22C:137/22M:152 Homework 2

Due: Tuesday, 4/26

Notes: (a) Solve all 8 problems listed below. We will grade some 5-subset of these. (b) It is possible that solutions to some of these problems are available to you via other graph theory books or on-line lecture notes, etc. If you use any such sources, please acknowledge these in your homework. You will benefit most from the homework, if you sincerely attempt each problem on your own first, before seeking other sources. (c) It is okay to discuss these problems with your classmates. Just make sure that you take no written material away from these discussions.

1. Problem 5.32 (iv), Section 5.5, page 35, Schrijver's notes.

2. Problem 5.36, Section 5.5, page 35, Schrijver's notes.

3. Problem 5.37, Section 5.5, page 35, Schrijver's notes.
   Let $G = (V, E)$ be a connected graph and let $k \geq 0$. Show that there exist k pairwise edge-disjoint spanning trees, iff for each t, for each partition $(V_1, V_2, \ldots, V_t)$ of V into t classes, there exist at least $k(t-1)$ edges connecting different classes of this partition.
   Solution: ($\Rightarrow$) Suppose G has k edge disjoint spanning trees. Each spanning tree is a basis in the cycle matroid $M(G)$ of G. Thus $M(G)$ has k pairwise disjoint basis. Using the claim in 5.36 we conclude that
   $$k(r_M(E) - r_M(E')) \leq |E - E'| \tag{1}$$
   for any $E' \subseteq E$. Now fix a t and a partition $(V_1, V_2, \ldots, V_t)$ of V. Let E_i be the edges in the induced subgraph $G[E_i]$ and let $E' = \cup_{i=1}^{t} E_i$. Now note that E' induces t connected components in G and therefore $r_M(E') = n - t$, where $n = |V|$. Also, $r_M(E) = n - 1$. This implies that the L.H.S. of inequality (1) is $k(t-1)$. Also, note that the R.H.S. of inequality (1) equals the number of edges in G that connect different classes in the partition $(V_1, V_2, \ldots, V_t)$. Since the choice of t and the choice of the partition $(V_1, V_2, \ldots, V_t)$ are arbitrary, this implies that for each t and for each partition $(V_1, V_2, \ldots, V_t)$ of V into t classes, there exist at least $k(t-1)$ edges connecting different classes of this partition.

   ($\Leftarrow$) Suppose that for each t and for each partition $(V_1, V_2, \ldots, V_t)$ of V into t classes, there exist at least $k(t-1)$ edges connecting different classes of this partition. Consider and arbitrary subset of edges $E' \subseteq E$ and suppose that E' induces $t \geq 1$ connected components. Let $S \subseteq E$ be the set of edge connecting different connected components of $G[E']$. Note that $|E - E'| \geq |S|$. Also, we have that $|S| \geq k(t-1)$ and therefore $|E - E'| \geq k(t-1)$. As shown earlier in the proof, $k(r_M(E) - r_M(E')) = k(t-1)$. and therefore we have $|E - E'| \geq k(r_M(E) - r_M(E'))$. By 5.36 this means that $M(G)$ has k pairwise disjoint basis, which is another way of saying that G has k pairwise edge disjoint spanning trees.

4. Let $D = (V, A)$ be a directed graph. For any $r \in V$, an *r-arborescence* is a subgraph $D' = (V, A')$ such that (i) $|A'| = |V| - 1$ and (ii) there is a directed path from r to every vertex in V. Show that given a directed graph $D = (V, A)$ and a weight function $w : A \to \mathbf{R}^+$ there is a polynomial time algorithm to find a mimimum weight arborescence in D.

Hint: Use the fact that the heaviest common independent set of two weighted matroids can be found in polynomial time.

Solution: Let $M(D) = (A, \mathcal{I}_M)$ be the cycle matroid of D. Note that each independent set of $M(D)$ is a set of arcs $A' \subseteq A$ such that the underlying undirected graph of $D[A']$ is acyclic.

