1. Consider the following special case of UNION-FIND. There are three phases where in each phase all the UNIONS precede the FIND. Can you design a more efficient implementation (than the one described in class).

2. We are given a sequence of integers in the range \([1, n]\) where each value occurs at most once. An operation called EXTRACT-MIN, occurs at arbitrary places in the sequence which detects the minimum element up to that point in the sequence and discards it.

For example in 4, 3, 1, E, 5, 8, E, ... the output is 1, 3.

Design an efficient algorithm to handle a sequence of such operations.

3. Design an efficient (preferably a linear time) algorithm for verifying if a given spanning tree is an MST.

4. ** Matrix chain product ** Given a chain \((A_1, A_2, \ldots, A_n)\) of matrices where matrix \(A_i\) has dimensions \(p_{i-1} \times p_i\), we want to compute the product of the chain using minimum number of multiplications.

   (i) In how many ways can the matrices be multiplied ?

   (ii) Design an efficient algorithm that does not exhaustively use (i).

5. Given two character strings \(x[1..n]\) and \(y[1..m]\), the edit distance is the cost of transforming the string \(x\) to \(y\) using a minimum number of operations from the set \{copy, replace, insert, delete\}. Design an efficient algorithm to find the minimum edit distance between two given strings.

   What if there are specific costs associated with each of the operation and you want to minimize the total cost ? This has direct application to DNA sequencing problem, i.e. how close they are to eachother.

6. Typesetting problem The input is a sequence of \(n\) words of lengths \(l_1, l_2, \ldots, l_n\) measured in characters. We want to print it nicely on a number of lines that can hold a maximum of \(M\) characters each. The criterion for "niceness" is as follows. No word can be split across lines with a blank separating words and each line should be as full as possible. The penalty for a trailing space of \(s\) is \(s^3\). If \(s_i\) is the space left in line \(i\), we want to minimize \(\sum_i s_i^3\).

   If the penalty function is \(\sum_i s_i\), would a greedy approach work ?

7. Given two strings \(s_1\) and \(s_2\) of lengths \(m\) and \(n\), find the longest common subsequence.

8. Optimal BST We are given a sequence \(K = \{k_1, k_2, \ldots, k_n\}\) of \(n\) distinct keys in sorted order with associated probability \(p_i\) that the key \(k_i\) will be accessed. Moreover, let \(q_i\) represent the probability that the search will be for a value (strictly) between \(k_i\) and \(k_{i+1}\). So \(\sum_i p_i + \sum_j q_j = 1\). How would you build the tree so as to optimise the expected search cost ? (The more probable value should be closer to the root.)

9. Given an NFA, how do you find out an equivalent regular expression ?

10. There are \(n\) destinations \(D_i\), \(1 \leq i \leq n\) with demands \(d_i\). There are two warehouses \(W_1, W_2\) that have inventory \(r_1\) and \(r_2\) respectively such that \(r_1 + r_2 = \sum_i d_i\). The cost of transporting \(x_{i,j}\) units from \(W_i\) to \(D_j\) is \(c_{i,j}(x_{i,j})\). We must ensure that \(x_{i,j} + x_{2,j} = d_j\) in a way so as to minimize \(\sum_i c_{i,j}(x_{i,j})\).

    Hint: Let \(g_i(x)\) be the cost incurred when \(W_1\) has an inventory of \(x\) and supplies are sent to \(D_j\) in an optimal manner - the inventory at \(W_2\) is \(\sum_{1 \leq j \leq n} d_j - x\).