CSE 101 – Nov 18, 2019 (Week 8)

Notes provided by Ben Sihota bsihota@ucsc.edu

Example: Edge classification

[. . .]

Example: Undirected graph edge classification

[. . .]

No forward (by definition) or cross edges (by theorem on P. 547) in undirected graph

Exercise: Rewrite DFS so that it prints each edge and its classification as it runs

Lemma

A digraph G is acyclic iff DFS(G) yields no back edges

Corollary (equivalent to lemma) A digraph G contains a directed cycle iff DFS(G) yields a back edge

Proof (right to left) Obvious

Proof (left to right) Uses **white path theorem**

Suppose G contains a directed cycle, call it C. Let y be the first vertex on c to be discovered. Let x be the origin of the edge that has y as its terminus. At time $d[y]$, G contains a white y to x path. Namely: all edges of C other than (x, y). By the white path theorem, x is a descendent of y when DFS is complete. Therefore $x \breve{} y$ is a back edge in G.

Topological sort

In 23.4 and 23.5 (and PA5) all directional graphs

Definition (**Acyclic**, means digraph contains no directed cycles)

A digraph that is acyclic is a **DAG: directed acyclic graph**

Let G be a DAG. A **topological sort** of $V(G)$ is a linear ordering of $V(G)$ such that if $(x, y) \in E(G)$, then x comes before y in the linear ordering.

Example $(1) \to (4) \to (2) \to (3)$

Fact: We can use DFS to find a topological sort in a DAG.