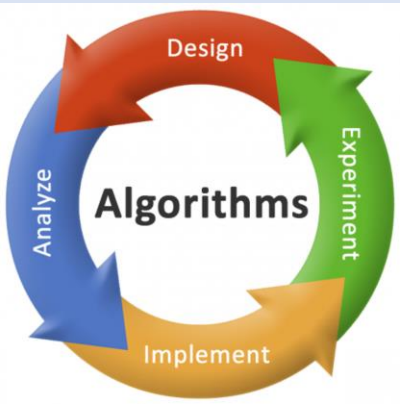
Graph Algorithms
Topological Sort

COP 3503
Fall 2021
Department of Computer Science
University of Central Florida
Dr. Steinberg



What is Topological Sort?

- A topological sort is a sequence of items in which one item precedes the others for certain reasons.
- Example – Courses and Prerequisites

Courses	Prerequisites
COMPSCI 100	MATH 120
COMPSCI 150	MATH 140
COMPSCI 200	COMPSCI 100, COMPSCI 150, ENG 100
COMPSCI 240	COMPSCI 200, PHYS 130
ENG 110	NONE
MATH 120	NONE
MATH 130	MATH 120
MATH 140	MATH 130
MATH 200	MATH 140, PHYS 130
PHYS 130	NONE

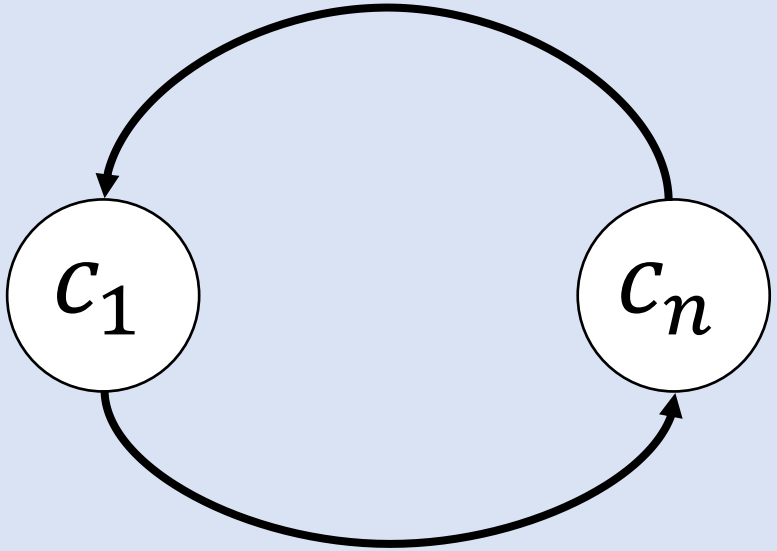
Example

- MATH 120, MATH 130, COMPSCI 100, MATH 140, COMPSCI 150, ENG 110, COMPSCI 200, PHY 130, MATH 200, COMPSCI 240

Courses	Prerequisites
COMPSCI 100	MATH 120
COMPSCI 150	MATH 140
COMPSCI 200	COMPSCI 100, COMPSCI 150, ENG 100
COMPSCI 240	COMPSCI 200, PHYS 130
ENG 110	NONE
MATH 120	NONE
MATH 130	MATH 120
MATH 140	MATH 130
MATH 200	MATH 140, PHYS 130
PHYS 130	NONE

Topological Sorting

- For a topological sort to exist, there must not be any cyclical prerequisite structure. NO DIRECTED CYCLES!!
- Example
  - c_1 is a prerequisite for c_n, and c_n is a prerequisite for c_1



Topological Sorting and Graphs

- Topological sorting is useful in graphs if we want to observe how vertices relate to each other.
- Input is a directed acyclic graph (dag)
- Output is topological storing of V in dag G(V,E)

Topological Sorting Algorithm

```
top_sort(adj,ts)
n = adj.last
k = n

for i = 1 to n
        visit[i] = false

for i = 1 to n
        if (!visit[i])
                top_sort_recurs(adj, v, ts)
```

```
top_sort_recurs(adj, start, ts)
visit[start] = true
trav = adj[start]

while(trav != null)
        v = trav.ver
        if(!visit[v])
                top_sort_recurs(adj,v,ts)
```

Example