1. Write a dynamic programming algorithm for computing $C(1,n)$ from the following formula. Before setting up the iteration loops carefully observe that all the needed values should be available. Analyze the space and time complexities of your algorithm. Draw a blank table for $C$ indicating the order of your computation (loops).

$C(i, j) = 0$, for all $i \geq j$

$C(i, j) = \max\{ C(i, k_1) + C(k_2, j) + 2$

\[
\text{for each } k_1 \text{ with } i < k_1 \leq n, \text{ and for each } k_2 \text{ with } 1 \leq k_2 < j\}, \text{ for all } 1 \leq i < j \leq n$
2a. What is the asymptotic *time-complexity* and *space-complexity* of the following code fragment in terms of \( n \)?

(1) count = 0;
(2) For \( i = 1 \) through \( n^2 \) do
    (3) For \( p = 1 \) through 3 do
        (4) For \( k = 1 \) through \( i \) do
            (5) count = count +1;
        end for loops;

2b. What is the asymptotic *time-complexity* and *space-complexity* of the following code fragment in terms of \( n \)?

(0) create a blank link list \( L \);
(1) For \( i = 1 \) through \( n \) do
    (2) For \( p = i \) through \( n*n \) do
        (3) create a new node and add to a link-list \( L \);
            // presume \( O(t) \) time-complexity for the step (3), for current list of size \( t \)
    end for loops;
3a. Run the following steps on the directed graph $G$ below.

Step 1. Using the depth first traversal (DFS) algorithm label the nodes of the graph $G$ in a post-order traversal (1 through 9).

Step 2. Draw and traverse the corresponding reverse graph $G'$ (where arcs of $G$ are reversed), again using the DFS algorithm, but now according to the high-to-low numberings of labels on the nodes from the previous traversal (i.e., use higher numbered node when multiple nodes are available to choose from).

Output: What do the output indicate about $G$? (Above is a specific graph algorithm for specific purpose.)

Directed graph $G$:
4. An integer 2D matrix (or image) may have at most one entry as a zero value. Write a recursive *divide-and-conquer* algorithm to find out the (pixel) coordinate of that entry, or return/print that all entries (pixels) have non-zero values. Analyze its space & time-complexity.
5. Answer briefly or mark on *true/false* for the following sentences. (You may explain your answer in a line if you want to.)

a. Name an algorithm for finding a shortest path on a weighted graph from a given starting node.

b. There is no polynomial-time algorithm for finding the shortest paths between all pairs of nodes in a weighted graph.  TRUE / FALSE

c. Name the *problem* of sequentially ordering nodes of a directed acyclic graph.

d. Mention a situation (property of the input list and algorithm’s pivot choice strategy) when *QuickSort* algorithm takes $O(n^2)$ time.

e. What is the number of *non-null* subsets of a set \{a, b, c, d, e, f, g, h\}?

f. How many *triangles* can be drawn out of $n$ nodes?

g. A problem $X$ is NP-class but does not belong to the NP-complete class. Then, $X$ must belong to P-class.  TRUE / FALSE

h. A problem $X$ has a polynomial-time transformation algorithm $T$ to a P-class problem $Y$. What can you comment on the output size of the algorithm $T$ with respect to the input size of $X$?

i. 5-SAT problem is an NP-class problem.  TRUE / FALSE

j. 5-SAT problem is an NP-hard problem.  TRUE / FALSE