Without loss of generality, let $V = \{1, 2, \ldots, n\}$. For each $i \in V$, let A_i be the set of edges in-coming into vertex i. Let $T(D) = (A, \mathcal{I}_T)$ be the transversal matroid with respect to the sets $A_1, A_2, \ldots, A_n$.

It is easy to verify that if I is an independent set common to $M(D)$ and $T(D)$, then $|I| \leq |V| - 1$ and if $|I| = |V| - 1$ then I is an r-arborescence of D for some $r \in V$. Thus using Edmond's matroid intersection algorithm, one can find in polynomial time if D has an r-arborescence for some $r \in V$. Furthermore, if the arcs of D have associated non-negative weights, then assuming that D has an r-arborescence, we can use Edmond's weighted matroid intersection algorithm to find in polynomial time an r-arborescence of maximum weight, for some $r \in V$. Finally, let $W = \max_{e \in A} w(e)$. To each arc $e \in A$, assign a new weight $w'(e) = W - w(e)$. Note that these weights are all non-negative and furthermore a heaviest r-arborescence with respect to the weight function w' is a lightest r-arborescence with respect to the weight function w. This gives a polynomial time algorithm to find a minimum weight r-arborescence in D, for some $r \in V$.

5. Problem 15, Chapter 5, page 118. This problem leads to a simple algorithm to compute a Δ-coloring for graphs that are not cliques or odd cycles.

   Solution (i): Let the greedy algorithm use the palette $\{1, \cdots, \Delta(G)\}$. The algorithm colors each vertex with the smallest available color. Suppose we have an ordering of the vertices $v_1, \cdots, v_n$ such that $degree(v_n) = \Delta(G)$, $\{v_1, v_n\}, \{v_2, v_n\} \in E(G)$, and $\{v_1, v_2\} \notin E(G)$. Then the algorithm colors v_1 and v_2 with 1. For any v_i, $i = 3, \cdots, n-1$, $degree(v_i) \leq \Delta(G)$, and there is at least one vertex v_j in $N(v_i)$ with $j > i$. Hence, when the algorithm colors v_i, we have colored at most $\Delta(G) - 1$ neighbors of v_i, leaving one color available for v_i. Since $degree(v_n) = \Delta(G)$, and v_1, v_2 are colored 1, $N(v_n)$ uses at most $\Delta(G) - 1$ colors, leaving at least one available color for v_n.

   Solution (ii) : Let $k = \Delta(G)$, and let v be a vertex of maximum degree. Let $u_1, \cdots, u_k$ be the neighbors of v. If $\{u_i, u_j\} \in E(G)$ for each $i, j \in \{1, \cdots, k\}$, then $u_1, \cdots, u_k, v$ form a clique. Hence for each vertex u_i, $degree(u_i) = k$. Since G is not complete, and is connected there is some vertex w adjacent to a vertex u_i. But this implies $degree(u_i) = k + 1$, contradicting the fact that k is the maximum degree. Hence, $u_1, \cdots, u_k, v$ is not a clique and therefore we must have two vertices u_i, u_j such that $\{u_i, u_j\} \notin E(G)$.

   Proof of Brooks' Theorem using lemmas (i) and (ii) : Let $G(V, E)$ be a graph satisfying the conditions of Brooks' theorem. Let v_n be a vertex of maximum degree. From Lemma (ii), there are two vertices v_1, v_2 adjacent to v_n such that $\{v_1, v_2\} \notin E(G)$.

   Now, consider a spanning tree of G rooted at v_n, where each vertex except v_1, v_2 are labeled in decreasing order of their distance from v_n. This gives us an ordering satisfying the conditions of Lemma (i), and from the proof of Lemma (i), we know that the greedy algorithm uses at most $\Delta(G)$ colors. Hence, $\chi(G) \leq \Delta(G)$ for graphs satisfying the conditions of Brooks' theorem.

