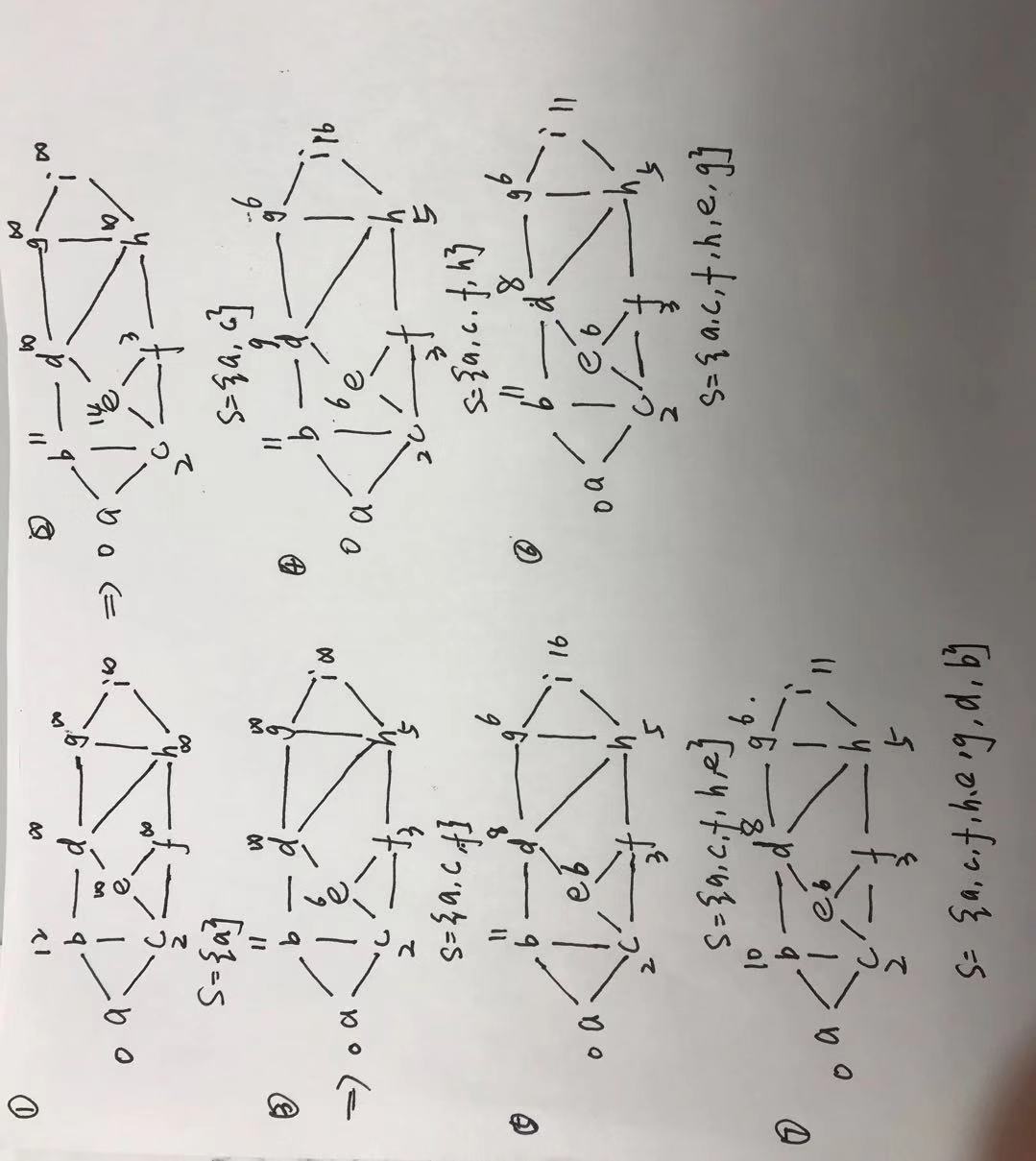
1.



2.

