Homework 8 Solutions

Amit C, Suman B, Sagar K, Zhao T
Computer Science Department
Dartmouth College

By far, the most common mistake in this homework was using induction incorrectly for graph problems. When proving something about graphs by induction, you want to reduce from n to $n-1$. This is because you are proving a statement for all graphs of size n, and not just a specific graph. By proving something by constructing an $n+1$ size graph using an n size graph, you have only shown it for "one particular" graph of size $n+1$, whereas you wanted to show it for all graphs of size $n+1$. (By size, I mean either the number of vertices or the number of edges, whatever you are inducting on.)

1. Prove the final step in the Tree Theorem: that (6) $\Longrightarrow$ (1). Refer to the lecture notes on the website.

Solution: We prove by induction on n that if $|V| = n$ and G is acyclic and $|E| = n-1$, then G is connected.

Base case ($n = 1$): This is trivial; G consists of a single vertex and no edges.

Induction step ($n \geq 2$): First we will prove that G has a leaf. Since $E \neq \emptyset$, we can pick a vertex $v \in V$ with $\deg(v) > 0$. Start walking from v until you reach a leaf, say u. This will happen because the graph is acyclic.

Now, $G - u$ has $n-1$ vertices and $n-2$ edges, so we can use the induction hypothesis to conclude that it is connected. Adding u to G with the original edge keeps it connected.

Common Mistake: Constructing $n+1$ vertex graph using an n vertex graph in induction step. If you did this, you missed showing existence of a leaf, which is crucial.

2. Solve Problem 9.18 parts (a), (b), and (c) from the [LLM] book.

Solution: 9.18 Part (a): Consider the following tournament digraph.

$$G = (\{1,2,3\}, \{(1,2),(2,3),(3,1)\}).$$

The paths $(1,2,3)$ and $(2,3,1)$ are both rankings.

9.18 Part (b): Let G is a tournament digraph with n vertices and say G is also a DAG. Let there be two rankings in G, say $R = (r_1, r_2, \ldots, r_n)$ and $S = (s_1, s_2, \ldots, s_n)$. Edges of S cannot all be of the form (r_i, r_j) where $i < j$, because R and S are different. Hence there is an edge in S of the form (r_i, r_j) where $i > j$, and this edge gives us a cycle when added to the path $(r_j, r_{j+1}, \ldots, r_i)$, which is a contradiction.

9.18 Part (c): We prove this by induction on number of vertices, i.e., we use the predicate $p(n)$ that if G is a tournament digraph with n vertices, then G has a ranking. Base case is for $n = 1$; G is just one vertex, and that vertex itself forms the ranking. For induction step, let $n \geq 2$. Remove any vertex v from G. By induction hypothesis, $G - v$ has a ranking, say $R = (r_1, r_2, \ldots, r_{n-1})$. If (r_{n-1}, v) is an edge in G, then we can extend R to get a ranking in G. If not, consider the minimum index i such that (v, r_i) is an edge. Then $(r_1, r_2, \ldots, r_{i-1}, v, r_i, \ldots, r_{n-1})$ is a ranking in G.

Common Mistake: Again, most of you added a vertex to a smaller graph in Part (c) of the problem. But in this case it did not cost you any marks because the proof works for all bigger graphs.

3. Solve Problem 11.11 from the [LLM] book.

Solution: Use Theorem 11.5.8 from the [LLM] book that says every regular bipartite graph has a perfect matching. Take G and remove a perfect matching whose existence is guaranteed by Theorem 11.5.8. You get a regular bipartite graph with each vertex having degree $j-1$. Repeat this process until you have j perfect matchings (i.e., j blocks).

Common Mistake: Same mistake one more time; adding a perfect matching to a $j-1$ regular graph and arguing that the resulting graph is j regular is not enough. By doing this, you just constructed "one" j-regular graph that can be divided into blocks. You need to argue that Hall's marriage theorem's conditions are met (or use Theorem 11.5.8 from [LLM] book), and there exists a perfect matching. Then you remove this perfect matching and apply the induction hypothesis.

4. Solve Problem 11.15 from the [LLM] book.

Solution: 11.15 Part (a): The obvious construction works. Add an edge between a column and a value if that column has that value. Call this graph $G = (L, R, E)$.

The graph is not necessarily degree constrained, because a column can have same four values; say it has four aces, and, say, four queens appear in different columns.

11.15 Part (b): Consider a set $S \subseteq L$ of any n columns. Then there are total $4n$ cards, call this set of cards C. If number of distinct values in C is less than n, then some value has to appear more than 4 times, which is not possible. In other words, $|S| \leq |N(S)|$. Since conditions in Hall's theorem are satisfied, there is a perfect matching in G.

