1.Solution:

Let A be the $|V| \times |V|$ adjacency matrix of G.

Pseudocode:

```
for i ← 1 to |V|
    for j ← 1 to |V|
        G^2 [i][j] ← 0
        for t ← 1 to |V|
            G^2 [i][j] ← max( G^2 [i][j] , a[i][t]*a[t][j])
return G^2
```

Time Complexity:

In this pseudocode, there are three loops. For the third loop, the operation only costs O(1) time. Thus the time complexity is $O(|V|^3)$.

2. Solution:

Discovery/finishing times from performing the DFS traversal of the graph are as follows:

Vertex	Discover time	Finish time
m	1	20
q	2	5
t	3	4
r	6	19
u	7	8
y	9	18
v	10	17
w	11	14
z	12	13
x	15	16
n	21	26
o	22	25
s	23	24
p	27	28

The sequence is obtained by reading off the entries in decreasing order of finish time: p(28), n(26), o(25), s(24),m(20), r(19), y(18), v(17), x(16), w(14), z(13), u(8), q(5), t(4).

3. Solution:

Suppose that the form of the graph is the adjacency list form.

Pseudocode:

```
for each vertex u ∈ V do
    in-degree[u] ← 0        # take time O(|V|)
for each vertex u ∈ V do
    for each v ∈ Adj[u]
        in-degree[v] ← in-degree[v] + 1 # computes the in-degree for all vertices, which take time O(|V|+|E|)
Q←∅ # Q is a queue
```

```
for each vertex u ∈ V do
    if in-degree[u] = 0 then
        Enqueue (Q,u) # collects a list with all vertices of in-degree 0, which take time O(|V|)
# If the in-degrees of some vertex decreases to 0, then it's added to the list
while Q≠∅ do
    u ← Dequeue (Q)
    output u
    for each v ∈ Adj[u] do
        in-degree[v] ← in-degree[v] − 1
        if in-degree[v] = 0 then
            Enqueue (Q,v) # take time O(|V|+|E|)
# The sorted result is given by the order in which they are extracted from the list.
```

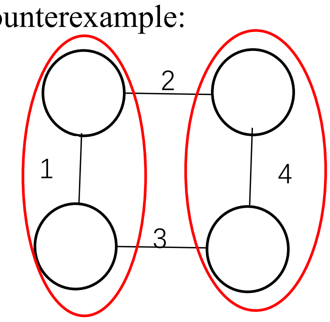
Time Complexity: O(|V|+|E|).

Discussion: If the input graph is cyclic, then the list becomes empty before finishing all the vertices, and it's impossible for this algorithm to complete the sort.

4. Solution:

The given algorithm will fail.

Counterexample:



According to the given algorithm, we can get a tree of total cost 1+4+2=7. But obviously, the weight of the minimum-spanning-tree should be 1+2+3=6. So the given algorithm fails to obtain a minimum spanning tree.