It remains to be shown that it is sufficient to consider graphs with $\Delta(G) \geq 3$ and $\kappa(G) \geq 2$. If $\Delta(G) = 0, 1$, then the graph is either a single vertex or a single edge. In either case, the graph is complete. If $\Delta(G) = 2$, then the graph is either a path or a cycle. If the graph is a path or an even cycle, the graph is bipartite and hence $\chi(G) = 2$. Otherwise G is an odd cycle and doesn't satisfy the conditions of Brooks' theorem. To see that it is sufficient to consider 2 connected graphs, If $\kappa(G) = 1$, the graph has a cut vertex v. Apply Brooks' theorem inductively to the components of $G - v$, and reorder the color classes so that the color of v is the same in each component.

6. Problem 27, Chapter 5, page 119.

   Proof: Since K_2^r contains K^r as a subgraph, it follows that the choice number of K_2^r is at least r. We show that $ch(K_2^2) \leq r$ by induction.

   The base case K_2^1 consists of 2 disjoint vertices and hence is easily verified.

   For the inductive case, assume that $ch(K_2^{r-1}) = r - 1$. Let $A_1, \cdots, A_r$ be the r parts of the graph, and let u_i, v_i denote the two vertices in A_i. Let $L(v)$ denote the list associated with vertex v. Suppose there exists a partition A_i such that $L(u_i) \cap L(v_i) \neq \emptyset$. Let $c \in L(u_i) \cap L(v_i)$. Let $H = G \setminus A_i$. Set $L(v) = L(v) \setminus c$ for all vertices $v \in H$ such that $c \in L(v)$. From the inductive hypothesis, H has a proper vertex coloring that does not use the color c, and we can extend the coloring of H to a coloring of G by setting $color(u_i) = color(v_i) = c$.

   On the other hand if for each partition $L(u_i) \cap L(v_i) = \emptyset$. For any partition A_i, note that $N(u_i) = N(v_i)$ and $|N(u_i)| = |N(v_i) = 2r - 2$, where $N(v)$ is the neighborhood of vertex v. Since $L(u_i) \cap L(v_i) = \emptyset$, it follows that $|L(u_i) \cup L(v_i)| = 2r$. Hence, we must have 2 unused colors in $L(u_i) \cup L(v_i)$, say c, c'. If there exists a coloring of $H = G \setminus A_i$ where $c \in L(u_i)$ and $c' \in L(v_i)$, we can easily extend the coloring of H to a coloring of G. Hence, assume that in all colorings of H, both unused colors of $L(u_i) \cup L(v_i)$ belong to $L(u_i)$ wlog. This implies that in any proper coloring of H, for all vertices w such that $c'' = color(w) \in L(v_i)$, both $c \notin L(w)$ and $c' \notin L(w)$. i.e., there is no vertex w that is colored with a color $c'' \in L(v_i)$ and contains either c or c'. If this was the case, the following re-arrangement of colors allows us to extend a proper coloring of H to a proper coloring of G. Set $color(v_i) = c''$, $color(w) = c$ and $color(u_i) = c'$ (assuming wlog that $c \in L(w)$). This contradicts the condition that there is no coloring where we have one unused color in $L(u_i)$ and one unused color in $L(v_i)$.

   Now consider H with the following lists. Set $L(w) = L(w) \setminus c''$ for each $w \in H$ with $c'' \in L(w)$. Set $L(w) = L(w) \setminus c$ for each $w \in H$ with $c \in L(w)$. Note that the previous condition implies $|L(w)| \geq r - 1$ for each $w \in H$. Again, applying the inductive hypothesis, we have a proper vertex coloring of H. This can be extended to a proper coloring of G as follows. Since neither c'' nor c is used up for any vertex of H, set $color(u_i) = c$ and $color(v_i) = c''$. □

7. The *greedy algorithm* for graph coloring takes as input a graph G and an ordering σ of the vertices, processes the vertices according to σ, and to each vertex v assigns the smallest available color $i \in \mathbf{N}$.

(a) Prove that every graph G has a vertex ordering σ such that the greedy algorithm with input G and σ uses $\chi(G)$ colors.

