Name: Andrii Bezrukov **UČO:** 570943

In this assignment your task is to solve the given problem by reducing to the maximum flow problem.

(a) Assume each student can borrow at most 10 books from the library, and the library has three copies of each title in its inventory. Each student submits a list of books he wishes to borrow. You have to assign books to students to check out a maximum number of volumes.
Model the problem as a flow problem. For this purpose, describe the construction of an appropriate network $N = (V, E, c, s, t)$. How can you deduce from the maximum flow value what is the maximum number of volumes that can be checked out? Justify your answer.

(b) Assume you are planning the daily working times during vacation periods in a hospital. Let T be the set of all vacation days. There exist k vacation periods $\{1, \ldots, k\}$ consisting of one or more contiguous vacation days $T_j \subseteq T$ (for $j \in \{1, \ldots, k\}$). Every day in T is contained in exactly one vacation period, i.e., we especially have $T_i \cap T_j = \emptyset$ for $i \neq j$. In the hospital, there are n doctors. Every doctor i specifies a set of vacation days $S_i \subseteq T$ on which he could be present. The goal is to design an operational plan that distributes the vacation days T among the doctors such that on every day there is exactly one doctor present. Additionally, we require that no doctor works on more than $C \in \mathbb{N}$ many vacation days overall and that, for every $j \in \{1, \ldots, k\}$, every doctor is assigned at most one day of the vacation period T_j.
Model the problems as a flow problem. For this purpose, describe the construction of an appropriate network $N = (V, E, c, s, t)$. How can you deduce from the value of a maximum flow whether an appropriate assignment exists, or not? In case that an appropriate assignment exists, how can you deduce such an assignment? Justify your answer.

1 question (a)

We model the library problem as a single-source, single-sink flow network

$$N = (V, E, c, s, t).$$

The vertex set is

$$V = \{s\} \cup \{u_i \mid i = 1, \ldots, n\} \cup \{v_j \mid j = 1, \ldots, m\} \cup \{t\},$$

where s is the source, each u_i represents student i, each v_j represents book title j, and t is the sink. We place three types of directed edges: from the source to each student, from each student to the titles on his or her wish list, and from each title to the sink.
The capacity function c is given by

$$c(s, u_i) = 10 \quad \text{for all } i,$$

so that no student can borrow more than ten volumes in total;

$$c(u_i, v_j) = 1 \quad \text{whenever student } i \text{ wants title } j,$$

so that each student may take at most one copy of any given title; and

$$c(v_j, t) = 3 \quad \text{for all } j,$$

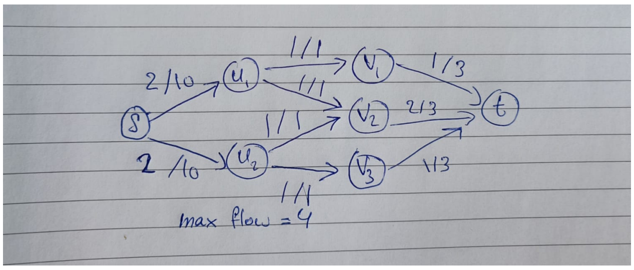

Figure 1

so that no title can be checked out more than its three available copies. There are no other edges in the network.
A unit of flow along the path

$$s \longrightarrow u_i \longrightarrow v_j \longrightarrow t$$

corresponds exactly to assigning one copy of title j to student i. Because all capacities are integers, the max-flow/min-cut theorem guarantees an integral maximum flow f^*, meaning every edge carries an integer amount of flow. The value $|f^*|$, defined as the total flow leaving s (equivalently, entering t), therefore counts exactly the number of individual book-copy assignments.
Hence, the maximum flow value

$$|f^*|$$

is exactly the maximum number of volumes that can be checked out under the given constraints. Each unit of flow represents one volume loaned, and the capacities enforce the “at most 10 per student” and “at most 3 per title” limits.

2 question (b)

2.1 Main Idea

Assume we have a set of vacation days T, which are divided into k contiguous, non-overlapping periods $T_1, \dots, T_k \subseteq T$. There are n doctors, each labeled i from 1 to n, and doctor i is available on a subset $S_i \subseteq T$. We want to choose exactly one doctor for each day in T, making sure no doctor works more than C days in total and also no doctor works more than one day in any given period T_j.

2.2 Network Model

To capture all these conditions at once, we build a single circulation-with-demands network $N = (V, E, s, t)$ with a source s, a sink t, and the following nodes and arcs:
First, for each doctor i we add a node u_i and connect s to u_i with an arc of capacity C; this enforces the rule that doctor i can't be assigned more than C days overall. Then for each doctor i and each period j, we create an intermediate node $w_{i,j}$ and link u_i to $w_{i,j}$ with capacity 1, so that doctor i can send at most one unit of flow (one work-day) into period T_j. From $w_{i,j}$ we draw edges to the day-nodes v_d for every day d in the intersection $S_i \cap T_j$, each of capacity 1; flow on $(w_{i,j}, v_d)$ means we've assigned doctor i to day d within period j.

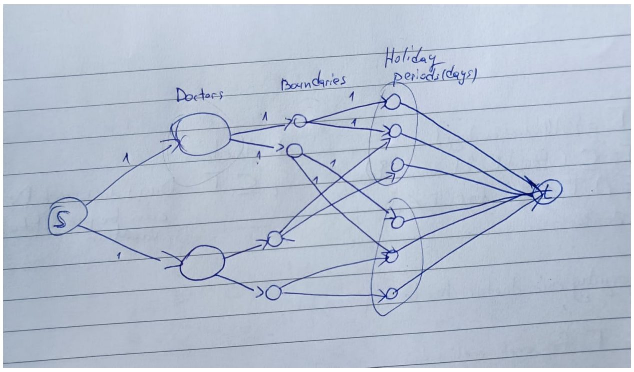

Figure 2

Every day d in T gets its own node v_d, and we attach an edge (v_d, t) with both its lower and upper bounds set to 1. This setup forces exactly one unit of flow through each day-node, guaranteeing one doctor per day. Finally, since the total number of days is $D = |T|$, we give the source a demand of $-D$ and the sink a demand of $+D$, with all other nodes having zero demand.

2.3 Feasibility and Schedule Extraction

In this unified construction, finding a feasible circulation that respects all capacities, bounds, and demands is equivalent to finding a valid schedule: one doctor per day, each doctor no more than C days in total, and at most one day in each vacation period, while honoring the doctors' availability. To decide if such a circulation exists, we apply the standard reduction to a max-flow problem for networks with lower bounds; if the resulting max-flow saturates the forced arcs, we have a solution, and the nonzero flows on the arcs $(w_{i,j}, v_d)$ tell us exactly who works which day.