

Assignment 16

The following are to be written up and turned in separately from the rest of the homework.

1. (a) Design a pseudo code algorithm, `isValidRBTree(T)`, using the EulerTour template, to determine whether or not a red-black tree is properly formed. That is, are there two consecutive red nodes and are the black heights of both children of every node the same.
(b) Implement your function in JavaScript using the EulerTour template.
 2. Using the DFS Template Method algorithm given in the lecture notes, give a pseudo code algorithm, `isConnected(G)`, to determine true or false, whether or not a graph G is connected, i.e., define `isConnected` and override the appropriate hook methods so this algorithm determines connectivity of a graph G .
 3. Using the DFS Template Method algorithm given in the lecture notes, give a pseudo code algorithm, `hasCycle(G)`, that determines true or false, whether or not a graph G has a cycle.
 4. Using the DFS Template Method algorithm given in the lecture notes, define a pseudo code algorithm, `connectedComponents(G)`, that computes the connected components of a graph G . Your method should return a sequence of vertices, 1 representative from each connected component.
6. a. Modify the breadth-first search algorithm so it can be used as a Template Method.
b. Give a pseudo code algorithm, `findPath(G, u, v)` that finds a path between u and v . You will need to override the appropriate methods so that given two vertices u and v of graph G , your BFS finds a path in G between them, or report that no such path exists. Note that this path will be a path with the minimum number of edges.
c. Can the template version of DFS be used to find the path between two vertices with the minimum number of edges? Briefly explain why or why not.
7. Based on either the DFS or the BFS template method algorithms, write an algorithm, `labelComponents(G)`, that labels each connected component with a different number, that is, each vertex in a component would be labeled with the same number. For example, each node in the first connected component would be labeled with a 0, each node in the second connected component would be labeled with a 1, etc.