1. You are given a rectangular piece of cloth of size A X B, where A and B are positive integers, and a list of clothing products that can be made using the cloth. For each product $p_i$ , $1 \le i \le n$, you know that a piece of cloth of size $a_i$ X $b_i$ is needed and that the selling price of the product is $c_i$ ($a_i$ , $b_i$ and $c_i$ are positive integers). You have a machine that cuts the cloth into two pieces, either horizontally or vertically. Design an algorithm which determines how to cut the cloth so that from resulting pieces you can make products of maximal total selling price. You can choose to make any number of each product (including none if desired).

2. A vertex cover of a graph G=(V,E) is a subset of vertices $S \subseteq V$ which includes at least one end point of every edge in E. Give a **linear time** algorithm which, given an **undirected tree,** outputs the smallest possible size for which there exists a vertex cover of that size.

3. Several families go out for dinner together. To increase their social interaction, they would like to sit at tables so that no two family members sit at the same table. Design an algorithm which produces a sitting arrangement that meets such objective or correctly outputs that there is no such arrangement. You are given the number of families, and for each family the number of its members. You are also given the number of available tables and the seating capacity of each table.

4. Design an algorithm which, given a flow network, outputs a minimal cut in the network and runs in polynomial time. You can use any algorithm presented in the lectures, together with the established estimate of its run time.

5. Assume you are given three sequences of letters, s1, s2 and s3.

    a. Show that if you find a longest common subsequence s' of s1 and s2 and then a longest common subsequence s" of s' and s3, then s" might NOT be a longest common subsequence for s1, s2 and s3.

    b. Design an algorithm which, given three sequences, finds a longest sequence common to all three sequences, i.e. a sequence of longest possible length which is a subsequence of all three sequences.

6. **EXTENDED CLASSES ONLY (3821 and 9801)** Assume that you are given a connected graph G with n vertices and a minimal spanning tree for that graph. You are also given a new vertex and its adjacent edges, connecting the new vertex with the existing vertices of G. Design an algorithm which constructs a minimal spanning tree for the enlarged graph and which runs in time linear in n, even if the starting graph has a quadratic number of edges. Prove your algorithm is correct, i.e. that you get a minimal spanning tree for the larger graph.