1-Let $T$ be a tree with positive costs assigned to edges. Design a linear time dynamic programming algorithm to compute a maximum cost independent set in $T$. You need to define all terms, write a recurrence relation, and then describe the algorithm.

2-Consider the problem of computing the longest common subsequence of 2 given sequences. Design a dynamic programming algorithm for solving this problem. You need to define all terms, and write a recurrence relation, and then proceed with the algorithm.

3-Outline the algorithm requiring $O(n)$ many network flow problem, for computing the edge connectivity of an directed graph.

4-Let $G = (V_1, V_2, E)$ be a bipartite graph. Outline an algorithm for finding a maximum matching in $G$ by transforming the problem to a network flow problem. What is the time complexity?

5- Outline the FF algorithm for solving the Network flow problem and prove that it computes a maximum flow when capacities are integers. (You do not need to define augmenting paths.)

6-Assume that we have a way of multiplying 2 by 2 algorithms using 7 scalar multiplications. Outline a divide and conquer algorithm for multiplying two $n$ by $n$ matrices, write a recurrence relation, and derive its time complexity.

7-Solve $T(n) = \sqrt{n}T(\sqrt{n}) + n$.

8-Solve $T(n) = 6T(n/2) + n^2$.

9-Let $G$ be a graph. Outline an approximation algorithm for computing a 2 times optimal vertex cover in $G$, using the concept of a maximal matching, and verify its correctness. What is the time complexity.