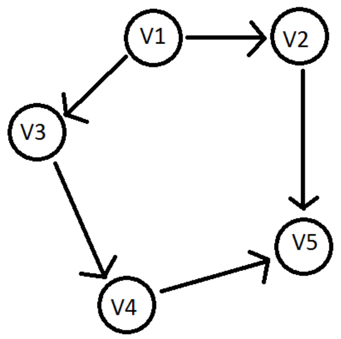
Stephen Woodbury   5/22/17   CMPS 102
I have read and agree to the collaboration policy. Stephen Woodbury.
Collaborators: none

Assignment 3_3 : Dynamic Programming : Long Path

a) Proving an Algorithm Wrong



Our Algorithm would yield the path: 1->2->5 where L = 3.
The correct answer should be 1->3->4->5 where L = 4.

b) Algorithm

Algorithm Components:

nodesLeadingToNode : Array of arrays of nodes that immediately lead up to a spec. node
maxLength: int length of longest path, init to 0.
tempLength: int length temp holder. Init to 0.

Algorithm

```
SetUp nodesLeadingToNode
int findLongestPath(n)

findLongestPath(j) {
  if(j <= 0)
    return 0
  for(all members m of nodesLeadingToNode(j))
    tempLength = findLongestPath(m)
    if(maxLength < 1+tempLength)
      maxLength = 1+tempLength
  return maxLength
}
```

Algorithm Description:

Set up the data structure *nodesLeadingToNode.* Call findLongestPath on n, the end node of the path. Init Length to 0. In *findLongestPath*, check if the node it was called on is equal or less than zero, if so, return 0. For all members that lead to the node called on, analyze. Determine the longest Path to them from node 1 by calling *findLongestPath(member). Compare each of the previous node's longest path lengths, if that plus 1 is longer than the current longest path known for the node the function was ori called on, replace the longest path length for the node. Once done iterating through all members, return the maxLength.*

Proof of Correctness : Termination

Claim: The algorithm terminates.

Pf: For every call of *findLongestPath*, nodes previous to the current node called on are analyzed, never is a node already analyzed called. There are only n nodes, meaning the function must terminate.

<u>Proof of Correctness : Longest Path Found:</u>
Claim: The algorithm returns the length of the longest path from node 1 to node n.
Pf: The algorithm accounts for every possibility of longest path for every node and if that length+1 is longer than the current longestPathLength for a specific node, that longestPathLength is immediately changed. Therefore the path length must be maximized for every node, including the n'th node from the first node.'

<u>RunTimeAnalysis:</u>
1) O(e) 2)Recurrence based.
T(n) = 5 + O(e)+ T(n-1).
n is really just the numbers of vertices.
Runtime is ***O(ve)*** where v = number verts and e = number edges.

<u>Space Analysis:</u>
O(e) for the data type nodesLeadingToNode.