COL351 Holi2023: Tutorial Problem Set 9

1. Recall from your discrete structures course that a multi-graph is said to be *regular* if the degrees of all its vertices are the same. Prove that every regular bipartite multi-graph with a non-empty edge set has a perfect matching. (Observe that the proof of Hall's theorem works for multi-graphs too.)

2. A standard set of 52 playing cards is distributed into 13 piles of 4 cards each, in an arbitrary manner. Prove that there exists a subset S of 13 cards which satisfies both of the following conditions.

   (a) S contains exactly one card from each of the 13 piles.

   (b) S contains exactly one card of each rank (one Ace, one King, one Queen, . . . , one Two).

3. Let $G = (L, R, E)$ be a bipartite graph. We say that a set $L' \subseteq L$ is *matchable* in G if there exists a matching in G in which all vertices in L' are matched. Let $\mathcal{I} \subseteq 2^L$ be defined as $\mathcal{I} = \{L' \mid L' \text{ is matchable in } G\}$. Prove that $(L, \mathcal{I})$ is a matroid. Such a matroid which can be represented as the collection of matchable subsets in some bipartite graph is called a *transversal matroid*.

4. Design a polynomial-time algorithm that, given a weighted directed graph, finds a negative weight cycle if one such cycle exists; else returns "NO". (This has nothing to do with flows. One way is to modify the Floyd-Warshall algorithm.)

5. Design an algorithm which, given a bipartite graph with a non-negative integer weight for every edge, finds a maximum weight matching in it. For simplicity, assume that the graph is a complete bipartite graph. Your algorithm must run in time polynomial in the size of the graph and the value of the optimum.

6. Recall problem 5 from tutorial 5, where the task was to plan your work on assignments to minimize total penalty. Assume that the per-day penalties are all non-negative integers. You are now in a position to attack that problem. Attack and solve it to get an algorithm which runs in time polynomial in the number of assignments and optimum penalty.

7. Let $G = (V, E)$ be a directed graph such that for every $u, v \in V$, $(u, v) \in E$ if and only if $(v, u) \in E$. (You will again realize that this assumption is without loss of generality, but it simplifies your life.) Let $C : E \longrightarrow \mathbb{R}_{\geq 0}$ be an assignment of non-negative capacities to the edges of G. A *circulation* in (G, C) is a skew-symmetric function $f : E \longrightarrow \mathbb{R}$ such that for all $v \in V$, $\sum_{u:(u,v)\in E} f(u, v) = 0$, and for all $(u, v) \in V$, $f(u, v) \leq C(v, u)$. Given a skew-symmetric weight assignment $w : E \longrightarrow \mathbb{R}$, the weight of any function $f : E \longrightarrow \mathbb{R}$, denoted by $w \cdot f$, is defined to be $\sum_{e\in E} w(e)f(e)$. Prove that for all G and w, G has a negative weight circulation if and only if G has a negative weight cycle consisting of positive capacity edges.

8. Let $G = (V, E)$ be a directed graph, $C : E \longrightarrow \mathbb{R}_{\geq 0}$ be an assignment of capacities, and $w : E \longrightarrow \mathbb{Z}$ be a skew-symmetric assignment of integral weights to its edges. Design algorithms which, given this information, compute the following.

   (a) Given vertices $s, t \in V$ additionally, a minimum weight $s - t$ maxflow.

   (b) Given demands $d(v)$ for each vertex v (such that $\sum_v d(v) = 0$), a minimum weight demand circulation.

   Your algorithms must run time polynomial in the size of the graph and the value of the optimum.