(b) For all $k \in \mathbf{N}$, inductively construct a tree T_k with maximum degree k and an ordering σ_k of $V(T_k)$ such that greedy algorithm with input T_k and σ_k uses $k+1$ colors. Note that this shows that the performance ratio of the greedy algorithm may be as bad as $(\Delta(G)+1)/2$.

Solution : Let $c : V \rightarrow \mathcal{N}$ be a coloring G that uses exactly $\chi(G)$ colors. Let σ be an ordering that satisfies : if $c(u) < c(v)$ for any vertices $u, v \in V$, then $\sigma(u) < \sigma(v)$. Then, the greedy algorithm processes all vertices with $c(v) = 1$, followed by vertices with $c(v) = 2$, and so on, and each vertex recieves the same coloring as c. Hence the greedy algorithm uses exactly $\chi(G)$ colors.

8. For all $k \in \mathcal{N}$, inductively construct a tree T_k with maximum degree k and an ordering σ_k of $V(T_k)$ such that the greedy algorithm with input T_k and σ_k uses $k+1$ colors. Note that this shows that the performance of the greedy algorithm may be as bad as $(\Delta(G)+1)/2$.

   Solution : Let T_0 be a single vertex. Given $T_1, \cdots, T_{i-1}$, we construct T_i by adding a leaf to each vertex of T_{i-1}. Then, $\Delta(T_i) = i$. The ordering σ_k is defined as follows. The leaves of T_k come first in the order, followed by the leaves of T_{k-1}, and so on.

   The construction is such that the leaves of T_k is adjacent to each vertex of T_{k-1} for each $k = 1, 2, \cdots$. Hence, applying the greedy algorithm with the ordering σ_k defined above ensures that each vertex of T_i, $i = 0, \cdots, k-1$ is adjacent to at least one vertex of each color $1, 2, \cdots, k-i$. Hence the single vertex of T_0 is colored with color $k+1$. Since the chromatic number of a tree is 2, the performance of the greedy algorithm is atleast $(\Delta(G)+1)/2$.

9. Let G be the *unit distance graph* in the plane; the vertices are all (infinitely many) points in the plane, with vertices joined by an edge if the Euclidean distance between them is exactly 1. Prove that G is 7-colorable but not 3-colorable.

   Solution :

   The graph shown in Figure 1 is a unit-distance graph that is not 3-colorable. This easily follows since we require 3 colors for the pentagon outside and the two vertices are adjacent to vertices 3 vertices each, at least one set of which must use 3 colors.

   The graph shown in Figure 2 shows a coloring of the unit-distance graph using 7 colors.

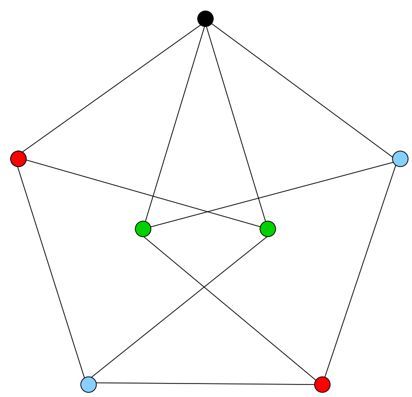

Figure 1: The Moser graph, a unit-distance graph whose chromatic number is 4

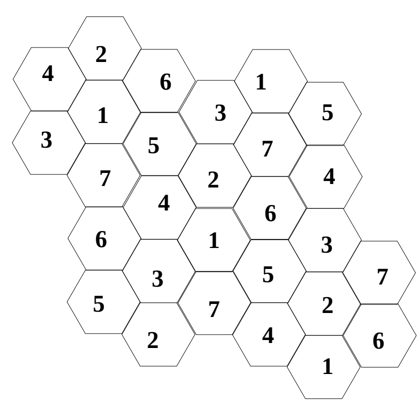

Figure 2: A periodic tiling of the plane with 7 colors. The diameter of the hexagon is < 1