Preliminary assignment - Paths in a Graph

Author: Anders Nylund

February 9, 2017

Application number: xxxxxx

Abstract

This document is the answer of applicant Anders Nylund to the Preliminary assignment "Paths in a Graph". The assignment is part of the admission process to Computer Science at University of Helsinki.

Task 1

```
1.  InfPath(G, v)
2.     return FindNodesRecursive(G, v, [])
3.
4.  FindNodesRecursive(G, v, checkedNodes)
5.  if checkedNodes contains v
6.     return true
7.  else
8.     returnValues = []
9.     foreach endNode in v.endNodes
10.       returnValues.add(FindNodesRecursive(G, endNode, checkedNodes)
11.       if returnValues contains true
12.          return true
13.    return false
```

The method InfPath takes a graph and an integer as argument and checks if an infinite path can be found, starting from the node with the index of the passed integer. InfPath returns the value of the method FindNodesRecursive. As InfPath takes only a graph and an integer as argument, FindNodesRecursive-method needed to be created as it adds one argument more. The pseudo code has the following assumptions. The arguments in the methods are passed by value, not reference. Every node has an iterable list of nodes that it has an arc to.

Instead of having comments in between the pseudo code, line numbers has been added before every line. This allows explanation of each row individually without making the pseudo code hard to read.

1. Declaration of method InfPath that takes a graph and an integer as argument.
2. InfPath returns the return value of FindNodesRecursive(). Here the graph G, starting node v and an initialized empty list is passed as argument.
3. -
4. Declaration of FindNodesRecursive. The arguments are given the same names as when passing them, except for the empty list, which is now called checkedNodes.
5. A control statement that searches for v in the list of checked nodes. The list of checked nodes consist of the previously checked nodes. If the currently investigated node has already been checked, it means that an infinite path has been found
6. If the statement before was true, the value true is returned
7. If the statement on line 5 was false the following block of code will be executed

8. Initialize a new list. This list will be populated of boolean values that are returned from FindNodesRecursive that is called recursively.

9. Iterate through every end node of the current start node

10. Call the method FindNodesRecursive and add the return value to the list of return values

11. On each iteration of end nodes, check if the current iteration returned true.

12. If preceding statement was true, return true

13. If the execution has made it to this point, no infinite paths were found, and the method returns false

The idea of the algorithm is to create a 'tree' of recursive calls. For every end node in the current node in iteration, a new call to find the end nodes is made. The recursive method is built to return true as fast as possible when an infinite path is found in the graph.

Task 2

```
Some fancy code
```