Lecture 8 – Directed Graphs

Reading: KT Section 3.4 and 3.6

Partial content of these slides have been obtained from the official lecture slides that accompany the textbook. A complete set of slides can be found at: http://www.cs.princeton.edu/~wayne/kleinberg-tardos/

Directed Graphs

What's different about them?
Direction is important

Connectivity in directed graphs

**Directed reachability.** Given a node $s$, find all nodes reachable from $s$.

**Directed $s$-$t$ shortest path problem.** Given two nodes $s$ and $t$, what is the length of a shortest path from $s$ to $t$?

**Graph search.** BFS extends naturally to directed graphs.

**Web crawler.** Start from web page $s$. Find all web pages linked from $s$, either directly or indirectly.
Assuming you began at node a, give the order of a BFS traversal of the graph is,

**Undirected:**

**Directed:**
Directed Acyclic Graphs

Def. A DAG is a directed graph that contains no directed cycles.

Def. A topological order of a directed graph $G = (V, E)$ is an ordering of its nodes as $v_1, v_2, \ldots, v_n$ so that for every edge $(v_i, v_j)$ we have $i < j$.

Dependency Graph on a DAG

- Usually reflect dependencies or requirements
  - i.e., Assembly lines, Supply lines, Organizational charts, ...
  - BTW: You cannot take 231 after 230 unless...

- Understanding dependencies requires “topological sorting”
Topological Sorting Algorithm

- Select a vertex $v$ that has no predecessor
- Remove $v$ from the graph (along with all associated arcs),
- Add $v$ to the end of a list of vertices $L$
- Repeat previous steps
- When the graph is empty, $L$'s vertices will be in topological order

Why does this work?

If $G$ is DAG, then $G$ has a topological ordering.

If $G$ is a DAG, then $G$ has a node with no predecessors.