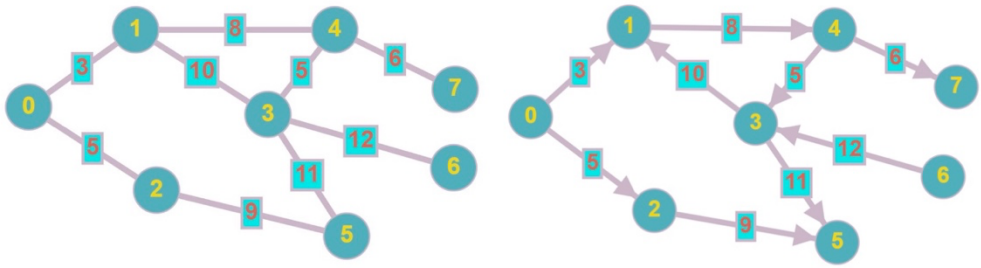
B08501011_PA3_report

Cycle breaking problem:



(a) weighted undirected graph (b) weighted directed graph

Figure 1: Graphs with cycles.

執行流程：

1. Undirected graph:
   - 執行 undiretedGraph function，輸入 edge 資訊，並判斷是 weighed or unweighted。
   - 若是 weighted 則先將 edge 依照 weight 進行排序（由大到小）
   - 執行 undirectedCycleCheck function，依序確認 edge 所連端點是否已被連結。
   - 確認完所有 edge 後，輸出總花費與刪除 edge。

2. Directed graph:
   - 執行 diretedGraph function，輸入 edge 資訊。
   - 執行 directedCycleCheck function，並先將 edge 視為 undirected，依序確認 edge 所連端點是否已被連結，若無則將將該 edge 分至 selected edge，反之，則將 edge 分治 deleting edge。
   - 依序將 deleting edge 中的 edge 重新放入 seleted edge 並利用 connectionCheck function 檢查是否形成 cycle，若不會形成 cycle 則將其放回 seleted edge，若會形成 cycle，則將之放至 deleted edge。
   - 確認完所有 edge 後，輸出 deleted edge 中被刪除 edge 與總花費。

重要函數介紹：

undirectedCycleCheck：

利用一個與 vertex 數一樣大的一維陣列（discoverVertices）來紀錄各 vertex 連結情形，且將所有元素初始化為零，並用一個 counter 來劃分不同 disjoint set。

Edge(a, b, w)：a, b 為端點，w 為權重。

1. 當 discoverVertices[a], discoverVertices[b]皆為零

   → 新的 disjoint set，將兩者 discoverVertices[a], discoverVertices[b]設為 counter 當前數字，counter++。
2. 當 discoverVertices[a], discoverVertices[b]一者為零

   → 新的 vertex 連結至 disjoint set，將零者設為不為零者的數字
3. 當 discoverVertices[a], discoverVertices[b]兩者皆不為零，且兩者不相等

   →兩 disjoint set 連結，將數字較大 disjoint set 數字設為數字小 disjoint set 的數字
4. 當 discoverVertices[a], discoverVertices[b]兩者皆不為零，且兩者不相等

   →同個 disjoint set 多餘連結，放入應刪除 edge

connectionCheck：

假設加入 Edge(a, b, w)會形成 cycle，則 b 必可連結至 a，所以只需確認 b 能否連至 a，檢查所有 edge，若有 Edge(a, x, w)，則 recusive call 看 x 是否可連至 b。結合一維陣列紀錄走過的 vertex，若有 cycle 則跳回上層繼續檢查其他 edge。檢查完所有 edge 後則可確認 b 能否連至 a。

directedCycleCheck：

先用 undirectedCycleCheck 的想法分類，再將待刪除的 edge 利用 connectionCheck 檢查放回會不會形成 cycle，若不會則放回，若會則刪除。