CS240: Homework 10

Sahit Mandala

December 10, 2015

Problem 1

False. See attached for counter example.

Notice that that T is a tree and S is a compliment of T. Furthermore, notice by inspection that S is connected. However, we can see that the path (1,4),(4,5),(5,1) is in fact a cycle, which makes S not a tree.

Problem 2

The chromatic number of a tree is 2 (except for the tree on 1 vertex, an edge case, which clearly has chromatic number 1). That is, we can always color a tree with 2 or more vertices with 2 colors s.t. adjecent nodes have no edges between them.

To show this, we shall construct an algorithm which colors the tree with 2 colors and show that if this algorithm fails, we have a contradiction.

The algorithm: First, pick some arbitrary node in the tree, call it u. Color vertex u "black". Then, for every colored vertex, if the vectex has some adjacent uncolored vertex, pick that vertex, say v (so v has atleast 1 colored adjacent vertex). If all of v's adjacent colored vertices are all "black", color v "white" and repeat. Otherwise, if all of v's adjacent colored vertices are all "white", color v "black" and repeat. (If neither condition holds, we terminate and fail). When there are no uncolored nodes left, we terminate.

At every iteration of the algorithm that does not fail, we add a color to some vertex such that its coloring is not shared by its adjacent vertices. If the program terminates successfully, every node will be colored such that there is no conflict with its neighbors, which means this algorithm has successfully 2-colored the tree.

Now suppose, for some tree T, that the algorithm fails at some step. That is, there is some uncolored node v such that it has atleast 1 "black" and 1 "white" peer, say b and w respectively. Notice, from our algorithm, that every colored node x has a path from the initial node u to x along colored nodes since we iteratively color nodes adjacent to already colored nodes and we initially started with colored node u. So there exists some simple paths A and B along colored vertices to b and w respectively. Importantly, neither of these paths can go through v since v is uncolored. However, consider the paths (A,(b,v)),(B,(w,v)). Both are distinct simple paths from u to v, which contradicts the property/definition of trees that every 2 nodes in the tree have exactly 1 simple path between them. Hence, our assumption was wrong and the algorithm could not have failed on the tree T. So the algorithm terminates on all trees T, and as per our argument above, constructs a 2 coloring of every tree; whence, every tree has chromatic number 2 (except for the tree on 1 vertex which has chromatic number 1)

Problem 3

Here, we can construct a graph on each committee where edges between committees indicate that there is atleast 1 member in both meetings. See attached for graph (note that we labelled F=Fisher,D=DeWitt,G=Goodman,V=
to indicate the members who are shared by both meetings; we will ignore these labels henceforth).

Notice that this graph is basically the complete graph on 6 nodes except for no edge between "Facilities" and "Graduate Advising". I claim that the chromatic number of this graph is 5. To show this, first I will show that 4 is not a possible chromatic number (i.e. there is no 4 coloring of this graph). Consider the induced subgraph on "Alumni Relations","Curriculum","Facilities","Graduate Admissions" (ignoring "Graduate Advising"). This is the complete graph on 5 vertices. Consider any coloring of these 5 vertices by 4 colors. Since there are 4 colors and 5 vertices picking from those 4 colors, by Pidgeonhole, it follows that atleast 2 of the vertices are the some color. In this induced subgraph, because it is complete, these 2 vertices must be connected; but this means that 2 adjacent edges in this coloring share the same color in the coloring. Because we choose any arbitrary coloring, it follows that any coloring of this subgraph has adjacent vertices with the same color. In the original graph including "Graduate Advising", this property holds since a 4 coloring of the whole graph of 6 vertices could then form a 4 coloring of the subgraph from above by removing "Graduate Advising" and its edges (after all, this removes edges and vertices without adding more dependencies), a contradiction. Overall, we have shown that 4 cannot be the chromatic number

To show that 5 is the chromatic number, it suffices so simply construct a 5-coloring on our graph, from which it follows that 5 is the smallest number that colors the graph where adjacent vertices are colored differently. We do this in our attached graph, where {A,B,C,D,E} are colors in our coloring. Notice that the only duplicate color is D on "Graduate Advising" and "Facilities". However, these 2 vertices don't share an edge, so no worries there. Ultimately, we have shown a 5 coloring on our graph in which adjacent nodes are colored differently.

To solve our original problem, we will notice that each vertice represents a committee and each edge represents a dependency (i.e. a shared member) which implies that the 2 meetings cannot co-occur. If {A,B,C,D,E} represent time slots, there we have shown a scheduling assignment in which every meeting has a time slot and no 2 meetings with a dependency (edge) have the same time slot (since no 2 adjacent vertices have the same color). Since there are no possible 4-colorings on the graph satisfying dependencies, it follows there are no way to use 4 timeslots to meet this criteria. Ultimately, this shows that 5 is the minimum number of timeslots required by the department.

Problem 4

Part a)

There are c distinct trees in the spanning forest.

In the original graph, the connected components form a partitioning of the graph into c parts. In fact, these partitions correspond to equivalence classes on the vertices under the relation "is connected to". Since all pairs of vertices within each connected component are connected to each other, they each form a tree in the spanning tree. And no 2 vertices between these partitions are connected in the spanning forest because they are not connected in G. Each partition in G corresponds to a unique tree in the spanning forest. Hence, there are c trees in the spanning forest.

Part b)

There are $n - c$ edges in the spanning forest.

Let some G be given with n vertices, m edges, and c connected components. Consider the spanning forest F. As per our argument in a), we know that there are c separate trees in F. Consider some arbitrary ennumeration on these c separate trees (i.e. assign a unique natural number between 1 and c to each). Let k_i be the number of vertices in the ith tree, for $i \in \{1, 2, ..., c\}$. We know that $\sum_{i=1}^{c} k_i = n$ since there are n total vertices. In every tree of F, we know that the number of edges in the tree is the number of vertices minus 1; so the ith tree must have $k_i - 1$ edges, for $i \in \{1, 2, ..., c\}$. Now consider the sum of the edges over all trees $\sum_{i=1}^{c}(k_i - 1)$, which is the number of edges in all of F.$\sum_{i=1}^{c}(k_i - 1) = \sum_{i=1}^{c}(k_i) + \sum_{i=1}^{c}(-1) = n - c$ since $\sum_{i=1}^{c} k_i = n$ and $\sum_{i=1}^{c} -1$ is merely $-1 * c$. So, overall, there are $n - c$ edges.

Problem 5

Part a)

$\epsilon|((1|01^*0)\{0,1\}^*)$

That is, the language of binary strings starting with 1 (and anything after), or starting with a 0, zero or more 1s, and 0 (and anything after), or empty.

Part b)

0^*11^*

That is, any binary string with zero or more 0s at the beginning and atleast one or more 1s exactly after (until the end).

Problem 6

See attached