Massachusetts Institute of Technology
6.042J/18.062J: Mathematics for Computer Science
Professor Albert R. Meyer

Handout 37
April 1, 1998

Quiz Solution

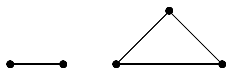
Problem 1. [25 pts]

Let L_n be the $n+1$-vertex simple "line" graph consisting of a single simple path of length n. For example, L_4 is shown in the figure below:

The line graph L_4

Let's say a simple graph has "two ends" if it has exactly two vertices of degree one, and all its other vertices have degree two. In particular, for $n \geq 1$, the graph L_n has two ends.

a) [3 pts] Draw a diagram of the smallest simple graph with two ends which is not isomorphic to any line graph. *Hint*: See part (d).
Solution:



The smallest two-ended graph not isomorphic to any line graph.

b) [8 pts] Explain briefly, but clearly, what is wrong with the following argument:

False theorem: Every simple graph with two ends is isomorphic to L_n for some $n \geq 1$.

False proof: We prove by induction on the number, $n \geq 1$, of edges in a simple graph, that every two-ended graph with n edges is isomorphic to L_n.
(Base case $n = 1$): A simple graph with one edge can only consist of the two vertices connected by that edge and some number of vertices not attached to any edge, i.e., vertices of degree zero. Since a two-ended graph cannot have vertices of degree zero, the only two-ended graph with one edge must consist solely of the two vertices connected by the edge, which makes it isomorphic to L_1.
(Induction case): Assume that $n \geq 1$ and every two-ended graph with n edges is isomorphic to L_n. Now let G_n be any two-ended graph with $n \geq 1$ edges. By hypothesis, G_n is isomorphic to L_n. Suppose an edge is added to G_n to form a two-ended graph G_{n+1}.

Since G_n is isomorphic to L_n, it consists of a simple path of length n. The only way to add an edge to the path and preserve two-endedness is to have that edge go from one end—that is, one of the degree-one vertices—to a new vertex, lengthening the path by one. That is, the resulting $n+1$-edge graph must be a simple path of length $n+1$, so it is isomorphic to L_{n+1}. Q.E.D.

Solution:

The inductive case is argued in an incorrect way. The inductive case must consist of a proof for any $n \geq 1$ that if G_{n+1} is two-ended then $G_{n+1} \sim L_n$ ($A \sim B$ is a notation for "A is isomorphic to B"), using the assumption that for every two-ended G_n, $G_n \sim L_n$ (alternatively, in strong induction, we can use an assumption that for every k, $1 \geq k \geq n$, if G_k two-ended then $G_k \sim L_k$).

The false proof above does not argue that at all. Instead, it states that if one takes a two-ended G_n which is isomorphic to L_n, and extends it by an edge to form a two-ended $(n+1)$-edge graph G', then $G' \sim L_{n+1}$. This is true, but to transform this argument into an inductive proof, one would have to argue that every two-ended $(n+1)$-edge graph G_{n+1} can be formed in this way, i.e. by adding another edge to some two-ended graph G_n. However, this cannot be argued because it is not true. For example, in the solution to part (a), we have a two-ended graph G_4 which cannot be formed by adding an edge to some two-ended G_3.

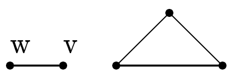
c) [7 pts] Here is another argument for the same False Theorem as in part (b). Explain exactly where it goes wrong.

False proof 2: Same induction hypothesis and base case as in part (b).

(Induction case): For any $n \geq 1$, let G_{n+1} be any two-ended graph with $n+1$ edges. Let G_n be the graph which results from removing one of the degree-one vertices v of G_{n+1} and the edge $\{v, w\}$ attached to it. So G_n no longer has the vertex, v, of degree one. But the degree of w is one less in G_n than it was in G_{n+1}, so G_n still has two vertices of degree one and one fewer vertex of degree two. Therefore G_n is also two-ended. By induction G_n consists of a simple path of length n. But G_{n+1} is obtained by attaching an edge from one end of the path to a vertex v not on the path, thereby lengthening the path by one. So G_{n+1} is isomorphic to a simple path of length $n+1$; that is, it is isomorphic to L_{n+1}. Q.E.D.

