COL351 Holi2023: Tutorial Problem Set 7

1. A maximal independent set (that is, an independent set none of whose proper supersets is independent) of a matroid is called a *basis* of a matroid. Prove that all bases of a matroid have the same cardinality. (This cardinality is called the *rank* of the matroid.) Let $\mathcal{B}$ be the collection of bases of a matroid $M = (S, \mathcal{I})$. Prove that

   (a) $\mathcal{B} \neq \emptyset$.

   (b) For all $B_1, B_2 \in \mathcal{B}$ and for all $b_1 \in B_1 \setminus B_2$, there exists a $b_2 \in B_2$ such that $(B_2 \setminus \{b_2\}) \cup \{b_1\} \in \mathcal{B}$.

   Conversely, let $\mathcal{B} \subseteq 2^S$ be a collection of subsets of S satisfying the above properties. Prove that $\mathcal{B}$ is the collection of bases of some matroid M with ground set S.

2. Let $M = (S, \mathcal{I})$ be a matroid and let w_e denote the weight of every element $e \in S$ (possibly some w_e's can be negative). Prove that Kruskal's algorithm returns a maximum weight basis of the matroid.

3. Let M be a matroid over the ground set S and let $\mathcal{B}$ be the set of its bases. Let $\overline{B} = \{S \setminus B \mid B \in \mathcal{B}\}$. Prove that $\overline{B}$ is the set of bases of a matroid over the ground set S. (This matroid is called the *dual* of M. Let us denote it by $\overline{M}$. Observe that $\overline{\overline{M}} = M$.)

4. Let $M = (S, \mathcal{I})$ be a matroid and e be an element of S. Prove that $(S \setminus \{e\}, \{I \mid I \in \mathcal{I} \text{ and } e \notin I\})$ is a matroid. This matroid is denoted by $M \setminus e$, and the operation of transforming M to $M \setminus e$ is called *deletion* of e. The matroid $\overline{\overline{M} \setminus e}$ is denoted by M/e and the operation of transforming M to M/e is called *contraction* of e. Prove that

$$M/e = \begin{cases} (S, \{I \setminus \{e\} \mid I \in \mathcal{I} \text{ and } e \in I\}) & \text{if } \{e\} \in \mathcal{I}, \\ M \setminus e & \text{otherwise.} \end{cases}$$

5. Let $G = (V, E)$ be a graph. A subset F of edges of G is called a *pseudoforest* of G if every connected component of F has no more edges that vertices. Design a polynomial-time algorithm which, given a graph G and weight $w_e \geq 0$ for every $e \in E$, computes a max-weight pseudoforest of G. Try getting an $O(m \log n)$ time algorithm, where $n = |V|$ and $m = |E|$.