Ma/CS 6a: Problem Set 5

Due noon, Thursday, November 6

1. Consider a connected undirected bipartite graph $G = (V_1 \cup V_2, E)$, such that $|V_1| = |V_2|$. Also, for every subset $A \subset V_1$ we have $|A| < |N(A)|$ (except for the trivial cases $A = \emptyset$ and $A = V_1$). Prove that for every edge $e \in E$, there exists a perfect matching of G that contains e.

2. We are given a set of $n \cdot m$ cards, which consists of m distinct types of cards, each appearing n times. We shuffle the cards and deal them into n rows, each containing m cards (that is, into an $n \times m$ matrix). Prove that there exists a set of m cards such that each is of a different type and each is in a different column (hint: don't let parallel edges scare you too much).

3. When Biscuit is not playing Dominoes, he likes playing the following game with Adam. They are given a connected undirected bipartite graph $G = (V_1 \cup V_2, E)$, such that $|V_1| = |V_2|$. Biscuit is always first, and he starts by choosing a vertex $v_1 \in V_1 \cup V_2$. Adam then travels from v_1 across one edge to a vertex v_2 (that is, $(v_1, v_2) \in E$). Then Biscuit travels from v_2 across one edge to a vertex v_3. The players continue to travel the graph in this way, and they are not allowed to visit vertices that already appear in the path the is created by these travels. The player that gets stuck without a valid edge to cross loses, and has to do the laundry for the week.

Prove that if G does not contain a perfect matching, Biscuit has a winning strategy. That is, using this strategy Biscuit wins no matter what Adam does (hint: Biscuit should start by finding a maximum matching M, and choosing a vertex that is not matched in M).

4. Let (V, E, s, t, c) be a flow network. Let $e \in E$ be the only edge in the network with a capacity that is not a multiple of five. Prove or disprove: if the size of the maximum flow is not a multiple of five, then every maximum flow of the network has a non-zero flow through e.

5. Let $G = (V, E)$ be a directed graph, and let $s, t \in V$. A subset $S \subset V$ is said to be a *disconnecting set* if after the removal of the vertices of S from G (together with the edges that are adjacent to them) there are no paths from s to t. Describe an efficient algorithm for finding a a set of disconnecting vertices of a minimum size. Prove that your algorithm is correct.