

MATH 3134 EXAM 2 SOLUTIONS  
Mar 18, 2005

1. (15 pts)

(a) Let  $G$  be an undirected graph. Define what a coloring of  $G$  means.

A *coloring* of  $G$  is a function  $f : V_G \rightarrow S$  from the vertices of  $G$  to a set  $S$  such that if  $U$  and  $V$  are adjacent vertices  $f(U) \neq f(V)$ .

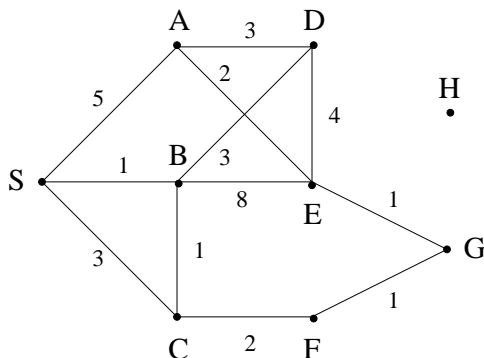
Note: There are many acceptable answers. You could talk about assigning colors to vertices, you could require that  $S$  be finite (we did in class), or you may not require that adjacent vertices have different colors. As with many other concepts in graph theory, the terminology varies from author to author. (If you look in another book on graph theory, it's always a good idea to check on the their definitions first.) Actually, a coloring is always required to assign different colors to adjacent vertices. But we gave a definition of colored graph in class which did not require this. I realize that the difference between *colored graph* and the *coloring* of a graph may have confused you, so I accepted answers that allowed adjacent vertices to have the same color.

(b) We say that a graph  $G$  is *color-critical* if it has the property that whenever a vertex  $V$  and the edges incident to it are removed from  $G$ , the resulting graph has smaller chromatic number than  $G$ . Let  $G$  be an undirected, color-critical graph of chromatic number  $k$ . Prove that the degree of each vertex of  $G$  is at least  $k - 1$ . (You must give the complete proof and may not refer to the identical result on the homework.)

Notice that adding a vertex to a graph and connecting to some other vertices can increase the chromatic number by at most 1, so removing a vertex can decrease it by at most 1. Let  $V$  be any vertex of  $G$  and let  $G'$  be the graph we get by removing  $V$  and its incident edges from  $G$ . By the above remark,  $G'$  has chromatic number  $k - 1$ . Consider a coloring of  $G'$  with  $k - 1$  colors. If we add  $V$  and its incident edges back to  $G'$ , the chromatic number has to increase, so we cannot use any of the  $k - 1$  colors to assign to  $V$ . This must be because  $V$  is adjacent to at least one vertex of each of the  $k - 1$  colors. So  $V$  must be adjacent to at least  $k - 1$  vertices, so  $\deg(V) \geq k - 1$ .

2. (25 pts)

(a) Consider the graph below. In order to find a shortest path from  $S$  to every other vertex, Dijkstra's algorithm is run. Suppose that at an intermediate stage of the algorithm, the set  $P = \{S, B, C, D\}$ .



Find the label and the predecessor of each vertex and fill in the table below.

For  $V \in P$ , the label must be the length of a shortest path from  $S$  to  $V$  and the predecessor is the vertex preceding  $V$  on such a path. So  $L(B) = 1$ ,  $Q(B) = S$ ,  $L(C) = 2$ ,  $Q(C) = B$ , and  $L(D) = 4$ ,  $Q(D) = B$ .  $L(S) = 0$  but the algorithm assigns no predecessor to  $S$ .

For  $V \notin P$ , the label must be the length of a shortest path from  $S$  to  $V$  all of whose vertices but  $V$  are in  $P$  and the predecessor is the vertex preceding  $V$  on such a path. So  $L(A) = 5$ ,  $Q(A) = S$ ,  $L(E) = 8$ ,  $Q(E) = D$ ,  $L(F) = 4$ ,  $Q(F) = C$ . There is no such path to  $G$  and  $H$ , so  $L(G) = L(H) = \infty$  and  $Q(G)$ ,  $Q(H)$  are whatever the algorithm assigned to them during initialization. This could be  $S$  or nothing at all, both answers are acceptable.

vertex	$S$	$A$	$B$	$C$	$D$	$E$	$F$	$G$	$H$
label	0	5	1	2	4	8	4	$\infty$	$\infty$
predecessor	—	$S$	$S$	$B$	$D$	$D$	$C$	—	—

Note: The textbook's version of Dijkstra's algorithm initializes the predecessor of every vertex to  $S$ , so  $Q(G) = Q(H) = S$  is another correct answer. But this is really not necessary because if there is any path from  $S$  to  $V$ ,  $Q(V)$  will be updated during the algorithm, and if not, we will know this because  $L(V)$  remains  $\infty$  when the algorithm completes, and we will not try to use  $Q(V)$  to reconstruct a shortest path from  $S$  to  $V$ .

- (b) Which vertex will the algorithm add to  $P$  next?

$F$  because it has minimal label among the ones not in  $P$ .