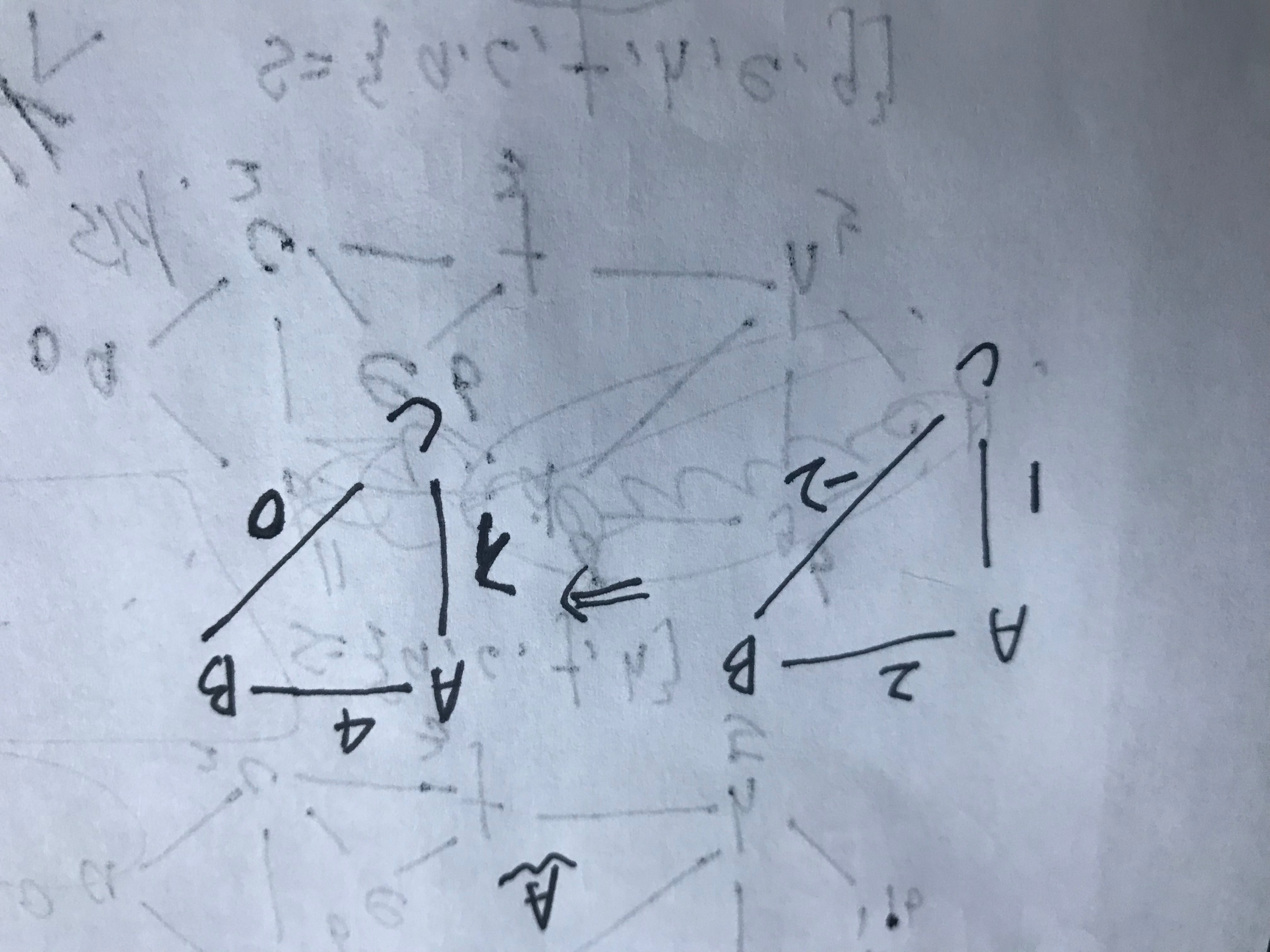
Slow algorithm results:

Fast algorithm results:

3.

4.Because the distance from v to any other vertex in G can be at most cn, I set a array of size cn that stores which nodes are at what distances. Then every time we need to extract the minimum distance node, we can loop over the array in O(cn)=O(n), and Relax function will cost O(1) every time, so the running time of Dijkstra’s algorithm will be improved to O(n+m).

5.The approach does not give correct results. There is a countexample.



In this graph, the shortest-path from A to C is (A-B-C), however, through Dijkstra’s algorithm the shortest-path from A to C is (A-C). Even if convert the negative edge to positive by adding a positive to all edges, the shortest-path from A to C does not change through Dijkstra’s algorithm, so this approach is not correct.

6. We can run the Floyd-Warshall algorithm one more iteration to detect the negative cycles, because if there is a negative cycles, there must exist some cheaper shortest\_path cost.

7.In this problem, both vertex and edges have weight. So I make some change to Dijkstra’s algorithm. In Relax function, in addition to calculate the weight of edge, we also need to add the weight of vertex. I will give Relax function below.

Relax(u,v,w1,w2)

If v.d>u.d+w1(u,v)+w2(v)

v.d= u.d+w1(u,v)+w2(v)

v.=u.