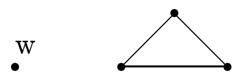
Solution:

The underlined statement is wrong. The degree of w is one less in G_n than it was in G_{n+1}, hence it can be equal to one if it was two in G_{n+1}, but it can also be equal to zero if it was one in G_{n+1}. The false proof above is wrong because it ignores this this second possibility. For example, if we take $G_{n+1} = G_4$ from the solution to part (a):



Graph G_{n+1}

We see that if you remove v then w will have degree zero in a resulting graph G_n:



Graph G_n

Obviously, G_n is not a line graph.

d) [7 pts] Describe how to make a small revision to one of the false proofs above so that it becomes a correct proof of the theorem "Every *connected* simple graph with two ends is isomorphic to L_n for some $n \geq 1$."

Solution:

The proof in part (c) can be easily fixed if the graphs are connected, because then the underlined statement will be true. Here is an example of how one could inject into the proof from part (c) an argument justifying the underlined statement for the case of connected graphs:

> [...But the degree of w is one less in G_n than it was in G_{n+1}.] Since G_{n+1} was two-ended, w must be of degree either one or two in G_{n+1}. However, it cannot be of degree one because then vertices v and w would be connected only to each other and disconnected from the rest of the graph, which would contradict the assumption that G_{n+1} is connected. Therefore, w must have degree two in G_{n+1}, and hence has degree one in G_n. Hence, [G_n still has two vertices of degree one and one fewer vertex of degree two. Therefore...]

The false proof in part (b) cannot be easily fixed. If we tried to use its structure to prove that all connected two-ended graphs are linear graphs, we would have to supply the argument that every connected two-ended graph can be formed by adding an edge to some connected two-ended graph with one less edge. To argue that would be as complicated as the above proof.

Problem 2. Consider the following simple graphs:

$$K_8 \quad C_8 \quad K_{4,4} \quad W_7 \quad Q_3 \quad L_7$$

(a) What is the minimum number of colors (chromatic number) needed to vertex-color each of these graphs so that adjacent vertices have different colors?

Solution:

$$K_8 = 8 \quad C_8 = 2 \quad K_{4,4} = 2 \quad W_7 = 4 \quad Q_3 = 2 \quad L_7 = 2$$

(b) Let v_1, v_2, ... be variables ranging over vertices and $E(v_1, v_2)$ mean that v_1 and v_2 are connected by an (undirected) edge. For each of the following logical statements, indicate exactly which of the six graphs above satisfies the statement.

Note: On these problems, it was a mistake if a student failed to include a graph that satisfied the statement or if a student included a graph that didn't satisfy the statement. With the exception of the second statement, if a student had 1 to 2 mistakes that student lost 1 point, and if a student had more than 2 mistakes, that student lost 2 points. On the second statement, students either received 0 or 2 points.

a) $\exists v_1 v_2 : v_1 \neq v_2 \wedge \neg E(v_1, v_2)$
Solution: This one specifies the graphs with two distinct vertices without an edge between them. Obviously this is not satisfied by K_8. All of the other graphs satisfy it. In fact, the only simple graphs with two or more vertices which don't satisfy this are the complete graphs K_n for $n \geq 2$. So the answer is:

$$C_8, K_{4,4}, W_7, Q_3, L_7.$$

b) $\forall v_1 v_2 : E(v_1, v_2)$
Solution: This one specifies all graphs where every vertex is attached to *every* vertex — not just every *OTHER* vertex. To satisfy this statement, there must be self-loops connecting each vertex to itself. Simple graphs do not have self-loops, so in fact the *only* simple graph which satisfies this statement is the empty graph (with no vertices).
K_8 was a popular mistaken answer to this question. It would have been correct had the statement been: $\forall v_1 v_2 : v_1 \neq v_2 \implies E(v_1, v_2)$.

c) $\forall v_1 v_2 : v_1 \neq v_2 \implies [E(v_1, v_2) \vee \exists v_3 : E(v_1, v_3) \wedge E(v_3, v_2)]$
Solution: This statement says that every pair of distinct vertices are connected by a path of length one or two. The graphs with this property are:

$$K_8, K_{4,4}, W_7.$$

d) $\exists v_1 v_2 v_3 v_4 : v_2 \neq v_3 \wedge v_2 \neq v_4 \wedge v_3 \neq v_4 \wedge E(v_1, v_2) \wedge E(v_1, v_3) \wedge E(v_1, v_4)$
Solution: The statement says that there are three distinct vertices v_2, v_3, v_4 such that there is some vertex v_1 (possibly one of them) that has an edge to each of them. In other words, there is a vertex of degree at least three. The graphs with such a vertex are:

$$K_8, K_{4,4}, W_7, Q_3.$$

(c) Write a logical statement like those in part (b) (involving only variables ranging over vertices, the expressions $E(v_i, v_j)$, $v_i = v_j$, $v_i \neq v_j$, logical connectives and quantifiers) which is satisfied precisely by those simple graphs which have a vertex of degree two.

