Contents

% Chapter 05 — Graph Search and Topological Order % CS161 Reader (Plotkin F10) %

1 Chapter 05 — Graph Search and Topological Order

Graph algorithms are where *invariants* become unavoidable: without them DFS/BFS are just vibes.

Primary patterns: invariant, induction (structural recursion).

Executable witnesses: `cs161lab.algorithms.graphs.dfs`, `cs161lab.algorithms.graphs.toposort`.

1.1 5.1 Graph model and notation

Let $G = (V, E)$ be a directed graph with $|V| = n$ and $|E| = m$, represented with adjacency lists.

A directed graph is a **DAG** iff it has no directed cycle.

1.2 5.2 DFS (Depth-First Search)

1.2.1 Algorithm

Maintain a state for each vertex: white (unvisited), gray (active), black (finished). When exploring v: 1. mark gray 2. for each neighbor u: - if white: DFS(u) - if gray: found a back edge (cycle witness) 3. mark black; append v to a finishing list

Trace events: `enter`, `edge`, `back_edge`, `exit`.

1.2.2 Invariant 5.1 (Stack invariant)

At any time, gray vertices are exactly those on the current recursion stack. Any edge from a gray vertex to a gray vertex is to an ancestor.

Proof. Immediate from the definition of when vertices become gray/black. □

1.2.3 Lemma 5.2 (Back edge implies cycle)

If DFS discovers an edge (v, u) with u gray, then the graph contains a directed cycle.

Proof. Since u is on the recursion stack, there is a directed path from u to v using DFS tree edges. Adding (v, u) closes a cycle. □

1.3 5.3 Topological ordering

A topological ordering is an ordering $(v_1, \ldots, v_n)$ such that every edge points forward.

1.3.1 Theorem 5.3 (DFS finishing times yield topo order)

If G is a DAG, sorting vertices by **decreasing finishing time** produces a topological order.

Proof. Consider any edge (v, u). When DFS explores (v, u): - if u is white, DFS must finish u before finishing v; - if u is gray, that's a back edge implying a cycle, impossible in a DAG; - if u is black, then u already finished. Thus $f(v) > f(u)$ for all edges (v, u), so decreasing finishing time orders edges forward. $\square$

Executable witness: `cs161lab.algorithms.graphs.toposort`.

1.4 5.4 Complexity

DFS runs in $O(n + m)$ time and uses $O(n)$ extra space (recursion stack + state).

1.5 Appendix — Trace events as proof witnesses

- `back_edge` is a concrete cycle witness.
- finishing order corresponds to the topological theorem's $f(v) > f(u)$ condition.