Write a Dynamic Programming algorithm whose complexity is O(N) for the following problem: You are given a fixed positive integer N. For each integer i in the range 1<=i<=N, calculate and print the sum of square of integers 1 up to i (i.e. 1+4+9+...+i^2). For example, if N=4, your algorithm should print the output below. Show that running time of your algorithm is O(N).If your algorithm is not Dynamic Programming, you will only get 1 point

```
1
5
14
30
```

Write an algorithm to find the node(s) with maximum number of incoming edges in a given directed graph. If there are multiple nodes which has this maximum number of incoming edges, your algorithm must output all of them. The input of the algorithm is the set of nodes, and the set of edges in the graph. Edges are represented as (K,L) which means there is a directed edge from K to L. Note that your algorithm must terminate for every possible input. Find the running time of your algorithm in terms of O(_)

Recall the Tower of Hanoi problem in the lecture slides: There are 3 rods. N disks are initially placed at a rod and all disks have different size. The objective is to move them from an initial rod to one of the other two rods (destination rod). Find the solution to this problem for N=4 disks. Develop a generic algorithm for N rods

Explain why A* algorithm is faster than Dijkstra's algorithm to compute the shortest path to a node

State the best-case and worst-case complexity of Linear Search and best-case and worst-case complexity of Binary Search (using O(..) notation). Briefly explain what are the best-case and worst-case situation (when they occur)

State the best-case, average-case worst-case complexity of Search over a Binary search tree (using O(..) notation).

State the best-case, average case and worst-case complexity of Insertion Sort using O(..) notation. Briefly explain what are the best-case and worst-case situation (when they occur)

Explain the difference between Binary tree, Binary search tree, AVL tree

Fill in the missing lines in the algorithm below for DFS post-order traversal on a tree.

```
 Algorithm DFS_postOrder
    Input: root node of the binary tree
     if(node == null)
         return;

     DFS_postOrder(____________________);
     DFS_postOrder(____________________);
     ____________________;
```

Which of the following statements is correct?

Higher efficiency means lower computation time
Running time of an algorithm is measured in seconds
Stacks have 1 end but queues have 2 ends
Binary search does not require input data to be sorted
Stacks cannot be implemented using ordinary arrays

Which of the following statements is incorrect?

Elements of a linked list do not have to be at consecutive memory location
A node in a tree can have at most 2 child node
Search over a Binary search tree is faster than linear search
Graphs do not have a root node
Collusion in a Hash table occurs when two different elements mapped to same location