Solution: The following is one possible solution:

Let $F(v_1, v_2, v_3)$ be the statement: $(v_1 \neq v_2) \wedge (v_1 \neq v_3) \wedge (v_2 \neq v_3)$. Thus $F(v_1, v_2, v_3)$ is satisfied iff v_1, v_2, and v_3 are distinct.

Let $G(v_1, v_2, v_3)$ be the statement: $E(v_1, v_2) \wedge E(v_1, v_3)$. Thus $G(v_1, v_2, v_3)$ is satisfied iff v_1 has edges to both v_2 and v_3.

Let $H(v_1, v_2, v_3)$ be the statement: $F(v_1, v_2, v_3) \wedge G(v_1, v_2, v_3)$. Thus $H(v_1, v_2, v_3)$ is satisfied iff v_1 has degree at least 2 because it has edges to distinct vertices v_2 and v_3.

$\exists v_1 v_2 v_3 \forall v_4 : H(v_1, v_2, v_3) \wedge (E(v_1, v_4) \rightarrow ((v_2 = v_4) \vee (v_3 = v_4))$

The intention in this problem was that a *simple* graph should satisfy the statement iff the graph had a vertex of degree two. The implicit restriction to simple graphs was misunderstood by some students, who tried to write a statement which would not be satisfied by nonsimple graphs — such as a multigraph with two edges between the same vertices. It isn't possible to exclude such multigraphs with the allowed statements, since these statements provide no way to differentiate edges with the same end points. It would have been better if the problem had emphasized that only simple graphs were under consideration.

The most common mistake here was to specify that the graph had to have a vertex of degree ≥ 2 but fail to specify that the same vertex had degree ≤ 2. This mistake cost 4 points.

Problem 3. Q_n is the simple graph with 2^n vertices known as the n-dimensional hypercube. It can be defined recursively by the conditions that Q_0 consists of a single vertex and Q_{n+1} is obtained by taking two copies of Q_n and adding edges between each vertex in one copy and the corresponding vertex in the other copy.

(a) How many edges are there in Q_3?

Solution: 12. There are 4 edges in Q_2 and and 4 edges are required to connect the two Q_2 graphs, and $2 * 4 + 4 = 12$.

(b) Describe a simple, closed-form formula for the number of edges in Q_n.

Solution: Let $E(n)$ be the number of edges in Q_n. Since each vertex in Q_n has degree n and there are 2^n vertices, the total number of edges is $E(n) = n2^n/2 = n2^{n-1}$. Although the recursive formula $E(n) = 2E(n-1) + 2^{n-1}$ with $E(0) = 0$ (the base case is important for this recursive formula) is a correct recursive formula, it is not a closed form solution.

(c) Prove below the correctness of your answer to part (b).

Solution: The problem gives the following recursive formula for the number of edges in Q_n. $E(n) = 2E(n-1) + 2^{n-1}$ and $E(0) = 0$. We need to prove that the closed form solution to the problem is equivalent to the recursive formula.

Proof: by induction on n.
Inductive Hypothesis: $P(n) : E(n) = n2^{n-1}$.
Base Case: $P(0) : E(0) = 0 = 0 * 2^{-1}$.

Inductive Step: Assume $P(n)$ to prove $P(n+1)$.

$$\begin{aligned} E(n+1) &= 2E(n) + 2^n \\ &= 2 * n * 2^{n-1} + 2^n \\ &= n * 2^n + 2^n \\ &= (n+1)2^n \\ &= (n+1)2^{(n+1)-1} \end{aligned}$$

Thus $P(n+1) : E(n+1) = (n+1)2^{(n+1)-1}$ is proved, and so $P(n)$ is proved for all $n \geq 0$.

