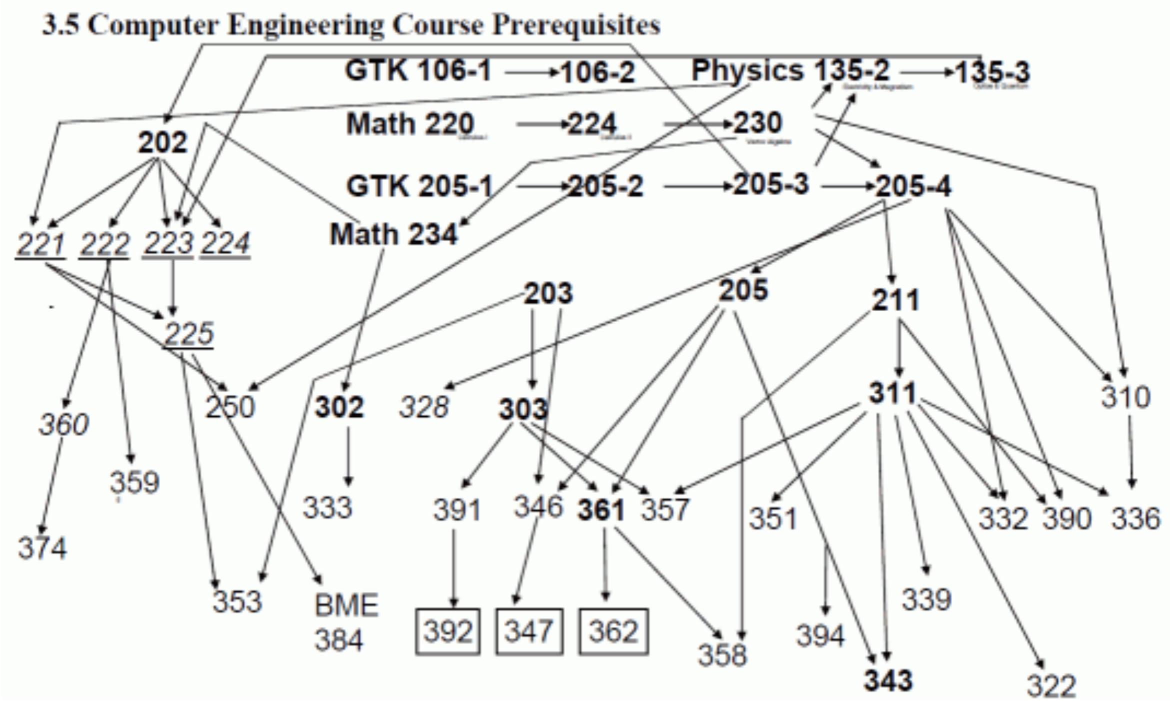
TOPOLOGICAL SORT

CS340

Topological Sort

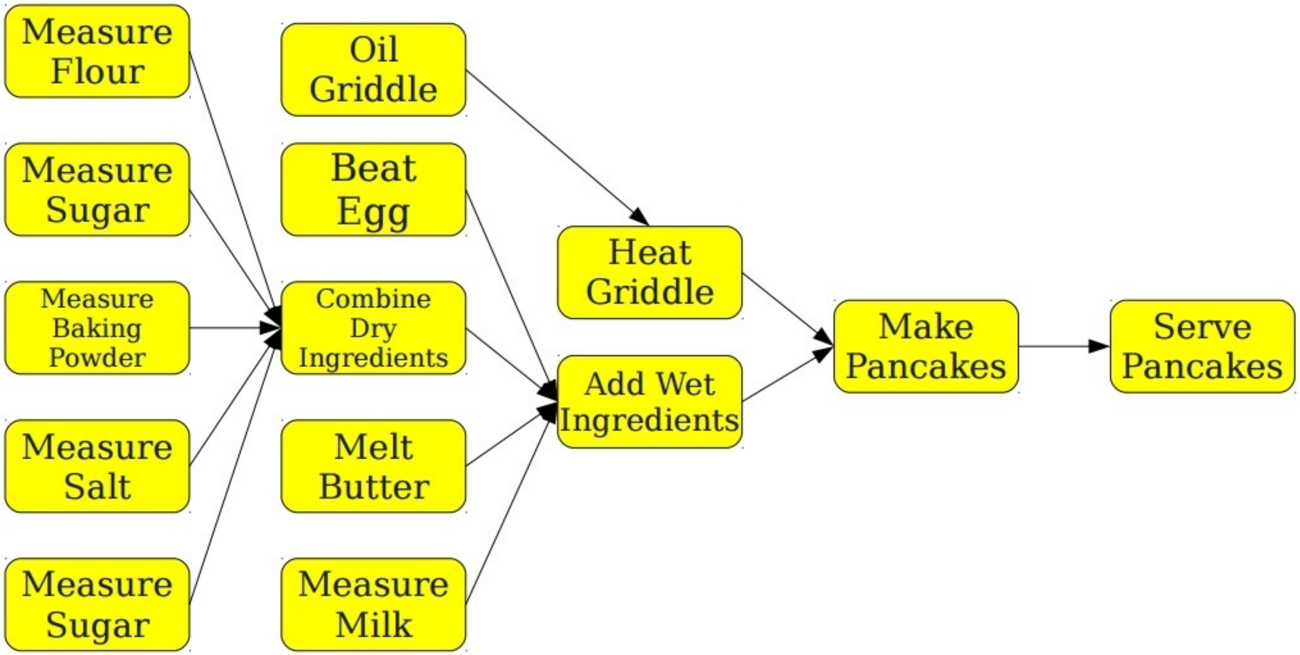
- Directed acyclic graph (dag)
  - A directed graph with no cycles.
- A topological sort of a dag is a linear ordering of all its vertices such that if G contains an edge (u,v), then u appears before v in the ordering.
