Problem Set 4 Solutions

Problem 1 An edge of a connected graph is called a *cut-edge* if removing the edge disconnects the graph. Prove that an edge e in a connected graph G is a cut-edge if and only if no simple cycle contains e.

Solution. First, we prove that if e is contained in a simple cycle, then e is not a cut-edge. Since G is connected, there exists a path P between every pair of distinct vertices. We must show that this remains true after e is removed. We will do this by considering the two nodes u and v on each side of e, and showing that u and v must still be connected by a different path P_2 when e is removed. This shows that all original paths between any two vertices in the path can be changed to include the path P_2 where the edge e used to be. Let us write out the simple cycle that contains e as $u, v, v_1, v_2, \ldots, v_n, u$. Because it is a simple cycle, all vertices are distinct (except for the double occurrence of u). This means that the edge e only occurs once in the cycle. When e is removed we can't travel along (u, v), however we can use the path $u, v_n, \ldots, v_2, v_1, v$. This implies that all vertices are still connected in the graph, because we have a "detour" around e when it is cut.

Next, we prove that if e is not a cut-edge, then e is contained in a simple cycle. We will make use of Theorem 1 on page 469 of Rosen, which states that there is a *simple* path between every pair of distinct vertices in a connected graph. Since e is not a cut-edge, the graph obtained by removing edge e is connected. So by the theorem, there exists a simple path P connecting the endpoints u and v of edge e, even after e is removed from the graph. Let $P = v, v_1, v_2, \ldots, v_n, u$. The path is simple so adjoining edge e to the end of P gives a simple cycle containing edge e. Specifically, $v, v_1, v_2, \ldots, v_n, u, v$.

□

Problem 2 Let $G = (V, E)$ be an undirected graph, and let $r : V \to \mathbb{R}$ be a function associating a real number with each vertex. Suppose r satisfies the property that for each $v \in V$ such that v has at least one neighbor, the value $r(v)$ is the average of the values of v's neighbors. Prove that if the graph is connected, then $r(u) = r(v)$ for all $u, v \in V$.

Solution. Any non-empty, finite set of real numbers is totally-ordered, and thus has a minimum element. So for any function r, there must be a *minimum vertex* u such that for all vertices $v \neq u$, $r(u) \leq r(v)$. Consider any valid function r and a connected graph G. Let u be a minimum vertex (which must exist). Since G is connected, either u is the only vertex (in which case the result follows trivially), or u must be adjacent to some non-empty set of nodes $\{v_1, \ldots v_k\}$, where $r(u) \leq r(v_i)$ for $i = 1, \ldots, k$. Since $r(u)$ is the average of all $r(v_i)$ for $i = 1, \ldots, k$, then we must have $r(u) = r(v_1) = \cdots = r(v_k)$, implying that $v_1, \ldots, v_k$ are all minimum vertices.

Therefore any minimum vertex must be adjacent to only minimum vertices. Since G is connected, this means that all vertices must be minimum vertices, and thus $r(u) = r(v)$ for all $u, v \in V$.

□

Problem 3 An n-node graph is said to be *tangled* if there is an edge leaving every set of $\lceil \frac{n}{3} \rceil$ or fewer vertices, that is, for every set of at most $\lceil \frac{n}{3} \rceil$ vertices, there is an edge between a vertex in the set and a vertex not in the set. As a special case, the graph consisting of a single node is considered tangled. (Recall that the notation $\lceil x \rceil$ refers to the smallest integer greater than or equal to x.)

(a) Find the error in the proof of the following claim.

Theorem? Every non-empty, tangled graph is connected.

Proof. The proof is by strong induction on the number of vertices in the graph. Let $P(n)$ be the proposition that if an n-node graph is tangled, then it is connected. In the base case, $P(1)$ is true because the graph consisting of a single node is defined to be tangled and is trivially connected.

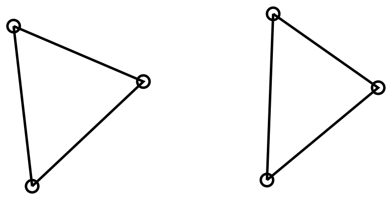
In the inductive step, for $n \geq 1$ assume $P(1), \ldots, P(n)$ to prove $P(n+1)$. That is, we want to prove that if an $(n+1)$-node graph is tangled, then it is connected. Let G be a tangled, $(n+1)$-node graph. Arbitrarily partition G into two pieces so that the first piece contains exactly $\lceil \frac{n}{3} \rceil$ vertices, and the second piece contains all remaining vertices. Note that since $n \geq 1$, the graph G has a least two vertices, and so both pieces contain at least one vertex. By induction, each of these two pieces is connected. Since the graph G is tangled, there is an edge leaving the first piece, joining it to the second piece. Therefore, the entire graph is connected. This shows that $P(1), \ldots, P(n)$ imply $P(n+1)$, and the claim is proved by strong induction. ■

Solution. The error is in the statement, "By induction, each of these two pieces is connected." This is not what the induction hypothesis says; the induction hypothesis states that *if a graph is tangled*, then it is connected. We can't be sure the two partitions are themselves tangled, in which case our inductive hypothesis says nothing about their connectivity. See 3(b).