Problem 4. Start with 100 coins on a table, 98 showing "heads" and 2 showing "tails." There are two ways to change the coins:

(i) flip over any ten coins, or

(ii) let n be the number of coins on the table showing heads. Place $n+1$ additional coins on the table, with all the new coins initially showing tails.

For example, you might begin by flipping eight heads and two tails, yielding 92 heads and 8 tails, then add 93 tails, yielding 92 heads and 101 tails.

(a) Model this situation as a state machine, carefully defining the set of states, the start state and the possible state transitions.

Solution:

This can be modeled by a state machine $M = \{Q, Q_0, \delta\}$, where:

- $Q = \mathbb{N} \times \mathbb{N}$, the first integer is the number of heads and the second integer is the number of tails;
- $Q_0 = \{(98, 2)\}$
- δ is given by the rule that the next state of (h, t) can be any state of the form:

  $(h - a + (10 - a), t + a - (10 - a))$ where $0 \leq a \leq 10,\ a \leq h,\ 10 - a \leq t$, or $(h, t + h + 1)$.

Comment: Most students forgot to specify the range of a precisely for the first transition.

(b) Explain how to reach a state with exactly one tail showing.

Solution:

a) Follow the example given in the question to get to the state $(92, 101)$.

b) Repeat for 10 times: Do operation 1 to flip 10 tails into heads. This will result in the state $(192, 1)$, which is the desired state.

(c) Define the following derived variables of a state q:

a) $C(q) =$ the number of coins on the table

b) $H(q) =$ the number of heads

c) $T(q) =$ the number of tails

d) $C_2(q) = C(q) \bmod 2$

e) $H_2(q) = H(q) \bmod 2$

f) $T_2(q) = T(q) \bmod 2$

Which of these variables is

i) increasing <u>NONE</u>

ii) nondecreasing <u>$C(q)$, $H_2(q)$</u>

iii) decreasing <u>NONE</u>

iv) constant <u>$H_2(q)$</u>

Comment: A lot of students failed to notice that a constant variable is also a nondecreasing variable, and did not put $H_2(q)$ in part fii.

(d) Prove that it is not possible to reach a state in which there is exactly one head showing.

Solution:

An invariant of the state machine is that the number of heads is even. We can prove this by induction on the number of operations.

Let $P(n)$ ::= "After n operations, the number of heads is even."

Base case: At the start state, the number of heads is 98, which is even. Therefore $P(0)$ is true.

Inductive Step: Assume $P(n)$ is true. Let (h, t) be the state after n operations. By inductive hypothesis, h is even. For the $(n+1)^{st}$ step, we have two cases to consider:

a) The first operation is executed: $(h, t) \to (h - 2a + 10, t + 2a - 10)$. Since h is even and a is an integer, $h - 2a + 10$ is also even. Therefore the number of heads is even after the $(n+1)^{st}$ step in this case.

b) The second operation is executed: $(h, t) \to (h, t + h + 1)$. The number of heads does not change in this case. Therefore it remains even after the $(n+1)^{st}$ step.

Therefore, $P(n+1)$ is also true. By induction, we have proved that the number of heads is always even. Since one is odd, it is not possible to reach a state in which there is exactly one head showing.

Problem 5. Solution:

Method 1.

$$\begin{aligned}
\sum_{i=0}^{n}\sum_{j=0}^{i} x^j &= \sum_{i=0}^{n} \frac{1-x^{i+1}}{1-x} \\
&= \frac{1}{1-x}\sum_{i=0}^{n}(1-x^{i+1}) \\
&= \frac{1}{1-x}\left[\sum_{i=0}^{n} 1 - \sum_{i=0}^{n} x^{i+1}\right] \\
&= \frac{1}{1-x}\left[n+1-x\sum_{i=0}^{n} x^i\right] \\
&= \frac{1}{1-x}\left[n+1-x\frac{1-x^{n+1}}{1-x}\right] \\
&= \frac{(n+1)(1-x)-x(1-x^{n+1})}{(1-x)^2} \\
&= \frac{(n+1)-(n+1)x-x+x^{n+2}}{(1-x)^2} \\
&= \frac{(n+1)-(n+2)x+x^{n+2}}{(1-x)^2}
\end{aligned}$$

Method 2. A few people noticed that

$$\sum_{i=0}^{n}\sum_{j=0}^{i} x^j = (n+1) + nx + (n-1)x^2 + ... + x^n$$

and used the perturbation method. The algebra is messy, but this works also.

