**Solution 11.1  Max-Flow by Hand.**

The value of a maximum flow is 43. A maximum flow is shown in the following figure. Next to each edge $e$, the capacity $c_e$ and the flow value $x_e$ are denoted in the order $x_e, c_e$. The minimum cut is indicated by the dotted line, the belonging edges are bold.

![Max-Flow Diagram](image)

The following figure shows the residual graph for the previous flow. Each edge is labeled with its residual capacity ($c_e - x_e$ oder $x_e$).

![Residual Graph](image)

**Solution 11.2  Championship Problem.**

Imagine that FCL wins both remaining games against FCZ and GCZ having enough goals. Then FCL has 37 points. To win the championship, FCSG may not receive any additional points, YB at most one, FCB at most two, and FCW at most six. The central observation for the construction of the network is that exactly two points are divided in each of the remaining games: if there is a winner, he receives both points; in the case of a tie each participant gets exactly one point. Thus, for each of the games FCSG-FCB, YB-FCW, FCW-FCSG, and FCB-YB we create a node in the network with an incoming edge from a source $s$ and two outgoing edges to the corresponding participants of the game. Each of these edges has a capacity of 2. From each participant, we generate an edge to a sink $t$ and set the capacity to the maximum number of points such that FCL still wins the championship. This leads to the following network.
FCL can only win the championship if there exists a flow of value 8 in the above network, because in each of the four remaining games, two points are spread among the corresponding participants. The max-flow min-cut theorem states that the value of a maximum flow is bounded by the value of a minimum cut. In the above network, there exists the following cut of value 7:

For this reason, no flow of value 8 exists. Thus, FCL cannot win the championship even if they win both the remaining games.

Solution 11.3 Assigning Students to Courses.

a) The set of nodes $V$ consists in a source $s$, a sink $t$, a node $s_i$ for each student $i \in \{1, \ldots, n\}$, and a node $k_j$ for each course $j \in \{1, \ldots, m\}$. We create the following edges:

- For each student $i \in \{1, \ldots, n\}$ there is an edge $(s, s_i)$ with capacity 5.
- There is an edge $(s_i, k_j)$ if and only if the student $i$ is interested in the course $j$, i.e., if $j \in K_i$. The capacity of each of these edges is set to 1.
- For each course $j \in \{1, \ldots, m\}$ there is an edge $(k_j, t)$ with capacity $T_j$.

Thus, we construct a network $N = (V, E, c)$ with

- $V := \{s\} \cup \{s_i \mid i \in \{1, \ldots, n\}\} \cup \{k_j \mid j \in \{1, \ldots, m\}\} \cup \{t\}$
- $E := \{(s, s_i) \mid i \in \{1, \ldots, n\}\} \cup \{(s_i, k_j) \mid i \in \{1, \ldots, n\}, j \in K_i\} \cup \{(k_j, t) \mid j \in \{1, \ldots, m\}\}$
- $c((s, s_i)) := 5 \quad \forall i \in \{1, \ldots, n\}$
- $c((s_i, k_j)) := 1 \quad \forall i \in \{1, \ldots, n\}, j \in K_i$
- $c((k_j, t)) := T_j \quad \forall j \in \{1, \ldots, m\}$

A possible distribution of the students exists if and only if there is a flow with value exactly $5n$ in the above network $N$.

b) We first analyze the size of the network $N$ in dependency of $n$ and $m$. Since there are more students than courses, we have $m < n$. In addition to the source and the sink, there are $n$
nodes for students and \( m \) nodes for the courses, thus \(|V| = 2 + m + n = \Theta(n)\). Each node \( s_i \), representing one of the students, has one incoming edge and 10 outgoing edges. For each node \( k_j \), representing a course, there is one more outgoing edge, so \(|E| = 11n + m = \Theta(n)\). Thus, the size of the network \( N \) is linear in \( n \). Moreover, the value of each flow in \( N \) is bounded from above by \( 5n \).

There are various algorithms to find a maximum flow. For example, the algorithm of Ford and Fulkerson calculates a maximum flow with \( O(|E| \cdot \phi^*) \) operations if \( \phi^* \) is the maximum flow value. So its running time is pseudopolynomial, and in general there are more efficient methods to compute a maximum flow. However, here we have both \(|E| = \Theta(n)\) and \( \phi^* = 5n = \Theta(n) \), and thus the running time of the algorithm of Ford and Fulkerson is only \( O(n^2) \), and the choice of this method is optimal \textit{in this particular case}.

c) In the network \( N \), a breadth-first search from a node \( s \) can be performed in order to find the nodes of the students \( s_i, i \in \{1, \ldots, n\} \). From these, there are only edges to courses \( k_j, j \in \{1, \ldots, m\} \). For these edges \((s_i, k_j)\), the flow value \( \phi((s_i, k_j)) \) can be examined: if it is 1 the student \( i \) takes the course \( k_j \) and we output the pair \((i, j)\). Otherwise, if the value is 0, then the student \( i \) does not take the course. In this manner, all courses of every student are found, and the running time is \( \Theta(|V| + |E|) = \Theta(n) \).

**Solution 11.4**  
\textbf{Matchings.}

a) Different solutions are possible, for example:

b) Halls’s theorem states that there is no perfect matching in a bipartite graph if there exists a set \( M \) of vertices with \(|\Gamma(M)| < |M|\), where \( \Gamma(M) \) is the set of neighbors of the vertices in \( M \). In our example, such a set is \( M = \{b, d, e\} \) with \( \Gamma(M) = \{h, i\} \). Another possibility is \( M = \{f, g, j\} \) with \( \Gamma(M) = \{a, c\} \). By Hall’s theorem, each of these sets proves that the graph has no perfect matching.