Submission Instructions

a. This assignment is to be solved in teams of two students. Take our warning on plagiarism very seriously. We assume that from the moment of submission you are certifying that it is solely the work of your team, except where explicitly stated otherwise, and that each teammate can individually explain any part of the solution.

b. Write clear answers. Justify all answers, except where explicitly not required. State any assumptions you make. Thoroughly spell-check and grammar-check your answers. Meticulously follow all instructions in this document and at the homework web-page.

c. Document each algorithm according to the coding convention of the course.

d. Name your report LastName$_1$.FirstName$_1$ − LastName$_2$.FirstName$_2$.pdf, but without using any special characters. The report must be in PDF format and must have the cover information of the demo assignment report.

e. Submit your report via the Course Manager server (whose clock may differ from yours) by the deadline given above.

f. One teammate must use the group submission feature of the Course Manager, and the other teammate must confirm the submission via the Course Manager before the deadline.

Failure to follow the instructions above may result in 0 (zero) points, and we reserve the right to process your solutions automatically.
Question 1: Reliable Communications (40 points)

A communication network consists of a set of stations (which may act either as transmitters or as receivers) and some communication channels (e.g., wires) between pairs of them. The network can be modelled as an undirected graph \( G = (V, E) \), where \( V \) is the set of stations and \( E \) is the set of available communication channels. For stations \( u, v \in V \), communications along the channel \((u, v) \in E\) has a probability \( f(u, v) \) of failure, where \( 0 \leq f(u, v) \leq 1 \). As an example, if \( f(v_1, v_5) = 0.2 \), then the probability that a transmission along the channel connecting \( v_1 \) and \( v_5 \) will fail is 20%. All such probabilities are independent.

In order to let two stations \( u \) and \( v \) communicate with each other, a path \( \langle u = v_0, v_1, \ldots, v_n = v \rangle \) in the graph connecting \( u \) and \( v \) must be chosen.

a. Give pseudocode for an efficient algorithm to find a most reliable path between two given stations, i.e., a path having the lowest probability of failure.

b. Analyse the worst-case running time of your algorithm.

(Hint: Given a path \( \langle v_0, v_1, \ldots, v_n \rangle \) in \( G \), in case the probabilities of failure on the different edges are independent, the probability that the overall communication between \( v_0 \) and \( v_n \) fails is given by \( 1 - (1 - f(v_0, v_1)) \cdot \ldots \cdot (1 - f(v_{n-1}, v_n)) \).)

Question 2: Party Seating Problem (40 points)

A group of \( n \) guests will attend a party. Some of the guests already know each other, but no one at the party knows everyone else. To encourage guests to meet new people, the organisers would like to seat all of the guests at two tables, arranged in such a way that no guest knows any other guest seated at the same table.

a. Formulate the two-table party seating problem as a graph problem. For each guest \( g \in \{1, \ldots, n\} \), you are given a list of the other guests known by \( g \); these lists are symmetric in the sense that if \( g_1 \) knows \( g_2 \), then \( g_2 \) also knows \( g_1 \). The sum of the lengths of these lists is denoted by \( k \).

b. Give pseudocode for an \( O(n + k) \) algorithm that determines whether or not a given set of guests may be seated at two tables, such that no guest knows any other guest seated at the same table. If such an arrangement exists, the algorithm should return it in the form of two lists of guests, one list corresponding to each table.

c. Argue that your algorithm is in fact \( O(n + k) \).

d. We can generalise the party seating problem to more than two tables in the following manner. Say the party is attended by \( p \) groups of guests, and that there are \( q \) tables. All members of a group know each other; no guest knows anyone outside his or her group. The sizes of the groups are stored in the array \( \text{Group}[1 \ldots p] \), and the sizes of the tables in the array \( \text{Table}[1 \ldots q] \). Once again, we wish to find a seating arrangement such that at most one member of any group is seated at the same table. Give a formulation of this generalised party seating problem as a maximum flow problem. (You do not need to provide pseudocode for this part of the question.)

Question 3: Controlling the Maximum Flow (20 points)

An edge of a flow network is called sensitive if decreasing the capacity of this edge results in a decrease in the maximum flow. Give pseudocode for an efficient algorithm that finds a sensitive edge in a network, and an analysis of the complexity of your algorithm.