5. Solve Problem 11.42 from the [LLM] book.

Solution: 11.42 Part (a): As the Euler tour covers all edges, we can count the degree of each vertex in the graph along the Euler tour. When the tour leaves a vertex, we add 1 to the degree of that vertex; when the tour enters a vertex, we also add 1 to the degree of that vertex. For each appearance of a vertex along the tour except the start and end of the tour, it is entered and left, so the degree of every vertex in the middle always increases by 2 each time in the counting; for the start and end vertex (closed walk), its degree increases by 1 in the beginning and by 1 at the end (2 in total). So the degree of each vertex is even.

11.42 Part (b): There are two cases. If the Euler walk covers all the vertices in the graph, because it does not include every edge, there exist an edge that is not on the walk and it connects two vertices on the Euler walk. So, we find an edge that is incident to a vertex on the walk. If the Euler walk does not cover all the vertices, There must be an edge from the vertices on the walk to the rest vertices; otherwise, the graph will not be connected. So that edge is incident to a vertex on the walk.

Common Mistakes: Some of you just assume the Euler walk does not cover all vertices without considering the other possible case where the Euler walk covers all vertices.

Some of you say that if there is an edge unincluded in the Euler walk and not incident to any vertex on the walk, the that edge must be in another separated connected component. That is an invalid argument. In order for that reasoning to be valid, you need that *none* of unincluded edges is incident to any vertex on the walk.

11.42 Part (c): Suppose **w** is not an Euler tour. Then it does not include every edge in the graph. From Part (b), we know that there must be an unincluded edge that is incident to a vertex on **w**; let's say it is v and it connects to u. Then we can construct another Euler walk that is longer than **w**:

$$\underbrace{v, \ldots, v}_{\mathbf{w}}, (v, u), u \ .$$

It contradicts that **w** is the longest Euler walk. So **w** must be an Euler walk.

11.42 Part (d) With the similar reasoning to Part (c), if there is an edge incident to the end of **w** and it is not included in **w**, then we can extend **w** to a longer Euler walk, which contradicts that **w** is the longest Euler walk. So all the edges incident to the end of **w** must already be in **w**.

11.42 Part (e) Let v denote the end of **w**. First, from Part (d) we know that all edges incident to v are already in **w**. Then we can employ the same method as Part (a) to count the degree of v. For each appearance of v in the middle of $\mathbf{w}$, its degree increases by 2. But for the occurrence of v at the end of $\mathbf{w}$, its degree only increases by 1. So the degree of v is odd.

11.42 Part (f) From Part (d) and Part (e) we know that the longest Euler walk must be a closed walk in a connected graph in which every vertex has even degree. And then from Part (c), we can conclude that the longest Euler walk is an Euler tour in a connected graph in which every vertex has even degree.

6. Recall the theorem that we stated in class: for every directed acyclic graph (DAG) G, if we define

$$\text{width}(G) = \max\{|A| : A \text{ is an antichain of } G\},$$
$$\text{pcov}(G) = \min\{|\mathscr{P}| : \mathscr{P} \text{ is a path cover of } G\},$$

then $\text{width}(G) = \text{pcov}(G)$. In class we proved the "easy" direction of this result, that $\text{width}(G) \leq \text{pcov}(G)$. This problem and the next one will walk you through the proof of the "harder" direction: $\text{width}(G) \geq \text{pcov}(G)$.

6.1. Start a proof by induction on the number of vertices of G. Write out a clear statement $p(n)$ that you will prove by induction on n. Prove the base case.

Solution: Let $p(n)$ be the statement "If G is a DAG with n vertices, then $\text{width}(G) \geq \text{pcov}(G)$."

We will show that $p(n)$ holds for all $n \geq 1$ by strong induction on n.

The base case is $n = 1$. Then G must be a single vertex and we trivially have $\text{width}(G) = \text{pcov}(G) = 1$.

Common Mistakes: Some of you used $n = 2$ as base case. There are two cases for $n = 2$: there is an edge or there is no edge. DAG does not require connectivity.

6.2. For the induction step, consider a DAG $G = (V, E)$ with $n \geq 2$ vertices. Let $k = \text{width}(G)$. Let π be a maximal path in G, from u to v. (Maximal means that no longer path fully contains π.) Let G' be obtained by deleting u and v from G. Finish the proof in the case that $\text{width}(G') \leq k - 1$.