- An ordering of its vertices along a horizontal line so that all directed edges go from left to right.
- This is a different kind of sort than we have done in the past.

Dag of course prerequisites



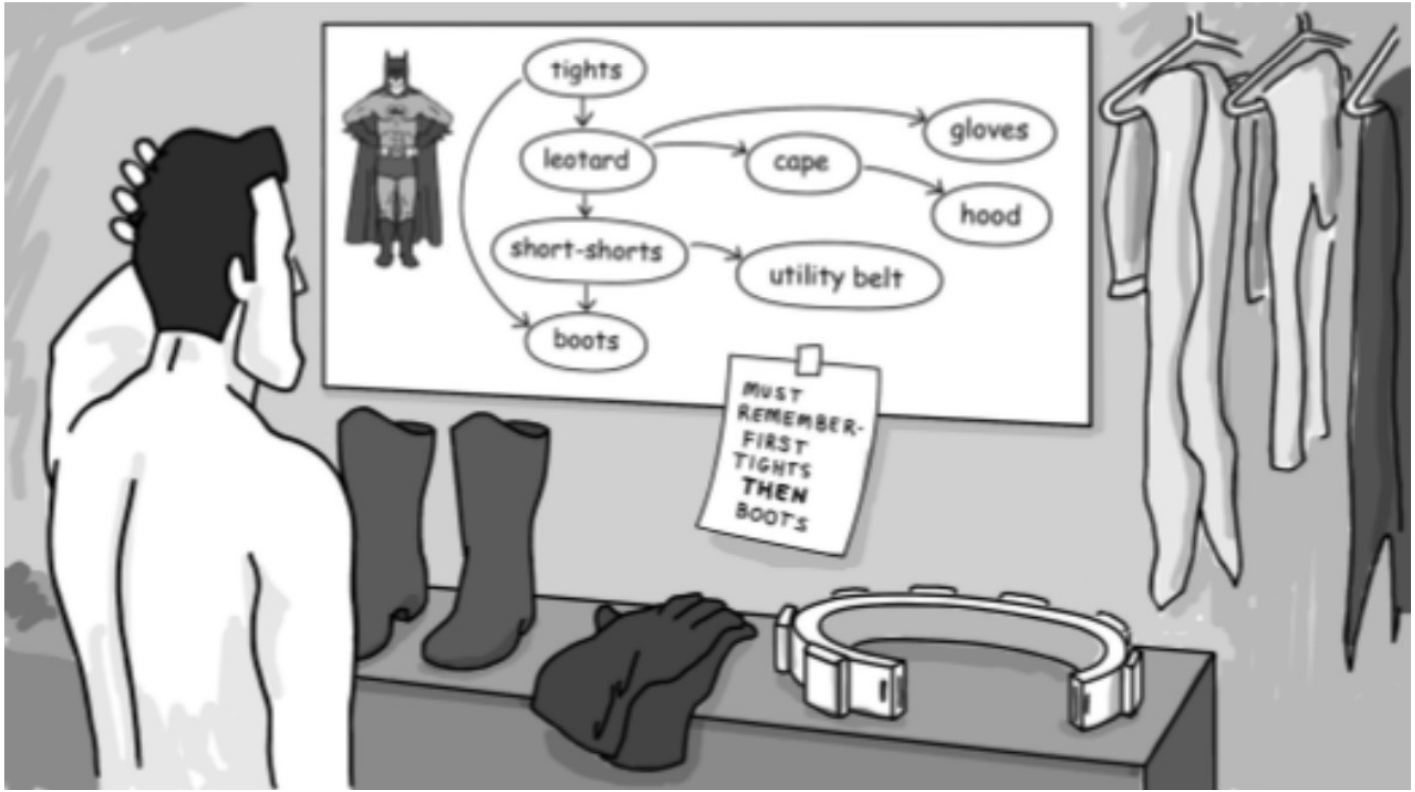
Dags are used in scheduling, when one thing must happen before another
https://www.cs.northwestern.edu/academics/courses/311/html/graphs.html

