Question 3

You are given a simple directed weighted graph with n vertices and m edges. The edge weights *may* be negative, but there are no cycles whose sum of edge weights is negative.

3.1 **[10 marks]** An edge e is said to be *useful* if there is some pair of vertices u and v such that e belongs to **at least one** shortest path from u to v.

Design an algorithm which runs in $O(n^3)$ and determines the set of useful edges.

Answer:

Suppose u, v, k is the index of vertices, u is the start index of vertices, v is the end index of vertices, k is the possible passing index of vertices, n is the total number of vertices. Suppose w, w' is $n \times n$ two-dimensional array which record the weight of edge to edge. $w[u][v]$ means the weight of weight from u to v, w' record the original weight between two vertices, w record the changing after finding the best way.

Traversing both w and w' arrays, if exist an edge from u to v, record the weight into $w[u][v] and w'[u][v]$, if not or $u = v$ record $w[u][v], w'[u][v] = \infty$.This part need to travel all of the $n \times n$ arrays, the time complexity is $O(n^2)$

Set $u, v, k = 1$.
Set a three-layer nested loop, use Floyd-Warshall algorithm method:
- the outermost layer loop k from 1 to n $(1 \leq k \leq n)$:
- the second layer loop u from 1 to n $(1 \leq v \leq n)$, if $u = k$, jump over:
- the input layer loop v from 1 to n $(1 \leq v \leq n)$, if $u = v \,\text{OR}\, v = k$, jump over:

- If $w[u][v]$ greater than $w[u][k] + w[k][v]$, set $w[u][v] = w[u][k] + w[k][v]$.

This part have three-layer nested loop and both of them need to travel size of n, the time complexity is $O(n^3)$ After that, compare array w and w'

- if $w[u][v] \neq w'[u][v]$, it means the original edge is not being chosen by the shortest path, there are not useful path.
- if $w[u][v] = w'[u][v] \,\text{AND}\, w[u][v] \neq \infty$, the path is useful.

This part need to travel all of the $n \times n$ arrays, the time complexity is $O(n^2)$.

The time complexity is $O(n^3)$.

3.2 **[10 marks]** An edge is said to be *very useful* if there is some pair of vertices u and v such that e belongs to **every** shortest path from u to v.

Design an algorithm which runs in $O(n^3)$ and determines the set of very useful edges.

Answer: