Graphs

Sai Ashirwad R

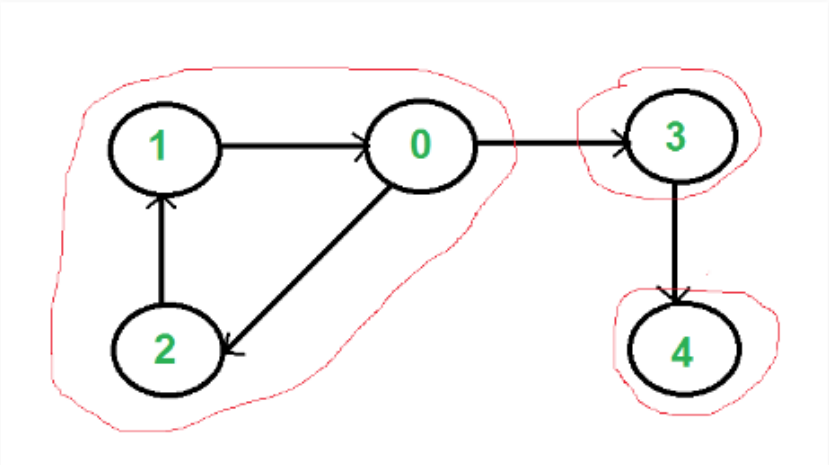
October 13, 2020

Contents

1 Strongly Connected Components

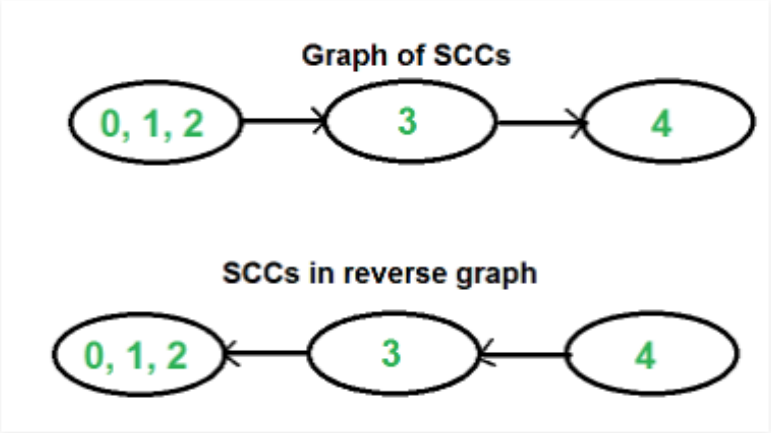
1.1 Kosaraju's Algorithm

Code C++



1.1.1 Description

- find all strongly connected components in O(V+E)
- DFS two times
  - DFS of graph produces single tree if all vertices reachable from starting point
  - else: forest
- may find tree or forest depending on where start
  - 0, 1, 2 -> tree
  - 3, 4 -> forest
- **finish time of a vertex that connects to other SCCs will always be greater than finish times of vertices in the other SCC**



1.1.2 Datastructures:

- stack s

1.1.3 Method

1. DFS traversal of graph
   - create empty stack S
   - recursive DFS traversal for adjacent vertices of vertex -> push vertex to stack

- starting from 0: 1, 2, 4, 3, 0

2. Reverse directions of all arcs -> transpose graph
3. DFS on transpose graph
    - while S is not empty
        - pop vertex v from S
        - take v as source -> DFSUtil()
        - DFS from v -> strongly connected components of v