Intro to Algorithms — Homework 6.

Q1.

By approaching to this question, the algorithm to find the number of distinct shortest paths is using BFS. The running time is $O(|V|+|E|)$, where $|V|$ is number of nodes, and $|E|$ is number of edges.

The implementation of this algorithm will be traversing the graph using BFS. For every vertex X have neighbors Y, we keep track all of visited nodes and its weight. Hence, for each iteration, the visited nodes and weight will be updated, and we will be able to compare the shortest paths.

Therefore, this algorithm have a running time in $O(|V|+|E|)$. Which is mean we traverse the V elements and also traverse all the possible edges, thus giving $O(|V|+|E|)$.

Q2. In order to find the length of the shortest cycle or report that the graph is acyclic in $O(|V|^3)$ running time algorithms.

First, we can modify the Dijkstra algorithm to do this job. We can start at path[i][j] = "inf". In the end of the Dijkstra algorithm, afthe $O(|V|^2)$ running time, we will have the shortest path on a given node which is path[i][j].

Next, by approaching to find the shortest cycle or report the acyclic, we add a step inside the Dijkstra Algorithms to traversal any given nodes to find the possible cycle. In this step, we also need to traversal all nodes, thus by the combination of Dijkstra algorithm and this one more step, we will have $O(|V|^3)$ time complexity.

The array of $|V|$ elements

```
for i in V:
    for j in V:
        path[i][j] = "inf"
        ⋮
        Dijkstra's Algorithm.
        ⋮
for all v in V:
    ⋮ Find the shortest cycle.
```

This give $O(|V|^2)$

$O(|V|^3)$ time complexity.

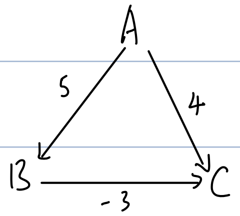
Q3. $l(x,y)$ weight of the edge. $w(x) \rightarrow$ weight of x (vertices)

By approaching to this question, we can applied Dijkstra Algorithm to help for the solution. In addition, during the process of Dijkstra Algorithm calculating the Distance between two vertex, using the variables to include the weight of the vertex ($w(x)$) and the weight of the edges. ($l(x,y)$).

Dijkstra Algorithm already help us find the smallest cost path from a source vertex to all other vertices. Since we also applied the queues technique in the implementation of Dijkstra Algorithm (Lab 6). We have running time $O((|V|+|E|)\log|V|)$. The 'makequeue' technique take the array of $|V|$ for the insertion operation. Also, 'deletemin' → popitem( ), we will have total of $|V|$ element and $|V|+|E|$ insert / decreasekey operation. From the textbook, in the binary heap implementation (descript above, Lab 6), we have running time of $O((|V|+|E|)\log|V|)$

Q4. The method is wrong, the reason is with the negative edge weight, the calculation of Dijkstra's Algorithm will be incorrect, because it will modify the original_ path_ length with the negative edge weight

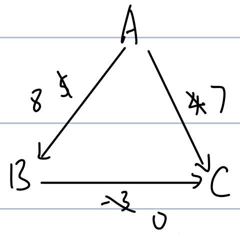
For example:



From this graph, to find the shortest from A → C, is A → B → C. which is 2 weight in total.

However, we won't have any negative weight during the calculation. hence, we need to add up the negative weight to 0, also added on all other edge weight.

So, the graph will become like this,



And the path become A → C, which is different compare to the previous one.

Thus, this method is wrong.