中可以寫入 output 檔外，在 void signalHandler(int signum)這個 function 中也可以寫檔，但我不知道為甚麼在 signalHandler 寫檔當程式終止時會出現 Segmentation fault，問過同學後發現他們都將寫檔拉出到另一個 function 完成，我也試著這樣做便沒有 Segmentation fault，請問助教這可能是因為什麼?

Q3) Reference

- https://en.wikipedia.org/wiki/Bron%E2%80%93Kerbosch_algorithm#With_vertex_ordering (Bron–Kerbosch algorithm)
- https://iq.opengenus.org/bron-kerbosch-algorithm/ (Bron–Kerbosch algorithm)
- https://www.itread01.com/content/1547405646.html (c++ set 交集 並集 差集)
- https://www.itread01.com/content/1546682975.html (C++中 set 用法詳解)