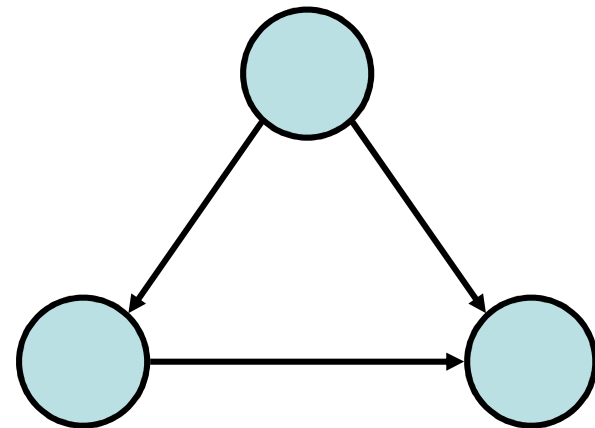
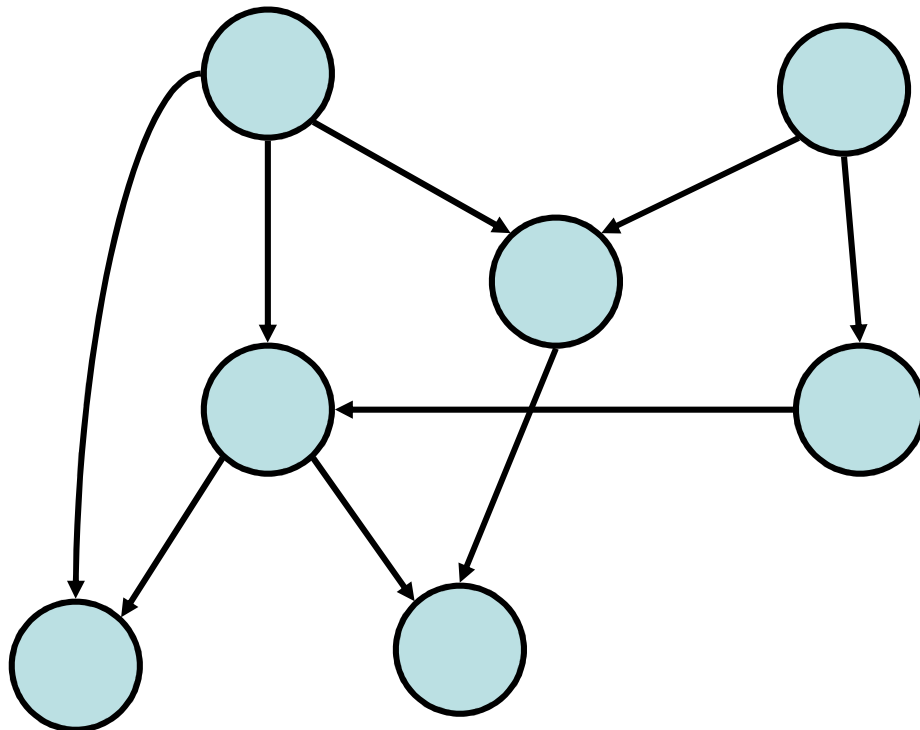


# DFS And Graph Cycles

- Thm: An undirected graph is *acyclic* iff a DFS yields no back edges
  - If acyclic, no back edges (because a back edge implies a cycle)
  - If no back edges, acyclic
    - No back edges implies only tree edges (*Why?*)
    - Only tree edges implies we have a tree or a forest
    - Which by definition is acyclic
- Thus, can run DFS to find whether a graph has a cycle

# Directed Acyclic Graphs

- A *directed acyclic graph* or *DAG* is a directed graph with no directed cycles:



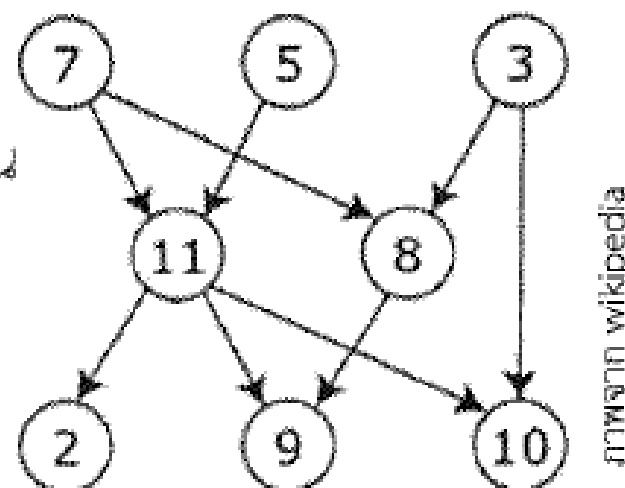
# DFS and DAGs

- A directed graph  $G$  is acyclic iff a DFS of  $G$  yields no back edges:
  - if  $G$  is acyclic, will be no back edges
    - Trivial: a back edge implies a cycle
  - if no back edges,  $G$  is acyclic
    - Argue contra positive:  $G$  has a cycle  $\Rightarrow \exists$  a back edge

