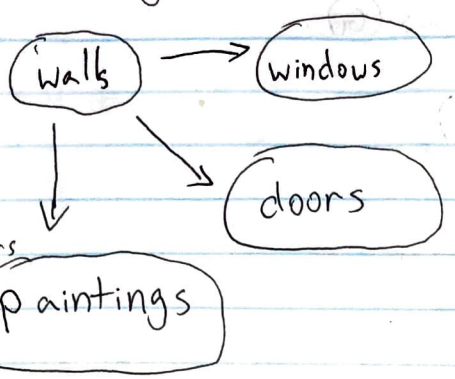
Gordon Ng CS 300 H.W #5

a)



nodes are a connection point inside a network that sends/recieves/create or store datas. We need to find out if we finished building the walls before we go to windows, doors or paintings. An edge is the connection of nodes, they can be directed. We do not have a double arrow from windows and doors due to an infinite loop problem. Windows, doors, and painting is on the same layer.

walls → win, pai, doors
win →
pai →
doors →

b) The property that you should not have cycles that keep you in an infinite loop, so the graph has an ending, aka the building can be complete.

c)

```
Update GraphAndDegrees(G):
    N = len(G)
    indegree = Zeros(N)
    for node in 1 to N:                    // n
        for linkedlist in node.next:       // m
            indegree[linkedlist] += 1;
    delete_nodes = []
    for i in (i to N):                     // n
        if indegree[i] == 0:
            delete_nodes.append[i]
    for delete in deletenodes:             // n
        for linkedlist in nodenext:        // m
            indegree -= 1;
            linkedlist.edges = null;
    return G, array.
```

We iterate through the linked list, list, and then each key in each linked list which is $O(n+m)$ together. Then removing and adding indegrees of 0 to deletenodes is $O(n)$, removing them and updating them is $O(m)$. The overall upperbound of this function is $O(n+m)$. This would be $3n+2m$ which is still $O(m+n)$. Another approach would to use sets, and add linked lists to the set, and delete the ones without another link for the linked list.

d) function Order_of_Tasks(G):

```
N = len(G);
Array B = BFS(G);
head node = B[0];
Array A. append (tuple(head node));
Order = []
while len(G) != 0:
        nodes_that_are_removed = UpdateGraphAndDegrees(G);
        Array A, append (tuple (nodes_that_are_removed))
        order += Array A
return Array A.
```

This is O(n+m) as the BFS function is used to find the source node in O(n+m) time, then it traverses O(n) of the graph to use UpdateGraphAndDegrees which is also O(n+m), which gives us an O(n+m) generated array of daily tuples.

Example: [(walls), (paintings, windows, doors)]