Exercise Set I, Algorithms II

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 *(easy)* Show that, given a matroid $\mathcal{M} = (E, \mathcal{I})$ and a weight function $w : E \to \mathbb{R}$, Greedy (as defined in the lecture notes) always returns a base of the matroid.

2 Given a matroid $\mathcal{M} = (E, \mathcal{I})$ and a weight function $w : E \to \mathbb{R}$, Greedy for matroids returns a base $S = \{s_1, s_2, \ldots, s_k\}$ of maximum weight. As noted in the lecture notes, any base consists of the same number, say k, of elements (which is said to be the rank of the matroid). We further assume that the elements of S are indexed so that $w(s_1) \geq w(s_2) \geq \cdots \geq w(s_k)$.

Let $S_\ell = \{s_1, \ldots, s_\ell\}$ be the subset of S consisting of the ℓ first elements, for $\ell = 1, \ldots, k$. Then prove that

$$w(S_\ell) = \max_{T \in \mathcal{I} : |T| = \ell} w(T) \text{ for all } \ell = 1, \ldots, k.$$

In other words, Greedy does not only returns a base of maximum weight but the "prefixes" are maximum weight sets of respective cardinalities.

3 *(easy)* Recall that a matroid $\mathcal{M} = (E, \mathcal{I})$ is a partition matroid if E is partitioned into *disjoint* sets $E_1, E_2, \ldots, E_\ell$ and

$$\mathcal{I} = \{X \subseteq E : |E_i \cap X| \leq k_i \text{ for } i = 1, 2, \ldots, \ell\}.$$

Verify that this is indeed a matroid.

4 *(half a *)* Consider a bipartite graph $G = (V, E)$ where V is partitioned into A and B. Let $(A, \mathcal{I})$ be the matroid with ground set A and

$$\mathcal{I} = \{A' \subseteq A : \ G \text{ has a matching in which every vertex of } A' \text{ is matched}\}.$$

Recall that we say that a vertex is matched by a matching M if there is an edge in M incident to v. Show that $(A, \mathcal{I})$ is indeed a matroid by verifying the two axioms.

5

5a *(*)* Consider a family $\mathcal{F}$ of subsets of the ground set E that satisfies: if $X, Y \in \mathcal{F}$ then either $X \cap Y = \emptyset$ (they are disjoint), $X \subseteq Y$ (X is a subset of Y), or $Y \subseteq X$ (Y is a subset of X). Show that for any positive integers $\{k_X\}_{X \in \mathcal{F}}$ (one for each set in $\mathcal{F}$) we have that $\mathcal{M} = (E, \mathcal{I})$ is a matroid, where

$$\mathcal{I} = \{S \subseteq E : |S \cap X| \leq k_X \text{ for every } X \in \mathcal{F}\}.$$

Such a matroid is called a laminar matroid.

5b Argh! Buying the DVD rental shop was not such a great idea. After the explosion of more convenient streaming services, you are now forced to close your business venture. But what should you do with all your DVDs? To be exact, you have n DVDs and each one is placed in one of the following genres: action, comedy, drama, horror or adventure. As you are a very nice person, you decide to distribute these DVDs among your most loyal customers. You have m loyal customers and for each DVD i and customer j there is a positive weight $w(i, j)$ that models how interesting DVD i is for customer j. Your goal is to find an assignment of DVDs to loyal customers satisfying the following:

- Each DVD is assigned to at most one customer.
- Each customer receives at most 5 DVDs in total and no more than 2 DVDs of the same genre.
- The total weight (called the social welfare) of your assignment is maximized.

Show that the problem of distributing the DVDs as above can be formulated as that of finding a maximum weight independent set in the intersection of two matroids.

6 Spanning trees with colors.

Consider the following problem where we are given an edge-colored graph and we wish to find a spanning tree that contains a specified number of edges of each color:

Input: A connected undirected graph $G = (V, E)$ where the edges E are partitioned into k color classes $E_1, E_2, \ldots, E_k$. In addition each color class i has a target number $t_i \in \mathbb{N}$.

Output: If possible, a spanning tree $T \subseteq E$ of the graph satisfying the color requirements:

$$|T \cap E_i| = t_i \qquad \text{for } i = 1, \ldots, k.$$

Otherwise, i.e., if no such spanning tree T exists, output that no solution exists.

Design a polynomial time algorithm for the above problem. You should analyze the correctness of your algorithm, i.e., why it finds a solution if possible. To do so, you are allowed to use algorithms and results seen in class without reexplaining them.