CS 3510: Design and Analysis of Algorithms
Section A, Spring 2017

Homework 2
Due: Monday 11:55PM, Feb 13th, 2017

Name: Alan Chiang

Notes:

1. Please don't write a program when you describe an algorithm. You should give an idea of what the algorithm does and then discuss the details of how your algorithm does specific tasks. If you give pseudo-code, you should explain your code.

2. For every algorithm that you are asked to describe, please clearly list the steps of your algorithm and explain its correctness and runtime behavior. We recommend separating the algorithm, correctness argument, and runtime analysis into three paragraphs.

3. You can assume that basic operations on integers which are not described to have a variable length (addition, multiplatication, subtraction, floor, ceiling, absolute value, comparison of less than, greater than, or equals, and anything you can form out of a constant number of these) can be done in constant time. You can also assume that, given i, you can read a_i from a list $a_1, ...a_n$ in constant time.

1. (10 points) Give an $O(|V| + |E|)$ algorithm to determine if undirected graph $G = (V, E)$ has a cycle of odd length.

   ANSWER: We begin by performing BFS on the graph. When a vertex is visited, it is labelled either red or blue (or any other boolean flag) as well as adding it to the visited list. The root is labelled red. All neighbors of the root are labelled blue. All of their neighbors are red, and so on so that the colors alternate by depth. The BFS terminates once we have visited and colored all vertices.

   We then run BFS again. This time we keep track of the current vertex's color and compare it to the color of all its neighbors. This gives us two cases: 1. The two adjacent vertices have the same color. We then return true, because this means there exists an odd-length cycle in the graph. We know because if two adjacent vertices are the same color, the graph is not 2-colorable. A graph is 2-colorable if and only if it is bipartite. A graph is bipartite if and only if it has no odd cycles. Therefore, if the two adjacent vertices are the same color, the graph posseses a cycle of odd length. 2. The two vertices are of different color. In this case we continue BFS.

   The second BFS either hits case 1, or it completes traversing the graph without hitting case 1. If BFS completes, we return false because there is no cycle of odd length in the graph.

   This solution runs in $O(|V| + |E|)$ time, because BFS runs in $O(|V| + |E|)$ time. We run BFS twice, and drop the constant 2.

2. (10 points) Let $G = (V, E)$ be a connected undirected graph. We say that an edge e is a *bridge* of G if and only if $G' = (V, E - e)$ is disconnected.

Prove the following statements or give a counterexample if they are false.

(a) An edge $e \in E$ is a bridge if and only if it is not a part of any simple cycle in G.

(b) Let $e = (v_1, v_2)$. If v_1 and v_2 are both articulation points, e is a bridge of G.

ANSWER: a. This is a biconditional, so we must prove both ways.

1. If e is not part of any simple cycle in G, e is a bridge. This is a true statement. To prove it, we begin with an edge e in our graph G. If e is not in any simple cycle and G is a connected undirected graph, e is the only path between the two vertices which e connects. We know this because the problem states that the graph $G' = (V, E - e)$ is disconnected, so that means e must be the only connection between its two vertices. If we remove e (forming new graph G'), then the two vertices will no longer be connected and thus be unreachable to each other. Since it has at least one unreachable vertex, G' must have at least two disconnected components. Because removing e from G will disconnect the resulting graph G', we know that e is a bridge.

2. If e is a bridge, it is not part of any simple cycle in G. This is a true statement. To prove it, we begin with a bridge e in our graph G. If e is a bridge, we know by the given definition of a bridge (in the problem statement) that it must be the only connection between its two vertices. A simple cycle in an undirected graph will always have two paths to every vertex in the cycle (clockwise and counterclockwise). If the bridge e is the only connection, there cannot be another path to its vertices, and so thus e cannot be part of a simple cycle.

b. This is false. If both articulation points are part of the same cycle, then we know that there are two paths to reach both vertices because G is a connected undirected graph. Removing e will remove one of the paths, but since the other will remain, the two articulation points will remain connected and thus the graph will also remain connected, so e cannot be a bridge.

3. (10 points) Let $G = (V, E)$ be an undirected graph. In class we saw that DFS can be used to identify articulation points in a graph.

   (a) Show that the root of the DFS tree is a articulation point if and only if it has more than one child in the tree.

   (b) Show that a non-root vertex v of the DFS tree is a separating vertex if and only if it has a child v' none of whose descendants (including itself) has a backedge to a proper ancestor of v.

ANSWER:

a. This is a biconditional statement and must be proved both ways.

