SCC0602 - Algoritmos e Estruturas de Dados I

Strong Connected Components

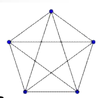
Professor: André C. P. L. F. de Carvalho, ICMC-USP
PAE: Rafael Martins D'Addio
Monitor: Joao Pedro Rodrigues Mattos

Today

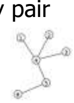
- Strongly connected components
- Transpose of directed graphs
- Algorithm to find strongly connected components

Introduction

- ***Complete graph***: undirected graph in which every pair of vertices is adjacent
  - There is an edge between each pair of nodes



- An undirected graph is ***connected*** if every pair of vertices is connected by a path
  - It has exactly one connected component
    - Every vertex is reachable from every other vertex



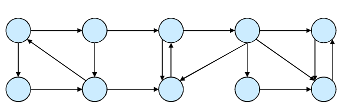
- A directed graph is ***strongly connected*** if every two vertices are reachable from each other
  - It has only one strongly connected component

Strongly connected components

- Classical application of DFS:
  - Decomposition of a directed graph into its strongly connected components
  - Performed by two DFS
  - Many algorithms that work with directed graphs begin with a decomposition
    - After, the algorithm is run separately in each strongly connected component
    - Final solution = combination of the separately created solutions

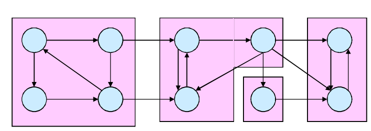
Strongly connected components

- Strongly connected component (*SCC*) of a directed graph $G=(V,E)$
  - Maximal set of vertices $C \subseteq V$ such that for all $u, v \in C$, there is both $u \rightsquigarrow v$ and $v \rightsquigarrow u$
  - Vertices *u* and *v are* reachable from each other



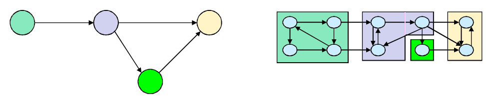
Strongly connected components

- Strongly connected component (*SCC*) of a directed graph $G=(V,E)$
  - Maximal set of vertices $C \subseteq V$ such that for all $u, v \in C$, there is both $u \rightsquigarrow v$ and $v \rightsquigarrow u$
  - Vertices *u* and *v are* reachable from each other



Component

- $G^{SCC} = (V^{SCC}, E^{SCC})$
- V^{SCC} has one vertex representing each SCC in G
- E^{SCC} has the existing edges between the corresponding SCCs in G
- G^{SCC} for the previous example:



G^{SCC} is a DAG

Lemma 22.13

Let C and C' be distinct SCCs in G, let $u, v \in C$ and $u', v' \in C'$, and suppose there is a path $u \rightsquigarrow u'$ in G

Then there cannot also be a path $v' \rightsquigarrow v$ in G

Proof:

- Suppose there is a path $v' \rightsquigarrow v$ in G
- Then there are paths $u \rightsquigarrow u' \rightsquigarrow v'$ and $v' \rightsquigarrow v \rightsquigarrow u$ in G
- Therefore, u and v' are reachable from each other, being not in separate SCCs

Transpose of a Directed Graph

- G^T = **transpose** of directed G
  - $G^T = (V, E^T)$, $E^T = \{(u, v) : (v, u) \in E\}$
  - G^T is G with all edges reversed
- Can create G^T in $\Theta(V + E)$ time if using adjacency lists
- G and G^T have exactly the *same* SCCs
  - u and v are reachable from each other in G if and only if reachable from each other in G^T

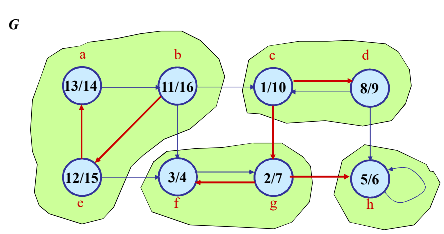
Algorithm to find SCC

SCC(G)
1 Call DFS(G) to compute finishing times f [u] for all u
2 Compute G^T
3 Call DFS(G^T), but in the main loop, consider vertices in f [u] decreasing order (as computed in the first DFS)
4 Output the vertices in each tree of the depth-first forest formed in the second DFS as a separate SCC

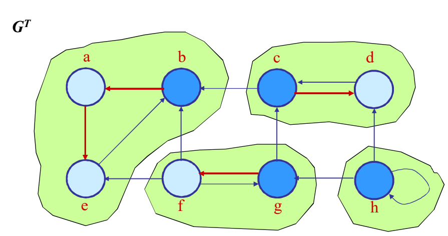
Running time: $\Theta(V + E)$

Example

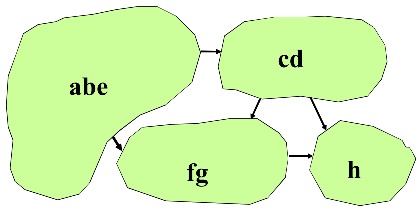
(Courtesy of Prof. Jim Anderson)



Example



Example



André de Carvalho - ICMC/USP 13

How does it work?

- **Idea:**
  By considering vertices in the second DFS in decreasing order of finishing times from the first DFS, vertices of the component graph are visited in topologically sorted order
  - Because DFS is running on G^T, there will be no visit to v from a u, where v and u are in different components
- **Notation:**
  - $d[u]$ and $f[u]$ always refer to the *first* DFS
  - Extend notation for d and f to sets of vertices $U \subseteq V$:
    - $d(U) = \min_{u \in U}\{d[u]\}$ (earliest discovery time)
    - $f(U) = \max_{u \in U}\{f[u]\}$ (latest finishing time)
    - U can be vertices from a SCC

André de Carvalho - ICMC/USP 14

Applications

- Usually, SCC is part of an application
- Social Networks
  - Look for a group of people that are generally strongly connected (similar tastes)
- Software Engineering
  - Look for classes that are SCC in large software
- Web
  - Look for related pages in the internet

André de Carvalho - ICMC/USP 15

Next Lecture

- Weighted Graphs
- Minimum Spanning Trees
  - Greedy Choice Theorem
  - Kruskal's Algorithm
  - Prim's Algorithm

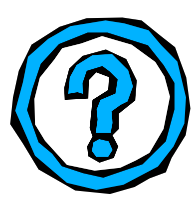
André de Carvalho - ICMC/USP 16

Acknowledgement

- A large part of this material were adapted from
  - Simonas Šaltenis, Algorithms and Data Structures, Aalborg University, Denmark
  - Mary Wootters, Design and Analysis of Algorithms, Stanford University, USA
  - George Bebis, Analysis of Algorithms CS 477/677, University of Nevada, Reno
  - David A. Plaisted, Information Comp 550-001, University of North Carolina at Chapel Hill

André de Carvalho - ICMC/USP 17

Questions



André de Carvalho - ICMC/USP 18