

1. Undirected unweighted/weighted graph:

將所有 $\text{edge}(u, v, \text{weight})$ 存入 vector，依 weight 由大至小排序，再利用 Kruskal Algorithm 選取 maximum spanning tree（但改為優先選 cost 大的），故 Cycle breaking 刪掉的 edges 會有最小的 cost。

Kruskal Algorithm 配合 Disjoint set 實現，從 weight 最大的 edge 開始選，當 edge 兩端的 vertices 利用 find set 找到的 root 相同，代表此 edge 會形成 cycle。

執行完 Kruskal 後，要刪除的 edge 存放於 vector 中，留下的為 maximum spanning tree，以 Adjacency list 方式儲存(方便 directed graph 做 DFS)。

Runtime(run Kruskal):

Make-Set: $O(V)$

Sorting: $O(E \lg E)$

Find-Set & Union: $O(E \alpha(E, V))$

Total: $O(E \lg E + V)$

2. Directed weighted graph:

先以與 undirected graph 相同方式執行 Kruskal 後，檢查被刪除掉的 edges，以 Heuristic 判斷，若 edge weight 為負數，確定刪除（可以使 Cycle breaking 的 cost 下降），若 edge weight 為正數，暫時將其加回 graph 中，再利用 DFS 找是否有 back edge，若有 back edge 代表會形成 cycle，必須刪除，反之則加回 graph 中。

Runtime:

Kruskal: $O(E \lg E + V)$

DFS for at most d times (執行完 Kruskal 刪除 d 個 edge) : $O(d(V+E))$

Total: $O(E \lg E + d(V+E))$

