COMP 282
Advanced Data Structures

Lecture 23
Topological Sorting
Topological Order

• Directed Graphs without cycles have a natural order
Motivation

• Useful for certain ordinary life situations:
  – If the vertices represent academic courses, and
  – The edges represent required prerequisites
  – Then the topological sorted order of the graph presents an order in which the courses must be taken to satisfy all prerequisites.
What Does it mean?

• If you arrange all the nodes of a graph in a straight line according to the topological sorted order, then

• All of the directed edges will point in the same direction.

• Arranging vertices in this fashion is known as “topological sorting”.

• Because all the edge must point in the same direction it is not possible to sort graphs that contain a cycle.
A simple algorithm

- `topSort(theGraph)`
  
  
  \[
  n = \text{number of vertices in theGraph}
  \]
  
  for step = 1 through n) {
    select a vertex v that has no successors
    aList.addFirst(v)
    delete from theGraph v and all edges to it.
  }
  
  return aList
Preserving the Graph

• How could we do it without destroying the graph?
• By adding markings to the nodes again…

```java
void topSort(Graph theGraph) {
    n = number of vertices in theGraph
    for step = 1 through n) {
        select a vertex v that has no unmarked successors
        aList.addFirst(v)
        mark v
    }
    return aList
}
```