Question 3 [25 points] You are given an undirected graph G = (V, E) with positive weighted edges, and an edge e ∈ E. Design an O(|V|^2) algorithm to compute the length of the shortest cycle containing e.

For this question, We can take advantages of Dijkstra's shortest path algorithm which returns the shortest distance from some vertex to all other vertices if all vertices are marked visited

The reason is that we want to find a vertex that has the smallest distance to u and v. The vertex has the property that d(u,vertex) + d(v,vertex) is minimum.

Let this edge be e =(u,v). For each vertex from u and v, we want to apply Dijkstra's shortest path algorithm separately on u and v.

Below is a general steps for my algorithm:

1. Run Dijkstra's shortest path algorithm on input(G,u). This step will output an array that records every vertices shortest distance to u. Let's say this array is A.
2. Run Dijkstra's shortest path algorithm on input(G,v). This step will output an array that records every vertices shortest distance to v. Let's say this array is B.
3. Now, for each vertices i in A and B, we output the vertex that A[i] + B[i] is minimum (we choose a vertex that has the smallest distance to u and v.
4. If there is no such vertex exists, the graph does not contain a cycle containing e.

Now, suppose the returned distance is P. We just need to do: P + edge weight of (u,v).
The running time of this algorithm is O(|V|^2). The reason is that step one and two takes O(|V|^2) (dijkstra's algorithm) and step takes O(|V|) since it just scans every vertex in the array.

Dijkstra's algorithm(this algorithm is taught in lecture notes, I will just briefly describe it here) Dijkstra's algorithm basically assigns the distance of every vertex in the graph to the current vertex to infinity and starts to visit vertex. Visited vertex are marked visited.The algorithm keep updating the distance if there is a shorter distance to get to the current vertex.(this is done by considering all neighbours of each vertex). As a result, the running time for this algorithm is O(|V|^2)If all vertices are visited, the algorithm is done.

Note that the above algorithm can be reduced to just using one dijkstra's algorithm. We don't need to find the intermediate vertex. Instead, we simply remove the edge e in the graph and apply the dijkstra's algorithm on (G,u). this will return the shortest distance of (u,v) without edge e. If the distance between u and v is infinity, the graph does not contain a cycle containing e. The final output should be edge weight of e + the shortest distance of (u,v) in the graph without edge e.