http://www.cs.unh.edu/~ruml/cs758
Topological Sorting

- The Problem
- Break

Union-Find
The Problem

Given a set of pairwise orderings $a \prec b$, find an ordering of all the elements that respects them or detect that no such ordering is possible.

How long does this take?
Topological Sorting

- The Problem
- Break

Union-Find

- midterm
- asst 8
Union-Find
Problem: find components in an undirected graph and answer membership queries

Two cases: static vs dynamic

How can we identify components in the static case?
**Union-Find ADT**

- **MAKE-SET**(\(x\))  makes new set containing \(x\)
- **UNION**(\(x, y\))  combine the set containing \(x\) with the set containing \(y\)
- **FIND-SET**(\(x\))  return a representative of the set containing \(x\)
find-components
1. foreach vertex \( v \)
2. \text{MAKE-SET}(v)
3. for each edge \((u, v)\)
4. \text{UNION}(u, v)

\text{in-same-component?}(u, v)
5. is \text{FIND-SET}(u) = \text{FIND-SET}(v)?
Disjoint Sets

set is a tree rooted at representative

How to implement make, union, find?
### Speed-Ups

**union by rank**  track approximate height, put shorter under taller

**path compression**  after \texttt{FIND-Set}, ensure touched nodes point directly to root
**Pseudo-code**

**Make-Set(x)**
1. \( x.p \leftarrow x \)
2. \( x.rank \leftarrow 0 \)
3. **Union(x, y)**
4. \( x \leftarrow \text{Find-Set}(x) \)
5. \( y \leftarrow \text{Find-Set}(y) \)
6. if \( x.rank > y.rank \)
   7. \( y.p \leftarrow x \)
   8. else
   9. \( x.p \leftarrow y \)
10. if \( x.rank = y.rank \)
11. increment \( y.rank \)
**More Pseudo-code**

Find-Set($x$)
1. if $x \neq x.p$
2. $x.p \leftarrow \text{Find-Set}(x.p)$
3. return $x.p$

For $m$ operations on $n$ sets, worst-case time is $O(m\alpha(n))$.

$\alpha(n)$ is inverse of Ackermann’s function. It is $\leq 4$ if $n \leq 2^{2048} = 16^{512}$. 
For example:

- What’s still confusing?
- What question didn’t you get to ask today?
- What would you like to hear more about?

Please write down your most pressing question about algorithms and put it in the box on your way out.

*Thanks!*