Dag of making pancakes

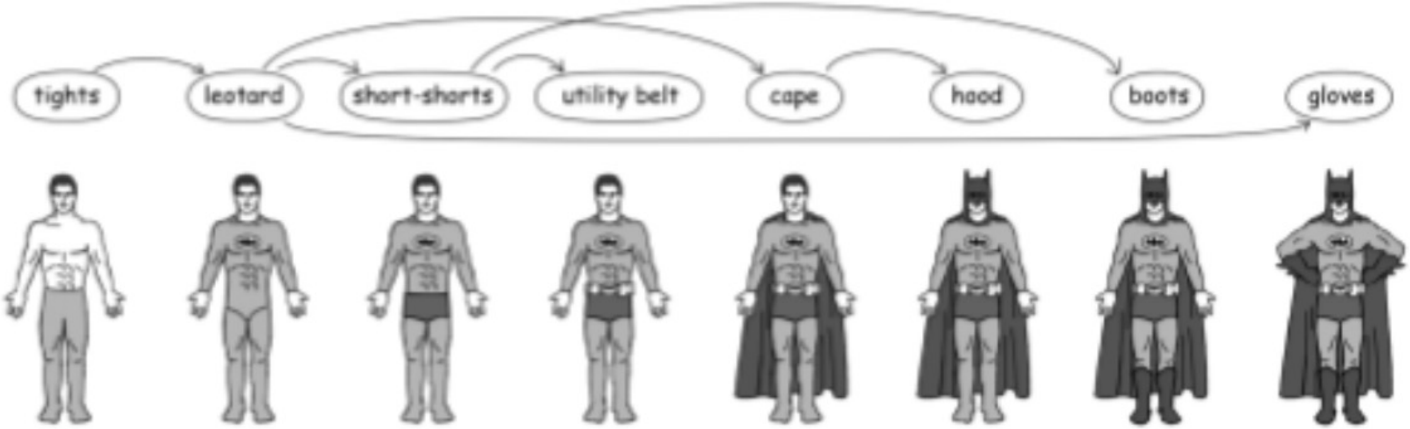


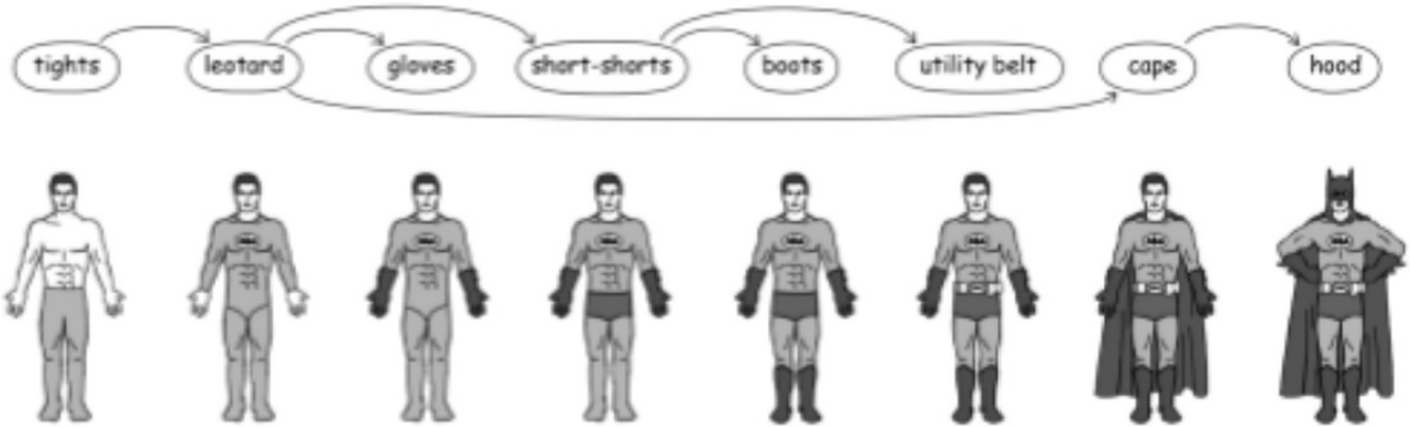
Dags can model dependencies

Topological Sort

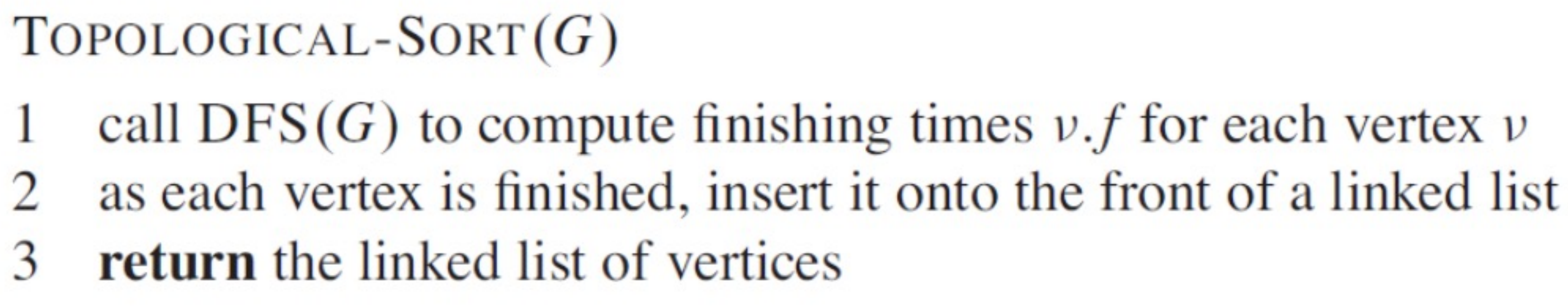


A directed edge (u,v) indicates that item u must be put on before item v


tights
leotard
short-shorts
utility belt
cape
hood
boots
gloves


tights
leotard
gloves
short-shorts
boots
utility belt
cape
hood

Topological Sort



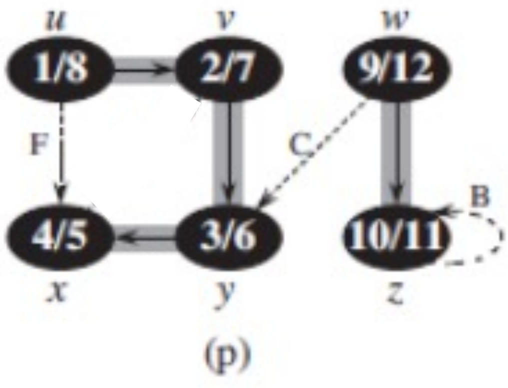
```
TOPOLOGICAL-SORT($G$)
call DFS($G$) to compute finishing times $v.f$ for each vertex $v$
as each vertex is finished, insert it onto the front of a linked list
**return** the linked list of vertices
```

- What is the time complexity?

Using DFS to detect a dag

- A directed graph G is acyclic if and only if a depth-first search of G yields no back edges

Topological Sort



(p)