1. If the root has more than one child, the root is an articulation point. This is true. By definition, the root has no parents. By the definition of child, siblings cannot be directly connected to each other, otherwise they would be parent-child instead of siblings. Thus, if the root has more than one child, the only way to access these children (and their subtrees) is through the root. Therefore, if the root is removed, there is no path between the different children's subtrees, so they are disconnected and the root is an articulation point.

2. If the root is an articulation point, it has more than one child. This is true by contrapositive. If the root has only one child, removing the root will not prevent that single child from accessing itself or any of its own children because every vertex has a path to itself and its own children by definition. Thus, removing the root produces no disconnected components and thus the root is not an articulation point.

b. This is a biconditional statement and must be proved both ways.

1. If neither v' nor its children have a backedge to a proper ancestor of v, then v is a seperating vertex. This is true. Without any backedges, the only connection between v + its ancestors and the subtree of v' + its children is the edge between v and v'. This means that if v is removed from the graph, there will no longer be any path from the ancestors of v to the subtree of v', so the graph will be split into two disconnected components. Therefore, v is a seperating vertex.

2. If v is a seperating vertex, then neither its child v' nor that child's descendants have any backedges to a proper ancestor or v. This is true. If v is a seperating vertex, its removal will divide the graph into at least two disconnected components, formed of the ancestors of v and the children of v. In order for two components to be disconnected, there can be no path between them. Because any back edges would form a path and therefore connect the disconnected components, contradicting the definition of a seperating vertex, there can be no back edges between the children and ancestors of v.

4. (10 points) The police department in the city of Computopia has made all streets one-way. The mayor contends that there is still a way to drive legally from any intersection in the city to any other intersection, but the opposition is not convinced. A computer program is needed to determine whether the mayor is right. Provide a linear time algorithm that can be used to verify the claim.

   ANSWER:

   This problem is identical to finding an algorithm which discovers all the strongly connected components in a graph. If the entire graph is in a single strongly connected component, such that any vertex may be reached from any other vertex in the digraph, then the mayor is right. We begin by performing DFS. Once DFS is finished, we compare the size of the visited list to the size of the graph. If the two sizes are equal then we continue with the algorithm. If not, we return false because not all vertices have been reached, so not all intersections may be reached from all others and the mayor is wrong.

   Assuming the algorithm doesn't terminate then reverse the directions of the edges in the DFS traversal tree, and run DFS again on the reversed tree. After this DFS completes, we check the second DFS's visited list. If this list's size is also equal to the size of the graph, then every vertex within the graph is accessible in both directions and the entire graph is a single strongly connected component. Thus we return true, because the mayor is right.

   This algorithm runs in linear time. The initial DFS runs in $O(|V|+|E|)$ time, the edge direction reversal runs in linear time as well because it must traverse the entire adjacency list, and the second DFS also runs in $O(|V|+|E|)$. These three parts of the algorithm run in sequence, not simultaneously, so the entire algorithm runs in linear time.

5. (10 points) An *implication graph* for a given 2-SAT formula has a vertex for every literal of the formula (two literals for each variable). Prove that a 2-SAT formula can be satisfied if and only if its implication graph does not have any strongly connected components containing both x and $\bar{x}$ for some variable x.

ANSWER:

This is a biconditional statement, so it must be proven both ways. 1. If the implication graph has no strongly connected components containing both x and $\bar{x}$, then the 2-SAT formula can be satisfied. We begin with the implication graph. x and $\bar{x}$ not being in the same strongly connected component means that x does not imply $\bar{x}$ and $\bar{x}$ does not imply x. If either of the above implications were true, then x would have to be both true and false simultaneously, which is clearly impossible. Because neither implication is true, this means that x does not have to be both true and false simultaneously, so there is no such logical contradiction in the 2-SAT formula. Without that logical contradiction, the 2-SAT formula can then be satisfied by the appropriate values of x and $\bar{x}$.

2. If the 2-SAT formula is satisfied, then the implication graph does not have any strongly connected components containing both x and $\bar{x}$. Because the 2-SAT formula is satisified, there exists some assignment of variables x and $\bar{x}$ such that the formula does not contain a logical contradiction that x is both true and false. This being the case, we know that the corresponding implication graph does not contain any logical contradiction either. x and $\bar{x}$ being in the same strongly connected component represents a logical contradiction whereby x implies $\bar{x}$ or $\bar{x}$ implies x. We are given that the 2-SAT formula is already satisfied, and that being the case, then the implication graph cannot have any strongly connecte components containing both x and $\bar{x}$.