Common errors.

a) Writing

$$n = \sum_{i=0}^{n} 1$$

rather than

$$n+1 = \sum_{i=0}^{n} 1$$

We took two points off for this error.

b) Writing

$$\sum_{j=0}^{i} x^j = \frac{1-x^i}{1-x}$$

rather than

$$\frac{1-x^{i+1}}{1-x}$$

We took three points off for this error.

c) Some people used the geometric series formula on $\sum_{i=0}^{n} x^{i+1}$ without noticing that the additional factor of x had to be factored out first.

Problem 6. **(a)** Prove that $\log(n!) = \Theta(n \log n)$.

Solution:

There are several ways to prove this.

(i) Using Stirling's Formula,

$$n! = \sqrt{2\pi n}(\frac{n}{e})^n(1+\Theta(\frac{1}{n}))$$

Therefore,

$$\begin{aligned}
\log(n!) &= \log(\sqrt{2\pi n}(\frac{n}{e})^n(1+\Theta(\frac{1}{n}))) \\
&= \log(\sqrt{2\pi n}) + \log((\frac{n}{e})^n) + \log(1+\Theta(\frac{1}{n})) \\
&= \log(\sqrt{2\pi n}) + n\log(\frac{n}{e}) + \log(1+\Theta(\frac{1}{n})) \\
&= \log(\sqrt{2\pi n}) + n\log(n) - n\log(e) + \log(1+\Theta(\frac{1}{n}))
\end{aligned}$$

The highest order term is $n \log n$, from which we see that $\log(n!) = \Theta(n \log n)$, as desired.

(ii) First prove $\log(n!) = O(n \log n)$, then prove $n \log n = O(n!)$.

$$\begin{aligned}
\log(n!) &= \log(n \cdot (n-1) \cdot \ldots \cdot 1) \\
&= \log n + \log(n-1) + \ldots + \log 1 \\
&= \sum_{i=1}^{n} \log i
\end{aligned}$$

Since each term in the sum is $\leq \log n$, we see that $log(n!) \leq n \cdot \log n$, for $n > 0$, so $\log(n!) = O(n \log n)$.
We also know that at least $\frac{n}{2}$ of the terms in the sum are as large as $\log(\frac{n}{2})$, so, for $n > 0$,

$$\begin{aligned}
\sum_{i=1}^{n} \log i &\geq \frac{n}{2} \log \frac{n}{2} \\
&= \frac{n}{2}(\log n - 1) = \Omega(n \log n)
\end{aligned}$$

So, $n \log n = \Theta(\log(n!))$.

(iii) We can also bound the sum given in part b above using the integral method.

$$\int_1^n \log x dx \leq \sum_{i=1}^{n} \log i \leq \log n + \int_1^{n-1} \log(x+1) dx$$

$$x \log x - x|_1^n \leq \sum_{i=1}^{n} \log i \leq \log n + [(x+1)\log(x+1) - (x+1)]|_1^{n-1}$$

$$n \log n - n - (1 \log 1 - 1) \leq \sum_{i=1}^{n} \log i \leq \log n + n \log n - n - (2 \log 2 - 2)$$

$$n \log n - n + 1 \leq \sum_{i=1}^{n} \log i \leq \log n + n \log n - n + 2$$

Since both the upper and lower bounds are $\Theta(n \log n)$, the sum is also.

We took off 3 points if you showed only one bound (i.e. you failed to show both an upper and a lower bound, as in part b).

We took off 2 points if you did the integral incorrectly. $\int \log x dx = x \log x - x$, not just $x \log x$.

We generally took off 5 points if you arrived at a bound of $n \log n$ without showing sufficient steps.

(b) Prove or disprove: For functions f, g from positive integers to positive integers,

$$f(n) = O(g(n)) \implies 2^{f(n)} = O(2^{g(n)}).$$

Solution:

This proposition is false. If you "proved" it, we took off all 8 points, since your "proof" was obviously flawed.

The easiest way to show that this is false is to give a simple counterexample:

Let $f(n) = 2n$, $g(n) = n$. $2n$ is obviously $O(n)$. But 2^{2n} is not $O(2^n)$, as you proved in Problem Set 5, Problem 5b.