- (c) The following description of Dijkstra's algorithm comes from a student's homework. There are at least 3 mistakes in it. Find them.

"Given a finite, undirected, weighted graph  $G$  (of positive weights), and a vertex  $S$ , Dijkstra's algorithm finds the shortest path<sup>1</sup> from  $S$  to every other vertex. It does this by keeping track of the  $k$  closest vertices to  $S$  in a set  $P$ , and assigning (temporary) labels to the vertices. For a vertex  $V \in P$ , its label  $L(V)$  is the (shortest) distance of  $V$  from  $S$ . For  $V \notin P$ ,  $L(V)$  is the shortest path<sup>2</sup> from  $S$  to  $V$  subject to the restriction that all of the vertices on this path except  $V$  must be in  $P$ . At every step, the algorithm picks a vertex  $U$  with the smallest label outside  $P$  and adds it to  $P$ . Then for every vertex  $V \notin P$ , it finds the vertex  $W \in P$  for which  $\underline{w(W, V)}$ <sup>3</sup> is minimal and sets  $L(V) = L(W) + w(W, V)$ . The algorithm ends when there are no more vertices left outside  $P$ ."

1. In general, there is no "the" shortest path. There may be several path that all have minimal length. So it should say "a shortest path."

2.  $L(V)$  is a number, not a path. It is the length of a shortest path from  $S$  to  $V$  subject to the restriction that all of the vertices on this path except  $V$  must be in  $P$ .

3. The algorithm finds  $W \in P$  so that  $L(W) + w(W, V)$  is minimal, otherwise there could be another vertex in  $X \in P$  through which there is a shorter path to  $U$ .

Note: The paragraph is not a complete recipe about Dijkstra's algorithm. It does not talk about how to initialize  $P$  and the labels, how to keep track of predecessors, and how to list shortest paths in the end. Before complaining about all these missing parts, you should find the three actual mistakes. Also, while Dijkstra's algorithm could end

when all the labels on the vertices outside  $P$  are  $\infty$ , there is no harm in continuing until all vertices have been added to  $P$ . In fact, this is how the algorithm is given in the textbook.

3. (10 pts)

(a) Let  $G$  be a directed graph. Define what it means for  $G$  to be weakly connected.

$G$  is *weakly connected* if ignoring the orientation of the edges, the resulting undirected graph is connected.

(b) Let  $G$  be a finite, directed (multi)graph. Give a necessary and sufficient condition for the existence of an Eulerian path in  $G$ .

$G$  has an Eulerian path if and only if  $G$  is weakly connected up to isolated vertices, it has vertices  $U$  and  $V$  such that

$$\text{outdegree}(U) = \text{indegree}(U) + 1,$$

$$\text{indegree}(V) = \text{outdegree}(V) + 1,$$

and  $\text{indegree}(W) = \text{outdegree}(W)$  for all other vertices  $W \neq U, V$ .

4. (10 pts) **Extra credit problem.**

(a) Prove that a finite, undirected, color-critical (see problem 1) graph  $G$  is always connected. (Hint: If  $G$  is not connected, what does its chromatic number have to do with the chromatic numbers of its connected components?)

Let  $\chi(G)$  denote the chromatic number of  $G$ . Let  $G_1, G_2, \dots, G_n$  be the connected components of  $G$ . Notice that  $\chi(G) = \max(\chi(G_1), \chi(G_2), \dots, \chi(G_n))$ . If  $k = \max(\chi(G_1), \chi(G_2), \dots, \chi(G_n))$ , then every  $G_i$  can be colored with at most  $k$  colors, so  $G$  can be colored with at most  $k$  colors. But any coloring of  $G$  is also a coloring of any of its connected components, so if  $G$  could be colored with less than  $k$  colors, so could all of the  $G_i$ .

Now let  $G_j$  be a connected component whose chromatic number is maximal. If  $G$  were not connected, we could choose another connected component different from  $G_j$  and remove a vertex from that. This would not change  $\chi(G_j)$ , and therefore  $\chi(G)$ , contradicting the fact that  $G$  is color-critical.

(b) Let  $G$  be a finite, undirected graph  $G$  whose chromatic number is  $k$ . Prove that  $G$  has a color-critical subgraph  $H$  whose chromatic number is also  $k$ .

If  $G$  has a vertex which can be removed without decreasing the chromatic number, remove it. Keep doing this as long as you can. Since  $G$  is finite, this process eventually ends and results in a subgraph  $H$  from which no more vertices can be removed without decreasing its chromatic number. As we removed vertices from  $G$  one-by-one to get  $H$ , the chromatic number always stayed the same, so  $\chi(G) = \chi(H)$ .

Note: You need to remove the vertices one by one. Knowing that removing either  $U$  or  $V$  from  $G$  does not decrease  $\chi(G)$  does not mean that removing both  $U$  and  $V$  will not decrease it either. (Can you find an example?) So after removing such a vertex, you need to re-evaluate which remaining vertices could be removed without changing  $\chi(G)$ . This is why we need  $G$  to be finite for this proof.