

Data structure:

建立兩個 class: Vertex 和 Edge。

Vertex 紀錄該 vertex 之號碼、顏色、key 值等等。

Edge 紀錄該 edge 兩端之 Vertex 和 weight。

使用 `vector<Vertex>` 來存 vertices，`vector<Edge>` 存 edges，另外用二維 `vector<vector<int>>` 做為 edges 的 adjacent matrix，最後用 `vector<Edge>` 存要刪除的 edges。

Implementation:

Undirected: 使用 Prim's algorithms，取自課本 pseudocode 並做一些修改，因為我們要砍掉最小的 edges 並使之為 acyclic，其意義相等於長一顆有最大 weight 的 minimum spanning tree。有些部分需要做修改。首先，初始化成負無限大，在此作業中只需要設成 -101 即可；Extract\_min 改成 Extract\_max，找 vector 中有最大 key 的 vertex；最後兩 vertices 比較 key 值需要取大者。

Directed: 跑一次 DFS 找出所有的 back edge，因為 cycle 必含有 back edge，所以這些 edges 要刪除，另外，因為要砍掉的 edge weight 要越少越好，所以考慮把所有負 weight 的 edges 做刪除，但同時注意砍掉此 edge 要確保圖還是 connected。如果 back edge 為空集合，那麼代表圖原為 acyclic，因此不用再找負 edge 而直接輸出 0 即可。

Review:

Undirected 的部分不難，只需簡單修改課本上 pseudocode 的一些地方而已。

Directed 的部分我遇到比較多問題，一開始我的想法是找到 back edge 後再沿著此 back edge 找 cross edge，並記錄下這些 cycle，接著把所有 cycles 做比較，找出共同交集的部分並刪除最小 edge。但找出所有 cycle 會花費太多時間，不太能在 polynomial time 下做到。後來我採用另個方式，同樣找出 back edge，然後盡可能把其他負 edge 做刪除，算是有點投機的作法，因為此題的 weight 會小於 0。