

# How you implement your code

## K-core

```
void k_core(vector<vector<int> > &graph ){
    for(int i=0;i<VERT_NUM;i++){
        D[dv[i]].push_back(i);
        L[i]=0;
        kco[i]=0;
    }

    for(int i = 0;i<VERT_NUM;i++){//degree
        if(D[i].size()==0)continue;
        for(int j=0;j<D[i].size();j++){//vertex this degree
            if(dv[D[i][j]]!=i)continue;
            if(L[D[i][j]]==1)continue;
            for(int z = 0;z < graph[D[i][j]].size();z++){
                if(L[graph[D[i][j]][z]] == 1 || dv[graph[D[i][j]][z]]==i)continue;
                dv[graph[D[i][j]][z]]-=1;
                D[dv[graph[D[i][j]][z]]].push_back(graph[D[i][j]][z]);
            }
            L[D[i][j]]=1;
            kco[D[i][j]]=i;
        }
    }
    return ;
}
```

我尋找k-core的做法和ppt一樣，然後使用了和維基百科上一樣的過程。首先在輸入的時候就記錄每一個點的degree，存放在dv[]這個陣列中。接著建立一個存放陣列的陣列D，在D[i]裡面存放degree為i的vertex的陣列。同時建立一個L List來放置被刪除的點，和一個kcore的list來存放，都先初始化為零。接著一個一個degree的跑迴圈。從degree 0開始，把degree i的點從圖上移除，加到另一個List L 裡面。這個點的kcore就是i。把 i 存到kcore list中這個vertex的位置。除此之外，和這個vertex相連的vertex，如果還沒被放到 L list的就將degree減一，放到D[i-1]裡面。也就是說把跟現在所輪到的變這個點相連的邊都刪除掉，那相連的這些點的degree當然也跟著改變。直到迴圈跑完kcore就找完了。

維基百科的做法：

- Initialize an output list  $L$ .
- Compute a number  $d_v$  for each vertex  $v$  in  $G$ , the number of neighbors of  $v$  that are not already in  $L$ . Initially, these numbers are just the degrees of the vertices.
- Initialize an array  $D$  such that  $D[i]$  contains a list of the vertices  $v$  that are not already in  $L$  for which  $d_v = i$ .
- Initialize  $k$  to 0.
- Repeat  $n$  times:
  - Scan the array cells  $D[0], D[1], \dots$  until finding an  $i$  for which  $D[i]$  is nonempty.
  - Set  $k$  to  $\max(k, i)$
  - Select a vertex  $v$  from  $D[i]$ . Add  $v$  to the beginning of  $L$  and remove it from  $D[i]$ .
  - For each neighbor  $w$  of  $v$  not already in  $L$ , subtract one from  $d_w$  and move  $w$  to the cell of  $D$  corresponding to the new value of  $d_w$ .

## Clique

Clique是建立在k-core找完的基礎上。在k-core的時候，就把所有k-core大於1499的vertex存到一個list，並且用他們建立一個新的graph。接著利用ppt裡面所提供的coloring color這個graph。在塗上顏色後，利用dfs尋找clique，不停地加入新的vertex，如果達不到1500，就換一個vertex再繼續下一層。

## Challenge you encounter in this project

### 1. Make file

在開始project的時候首先遇到的困難就是makefile。一開始我是試圖自己寫makefile，但是不知道為什麼一直沒有成功。後來我就採用同學提供的做法，拿之前project的makefile，就沒有問題，馬上就成功了。我之後可能還要再多研究一下怎麼寫makefile。

### 2. Segmentation error

在完成makefile後，我試著直接用sample code編譯，但是馬上就出現了error。後來發現的第一個錯誤就是括號。在使用vector的時候，只要出現>>就會出現error，但在中間加了一個空格後才可以開始成功編譯。但這之後一開始跑又出現了segmentation error。這次經過一陣子的debug後發現完全不是code的問題，而是我執行編譯的指令錯了。又改了之後，開始跑我自己寫的k-core，結果又出現了segmentation error。原本我試著用簡單的printf來檢查到底是哪一行出錯，但後來發現是迴圈跑到一半就會開始segmentation error，而且每次出現error的時間點並不一致。後來我改掉了我iterator的寫法，就順利地跑完了。

### 3. 演算法

其實到最後我都沒有找到特別有效率的clique的演算法，翻了很多網站和論文，但因為我還是太弱了，大部分都看不懂，所以最後只用了很基本的方法。

## Reference that give you idea (github/paper...)

[https://en.wikipedia.org/wiki/Degeneracy\\_\(graph\\_theory\)](https://en.wikipedia.org/wiki/Degeneracy_(graph_theory))

<https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/>