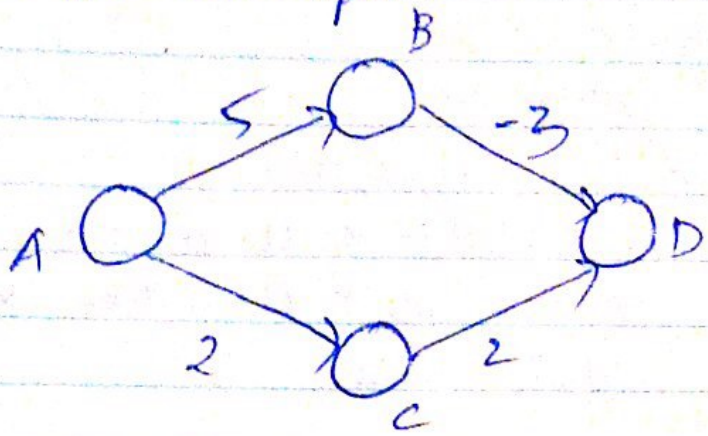
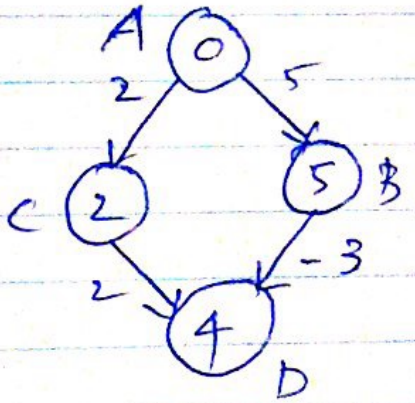
CS 430 Assignment - 9

① i) To prove: Dijkstra's algorithm will not work for graphs with negative edges.

Consider the example:



Applying Dijkstra's algorithm to above graph produces the vertices in the order: A, C, D, B
forming shortest path tree as follow



This is not the shortest path, proving that Dijkstra's algorithm fails for graphs with negative edges

(1) ii) Dijkstra's Algorithm to solve the given problem:

u.distance ← ∞
u.Pi ← NIL
Extract_max( )
for all vertices V
    Relax vertices (u, v, w)
        if v.distance < u.distance × (u,v)
            v.distance = u.distance × w(u,v)
            v.Π = u

(2) i) Negative Cycle → A cycle of vertices in a graph with total weight < 0.

Bellman-Ford algorithm takes V-1 steps to calculate distance for all nodes in a graph. (Without a negative cycle).
If there is a negative weight cycle, you can go on relaxing it nodes indefinitely.

Thus running Bellman ford algorithm 1 extra time after usual 'V-1' steps will detect the negative cycles.
For all vertices,

If there is a negative cycle,
$$G[n-1, v] \neq G[n, v]$$
else,
$$G[n-1, v] = G[n, v]$$

(2) ii) Negative cycle in Floyd-Warshall Algorithm.

If there is a negative cycle in Graph G, then there will be atleast one vertex v_i, that belongs to that cycle such that $D_{ii} < 0$.

Thus to also detect negative cycle in a graph, we have to check for negative entries in the diagonal entries of output D[] of the FLOYD-WARSHALL Algorithm.