1) 4.3 $G = (V, E)$ undirected graph

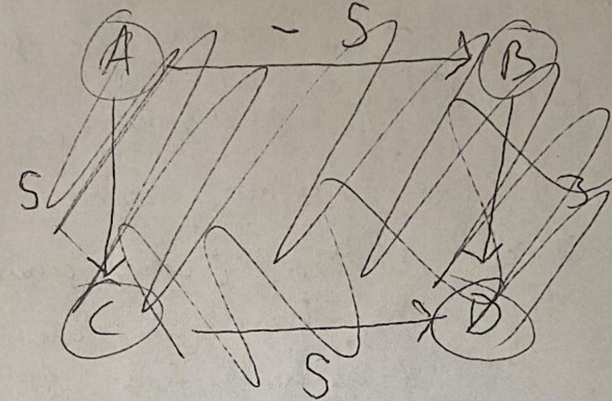
whether G contains simple cycle of length 4

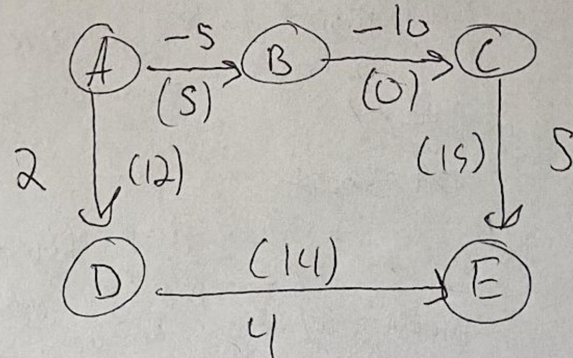
run time at most $O(|V|^3)$

- simple cycle exists if there is a pair of distance vertices such that they have at least 2 common neighbors
- $O(|V|^2)$ pair of distinct vertices
- (x, y) we distinct vertices ~~such that they~~
- check if x & y (~~which~~ where x ≠ y) to see if they have 2 common neighbors
    - For each vertex A that is adjacent to X, check to see if it is also a vertex that is adjacent to Y
    - ~~A Best~~
        - Do this through hashing (setting up a hash table ~~to~~ & populating it w/ every adjacent vertex)
    - The hashing is done in O(1) time

2) 4.8

Algorithm for finding shortest path from node s to node t. Directed graph w/ negative edges. Add a large constant to each edge weight so that all weights become positive, then run Dijkstra's algo at node s & return shortest path found to node t





Go from A to E, shortest path

~~A~~ Before constant - A, D, E length = 6

Constant value =+10 After constant - A, B, C, E length = 20

((A, D, E) is now 26)

3) 4.11 Directed graph w/ Positive edge lengths
returns length of shortest cycle in the graph
At most $O(|V|^3)$

- Let G be a graph
- Matrix M_{AB} w/ shortest Path between A & B
- For any vertice C, D, there is a shortest paths between C & D
- $M_{CD} + M_{DC}$ is Length of cycle
- Compute the minimum of $M_{CD} + M_{DC}$ for any Pair of vertices

run Time

- Any row R of ~~A~~ the matrix M ~~can~~ is $O(|V|^2)$ by using Dijkstra's algorithm
- ~~Time~~ computing $M_{CD} + M_{DC}$ is $O(|V|)$

$O(|V|^2) + O(|V|) = O(|V|^3)$