□

(b) Draw a tangled graph that is not connected.

Solution.



□

(c) An n-node graph is said to be *mangled* if there is an edge leaving every set of $\lceil \frac{n}{2} \rceil$ or fewer vertices. Again, as a special case, the graph consisting of a single node is considered mangled. Prove that every non-empty, mangled graph is connected.

Solution. The proof is by contradiction. Assume for the purpose of contradiction that these exists an n-node graph that is mangled, but not connected. Then the graph must have at least two connected components. However, there can be at most one connected component with more than $\lceil \frac{n}{2} \rceil$ vertices, since the graph has only n vertices in total. Therefore, there exists a connected component with $\lceil \frac{n}{2} \rceil$ or fewer vertices. Since the graph is mangled, there is an edge leaving this component. But this contradicts the definition of a connected component.

□

Problem 4 A 2-colorable graph is called *bipartite* (two pieces).

(a) Prove by induction that every tree is bipartite.

Solution. We first prove the following claim:

Lemma. *By a* leaf *in a graph, we mean a vertex of degree* 1. *We prove that every tree on at least* 2 *nodes has a leaf.*

Proof. Let T be a tree, and let P be a simple path in T of maximum length. (Because we disallow repeated vertices, the length of any simple path is upper-bounded by the number of vertices in T. This implies there must be at least one simple path with maximum length.)

Suppose that $P = \{v_1, v_2 \ldots, v_k\}$ in this order. We claim that v_1 is a leaf. For if not, then there is an edge (v, v_1) in T, where $v \neq v_2$. Now if $v \notin P$, we can attach v to the end of P to obtain a longer path — a contradiction. But if $v \in P$, then since $v \neq v_2$, this will give a cycle in T — also a contradiction. ∎

Given this fact, we prove the theorem by induction on n.

Proof. Base case: $n = 1$. We can color the single node anything we want, thereby using at most two colors.

Inductive step. Suppose that we know that every tree on n nodes is bipartite, for some $n \geq 1$. Let T be a tree on $n + 1$ nodes; we need to show how to two-color it. Since $n + 1 \geq 2$, we know that T has a leaf v. Let w be the unique neighbor of v. Let $T' = T - \{v\}$. Note that T' is a tree, since it is still connected and has no cycles. Thus, by the induction hypothesis, we can two-color it.

Now we define the following two-coloring of T. If u is a vertex of T not equal to v, give u the color that it had in our two-coloring of T'. Then give v whichever color we didn't give to its neighbor w. As this is a valid two-coloring of T, this completes the inductive step and the proof. ∎

□

(b) Prove that a graph is bipartite if and only if every simple cycle is of even length. (**Hint:** Can v have both an even and an odd length path to w?)

Solution. Assume a graph has a simple cycle of odd length. Let the cycle be $v_1, v_2, \ldots, v_n, v_1$, where n is odd. Suppose that two colors (red and blue) sufficed to color the graph containing this circuit. Without loss of generality let the color of v_1 be red. Then v_2 must be blue, v_3 must be red, and so on, until finally v_n must be red (since n is odd). But this is a contradiction, since v_n

is adjacent to v_1. Therefore at least three colors are needed. Hence a graph is 2-colorable only if every simple cycle is of even length.

Conversely, assume the graph has no simple cycle of odd length. We show that it is possible to color the graph using two colors. Without loss of generality, assume that the graph is connected. (As otherwise we can repeat the argument for each connected component of the graph). Select a node, say m, in the graph and color it red. Color all the nodes adjacent to m blue. Call the set of all such nodes S_1. Now color all the nodes that are adjacent to the nodes in S_1 but have not yet been colored, red. Call the set of all such nodes S_2. Continue this process until you have colored all the nodes. We now show that no two nodes in S_i are connected by an edge, for all i. This is sufficient to establish that our coloring is valid, since two adjacent nodes are either in the same set S_i — which we are going to show can't happen — or are in two consecutive sets S_i and S_{i+1} that are colored differently. This is true because if one vertex gets included in S_i before its neighbor is colored, then the neighbor will be colored in the next step.

Assume that we can find two nodes u and w in some set S_i that are adjacent in the graph. If u and w are adjacent to a node in S_{i-1} then we have a simple cycle of length 3, which is impossible. So let $u_{i-1} \neq w_{i-1} \in S_{i-1}$ be nodes adjacent to u and w respectively. Now if u_{i-1} and w_{i-1} are adjacent to a single node in S_{i-2}, say c, then $u, u_{i-1}, c, w_{i-1}, w, u$ is a simple cycle of length 5, which is impossible. So let $u_{i-2} \neq w_{i-2} \in S_{i-2}$ be nodes adjacent to u_{i-1}, and w_{i-1} respectively. Continuing this way we arrive at nodes $u_1 \neq w_1 \in S_1$. However these are adjacent to m. This gives us a simple cycle $m, u_1, u_2, \ldots, u_{i-1}, u, w, w_{i-1}, w_{i-2} \ldots, w_1, m$, where $u_j, w_j \in S_j$. But this cycle has length $2i+1$, which is odd, hence a contradiction. Therefore, for all i we do not have two nodes in S_i that are connected by an edge. Therefore our coloring is valid.