Solution: Since G' has fewer than n vertices, by the induction hypothesis, $\text{pcov}(G') \leq \text{width}(G') \leq k - 1$. Therefore G' has a path cover $\mathscr{P}$ consisting of at most $k - 1$ paths. The only vertices of G not covered by $\mathscr{P}$ are u and v; these can be covered by adding π to $\mathscr{P}$. Therefore $\text{pcov}(G) \leq (k - 1) + 1 = k = \text{width}(G)$.

6.3. Now consider the other case: $\text{width}(G') > k - 1$. Let A be a maximum antichain in G'. Explain why $|A| = k$.

Solution: We claim that A is also an antichain in G. This will imply $|A| \leq \text{width}(G) = k$. We already have $|A| = \text{width}(G') > k - 1$, so we must have $|A| = k$.

To prove the claim, suppose A is not an antichain in G. Then there exists a path π' in G from $w \in A$ to $x \in A$ where $w \neq x$. Since this path did not exist in G' (because A is an antichain in G'), it must go through either u or v ((v, u) is not an edge in G because G is a DAG). This is impossible because u has in-degree 0 and v has out-degree 0, due to maximality of π.

6.4. Continuing the second case, define the sets

$$\text{from}(A) = \{x \in V : a \rightsquigarrow x \text{ for some } a \in A\},$$
$$\text{to}(A) = \{x \in V : x \rightsquigarrow a \text{ for some } a \in A\}.$$

Prove that $\text{from}(A) \cap \text{to}(A) = A$ and that $\text{from}(A) \cup \text{to}(A) = V$.

Solution: Consider an arbitrary $x \in V$.

If $x \in \text{from}(A) \cap \text{to}(A)$, then $x \in \text{from}(A)$ so there exists $a \in A$ with $a \rightsquigarrow x$, and $x \in \text{to}(A)$ so there exists $a' \in A$ with $x \rightsquigarrow a'$. Therefore $a \rightsquigarrow x \rightsquigarrow a'$. Since A is an antichain, $a = a'$. Since G is a DAG,

$x = a$. Therefore $x \in A$. We conclude $\text{from}(A) \cap \text{to}(A) \subseteq A$. The other direction, $A \subseteq \text{from}(A) \cap \text{to}(A)$ is immediate. This proves the first equation.

For the second equation, suppose $x \notin \text{from}(A)$ and $x \notin \text{to}(A)$. Then $x \notin A$, and $A \cup \{x\}$ is an antichain because there will be no path between x and any vertex in A. This contradicts the maximality of A. Therefore we must have $x \in \text{from}(A) \cup \text{to}(A)$ for *every* x.

7. This is a continuation of the previous problem.

7.1. Consider the subgraph G_{from} of G (notice: I said G, not G') induced by from(A): i.e., the digraph obtained by starting with G and deleting all vertices not in from(A). Explain why the induction hypothesis can be applied to G_{from}.

Solution: We have $u \notin A$ because A consists of vertices of G'. Therefore $u \notin \text{from}(A)$, because u has in-degree 0. Therefore G_{from} does not contain u, and so it has fewer than n vertices. Thus, the induction hypothesis applies to it.

Common Mistakes: You are supposed G_{from} has fewer vertices than G. It is not sufficient to only show G_{from} is a DAG.

7.2. Apply the induction hypothesis to G_{from} and conclude the existence of a suitable path cover.

Solution: We concluded in #6.3 that $|A| = k$. By construction, A is a maximum antichain in G_{from}. Thus, by the induction hypothesis, G_{from} has a path cover $\{\pi_1, \ldots, \pi_k\}$, using k paths.

7.3. Define an analogous digraph G_{to} and conclude the existence of another path cover.

Solution: Let G_{to} be the subgraph of G induced by to(A). Similar to the above argument, we conclude that G_{to} does not contain v, so the induction hypothesis applies to it and it has a path cover $\{\pi'_1, \ldots, \pi'_k\}$, using k paths.

7.4. Finish the proof by suitably "stitching together" these path covers.

Solution: Each path π_i must contain exactly one element of A, because A is an antichain: call this element a_i. Path π_i must start at a_i: if it started at some $x \neq a_i$, then because $x \in \text{from}(A)$, this would contradict either the antichain property of A or the DAG property of G.

Similarly, each path π'_i must contain exactly one vertex in A, and must end at that vertex. Renumber the paths so that π'_i ends at the same a_i that π_i starts at.

Now we can stitch each π'_i with π_i, obtaining a new collection of k paths. These paths cover G, because $\text{from}(A) \cup \text{to}(A) = V$. We conclude that $\text{pcov}(G) \leq k = \text{width}(G)$, and the proof is complete.