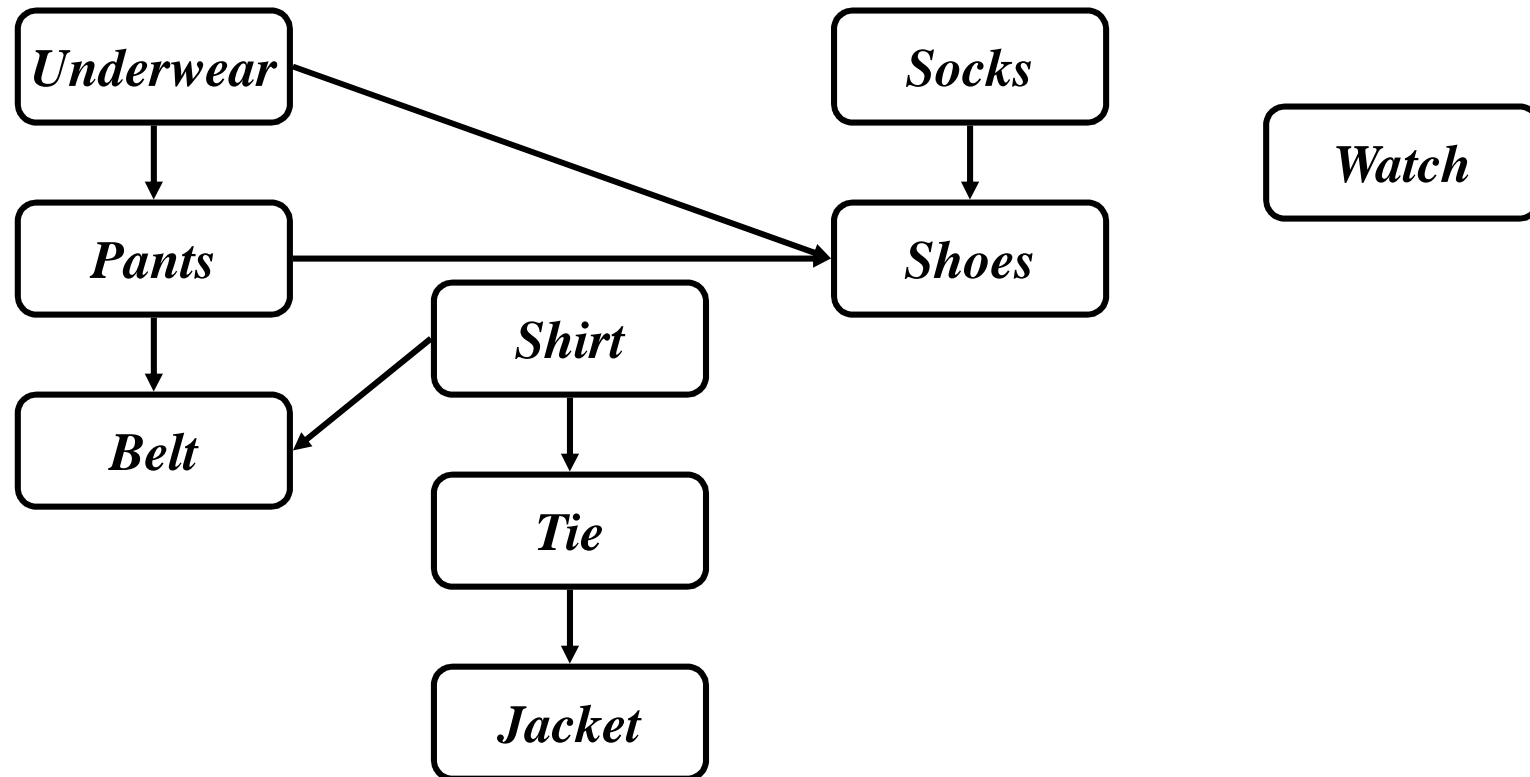


## Topological Sort (toposort)

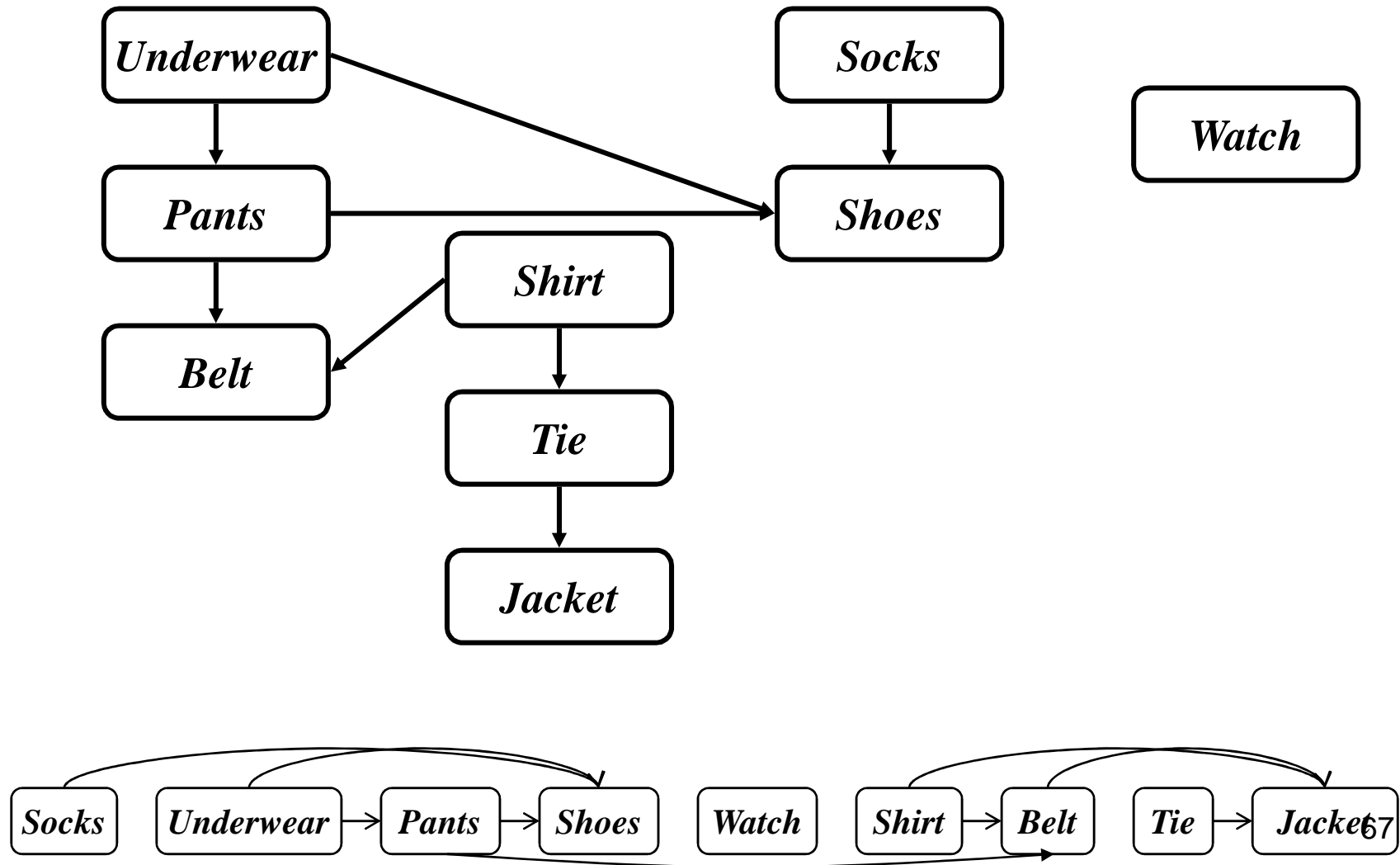
- เป็นการจัดลำดับโหนดในกราฟที่มีทิศทางและไม่มีรูป (Directed Acyclic Graph / DAG)
- มักใช้กับการจัดแผนงาน (Job Scheduling) ที่ขั้นตอนของงานมีทั้งส่วนที่ขึ้นต่อกันและเป็นอิสระต่อกัน
- ขั้นตอนงานที่ขึ้นต่องานอื่นหมายความว่า จะทำขั้นตอนนั้นได้ถ้างานอื่นที่ว่าทำเสร็จไปก่อนแล้ว
- จากภาพ ขั้นตอนงาน 9 จะทำได้ก็ต่อเมื่อขั้นตอนงาน 8 และ 11 ทำเสร็จแล้ว ในทำนองเดียวกัน ขั้นตอนงาน 11 จะทำได้ก็ต่อเมื่อขั้นตอนงาน 5 และ 7 ถูกเสร็จไปก่อน



# Getting Dressed



# Getting Dressed



# Topological Sort Algorithm

```
Topological-Sort()
```

```
{
```

```
    Run DFS
```

```
    When a vertex is finished, output it
```

```
    Vertices are output in reverse  
    topological order
```

```
}
```

- Time:  $O(V+E)$
- Correctness: Want to prove that
$$(u,v) \in G \Rightarrow u \rightarrow f > v \rightarrow f$$

# Problems

- **10305 - Ordering Tasks**
- <http://uva.onlinejudge.org/external/103/10305.html>