$\square$

(c) Explain why (a) follows immediately from (b).

Solution. Since a tree has no simple cycles, it has no simple cycles of odd length, and thus (a) follows directly from (b).

$\square$

Problem 5 The following algorithm, called the *Russian peasants algorithm*, can be used to multiply any two natural numbers x and y using only left bit-shift (i.e., multiply by 2) and right bit-shift (i.e., divide by 2 and drop the remainder, if any) operations. The answer is accumulated in variable a; variables r and s are for temporary storage.

```
r := x;
s := y;
a := 0;
do until s = 0:
    if s is even then
        r := 2r;
        s := s/2;
    else
        a := a + r;
        r := 2r;
        s := (s − 1)/2;
```

The answer xy is the value left in the accumulator a when the procedure terminates.

(a) Model the algorithm as a state machine. That is, define the states Q, the start states Q_0, and the steps δ.

Solution.
$Q = \{(r, s, a, x, y) | r, s, a, x, y \in \mathbb{N}\}$
$Q_0 = \{(x, y, 0, x, y)\}$
$\delta : (r, s, a, x, y) \to (r', s', a', x', y')$, where
$x' = x$, $y' = y$, $r' = 2r$
Even s: $s' = s/2$, and $a' = a$
Odd s: $s' = (s-1)/2$, and $a' = a + r$

$\square$

(b) List the sequence of states that appears in an execution of the algorithm for inputs $x = 5$ and $y = 9$.

Solution. $(5, 9, 0, 5, 9) \to (10, 4, 5, 5, 9) \to (20, 2, 5, 5, 9) \to (40, 1, 5, 5, 9) \to (80, 0, 45, 5, 9)$

$\square$

(c) Prove that the algorithm gives the correct answer if it terminates, that is, that $s = 0$ implies $a = xy$. (Hint: Use induction on the number of steps in the execution. You will have to strengthen the inductive hypothesis.)

Solution. We prove the following invariant.

$P(r, s, a, x, y) =$ "$xy = rs + a$".

Base case: For any initial state we have $P(x, y, 0, x, y)$ which is true.

Inductive step: Consider a transition from (r, s, a, x, y) to (r', s', a', x', y'). We need to prove that $P(r, s, a, x, y) \to P(r', s', a', x', y')$.

There are two cases based on the type of action, *even* or *odd*.

even:
Doubling r while halving s doesn't change the product rs, and a doesn't change. So $r's' + a' = rs + a = xy$.

odd:
$r' = 2r$, $s' = \frac{s-1}{2}$, and $a' = a + r$. So $r's' + a' = 2r\frac{s-1}{2} + a + r$. Reorganizing terms yields $r's' + a' = rs + a = xy$.

The invariant is proved. Now when $s = 0$ this implies $a = xy$ and we have our result.

$\square$

(d) Prove that the algorithm terminates.

Solution. We first notice the invariant that $s \geq 0$ for any reachable execution state. We could prove that this is an invariant by showing it is true for the base case, $y \geq 0$, and furthermore, no transition exists from a state with $s \geq 0$ to $s < 0$ when s is odd or even. A more advanced argument is to say that this is enforced by the "type" of s, which is that $s \in \mathbb{N}$.

Now consider the value of s during any valid execution. Since $s \geq 0$, as shown above, no transition increases s. Furthermore, for $s > 0$, all transitions decrease s by at least 1. Thus for any values of $x, y \in \mathbb{N}$, s decreases to 0 in a finite number of steps, and thus the program terminates in a finite number of steps.

□

Problem 6 Attached to this problem set are some students' solutions from a previous problem set. Mark up these solutions, indicating things that are incorrect or unclear, and if possible show how the solution can be made completely correct and clear.

(a) Problem Set 2, Problem 1

Solution. A number of different answers are acceptable here. To guarantee full credit, however, you should point out the inherent hand-waviness going on with the statement "and imposes constraints on." Correctly understanding the constraints imposed on regions was critical to the correct solution of the problem. This was not addressed by the two cases presented.

Saying that the proof had "buildup error" is another reasonable answer, but does not address the point above.

□

(b) Problem Set 2, Problem 4

Solution. The cases covered in the proof are not exhaustive. Does he/she cover the case when the added square has 7 neighboring squares? Of course, this doesn't really matter if one talks about the number of adjacent edges, but this was not the approach taken here.

Another reasonable partial answer is that it was never clearly stated where the inductive hypothesis was being used.

□

(c) Problem Set 2, Problem 8

Solution. There are a number of correct answers for this problem. There were several cases where variables were defined haphazardly (like Q), or incorrectly (like R=ricky is a set). Also, the final answer is stated without a real logical argument from the preceding step.

In part (b) there is no clear mention of a locally fair ranking (among other things).

□