CSE421: Introduction to Algorithms

Homework 8

Due: November 28, 2018

1. Given two disjoint sets X and Y. Our goal is to choose the largest possible collection of pairs (x, y) where $x \in X$, $y \in Y$, subject to the following constraints:

   - Each element $x \in X$ can appear in at most $c(x)$ pairs.
   - Each element $y \in Y$ can appear in at most $c(y)$ pairs.
   - Each pair $(x, y) \in X \times Y$ can appear (repeatedly) in the output at most $c(x, y)$ times.

   where $c(x)$, $c(y)$ and $c(x, y)$ are non-negative integers.

   Give a polynomial time algorithm to find the number of pairs. Prove the correctness and the runtime of the algorithm.

2. Consider the problem of perfectly tiling a subset of a checkerboard (i.e. a collection of unit squares, see example below) with dominoes (a domino being 2 adjacent squares). Give a polynomial time algorithm to decide whether there is a perfectly tiling.

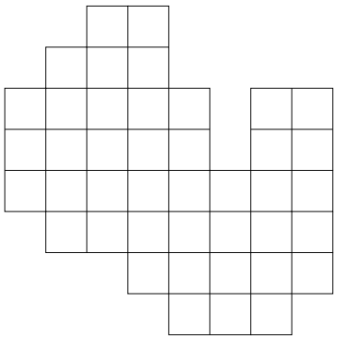

Figure 1: Example of a subset of a checkerboard.

3. Let $G = (V, E)$ be an undirected graph and let $w(e) \geq 0$ for edge weights for $e \in E$. Recall that a subset $M \subseteq E$ is called a *matching* if the edges in M do not share a node. The *Maximum Weight Matching* problem consists of finding the matching M so that the weight $w(M) := \sum_{e \in M} w(e)$ is maximized. This is indeed a polynomial time solvable problem, but the algorithm is highly non-trivial and way beyond the scope of our lecture. But it is easy to design a greedy algorithm that finds 2-approximation. To be precise we suggest the following algorithm:

   (1) Sort the edges so that $w(e_1) \geq w(e_2) \geq \ldots \geq w(e_m)$
   (2) Set $M := \emptyset$
   (3) FOR $i = 1$ TO m DO
      (4) If $M \cup \{e_i\}$ is still a matching THEN update $M := M \cup \{e_i\}$

Let us denote $\texttt{OPT} := \max\{w(M) : M \subseteq E \text{ is matching}\}$ as the value of the optimum solution and $\texttt{GREEDY}$ as the value of the solution produced by the greedy algorithm above. Solve the following:

(a) Show that for any $\varepsilon > 0$ there is an instance where $\texttt{GREEDY} \leq (\frac{1}{2} + \varepsilon) \cdot \texttt{OPT}$.

(b) Suppose that $\{a_1, \ldots, a_p\} \subseteq E$ are the edges selected by greedy and suppose that $\{b_1, \ldots, b_q\} \subseteq E$ are the edges selected by the optimum solution. Moreover suppose those edges are sorted so that $w(a_1) \geq w(a_2) \geq \ldots \geq w(a_p)$ and $w(b_1) \geq w(b_2) \geq \ldots \geq w(b_q)$. For the sake of simplicity, assume that q is even. Show that $w(a_i) \geq w(b_{2i-1})$ for all $i \in \{1, \ldots, \frac{q}{2}\}$.

(c) Prove that $\texttt{GREEDY} \geq \frac{1}{2} \cdot \texttt{OPT}$.

4. **Extra Credit:** Suppose we are given a directed network $G = (V, E)$ with a root node r and a set of terminals $T \subset V$. We'd like to disconnect many terminals from r, while cutting relatively few edges.

   We make this trade-off precise as follows. For a set of edges $F \subset E$, let $q(F)$ denote the number of nodes $v \in T$ such that there is no r-v path in the subgraph $(V, E - F)$. Give a polynomial-time algorithm to find a set F of edges that maximizes the quantity $q(F) - |F|$. (Note that setting F equal to the empty set is an option.)