Computer Algorithms I

Spring 2020

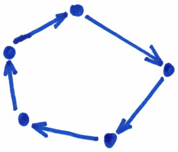
Topological sorting

directed graph $G = (V, E)$ of jobs to be performed in some order

$(u, v) \in E$: job u must come before job v

find a good ordering of the jobs!

is that always possible?



find a good ordering of the jobs if possible and output "not possible" otherwise!

Topological sorting algorithm

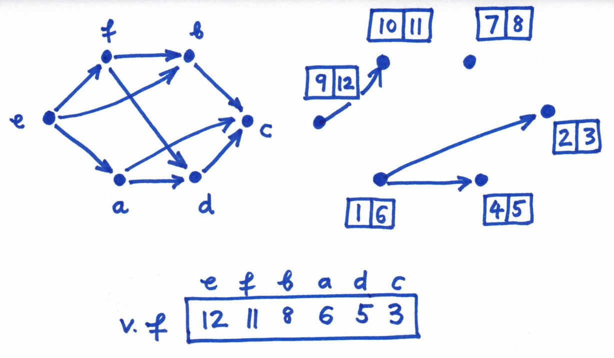
run depth-first search and order vertices according to decreasing order of finishing times

if this ordering does not work output "not possible"

else output ordering found

complexity $O(|V| + |E|)$, so optimal

Topological sorting example



Correctness I

if G has a directed cycle then it has no topological sorting: algorithm must be correct in this case

G is acyclic: no directed cycles

Claim 1: if G is acyclic then it has a topological sorting and the algorithm finds it!

Edge types in acyclic graphs

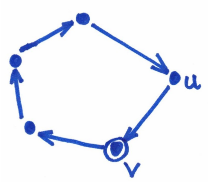
Theorem

G is acyclic iff there are no back edges in DFS forest

G is cyclic iff there are back edges in the DFS forest

$\Leftarrow$: if there is a back edge (u, v) then G is cyclic: $v \to \ldots \to u$

$\Rightarrow$: if G is cyclic then there is a back edge



let v be the vertex discovered first, then by the white path theorem u becomes a descendant of v, so (u, v) is a back edge

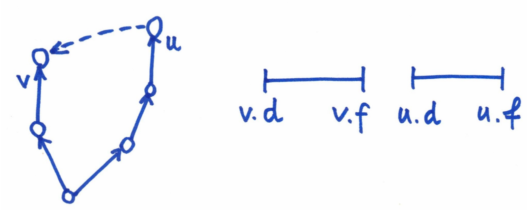
Correctness II

Claim 1 again: if G is acyclic then it has a topological sorting and the algorithm finds it!

if G is acyclic then every edge is a tree edge, a forward edge or a cross edge

Claim 2: for each such edge (u, v) it holds that $u.f > v.f$

for tree edges and forward edges follows from parenthesis theorem



cross edges (both types): intervals disjoint, if $[u.d, u.f] < [v.d, v.f]$ then v becomes a child of u by the white path theorem