

In the name of God

Network Flows

2. Search Algorithms

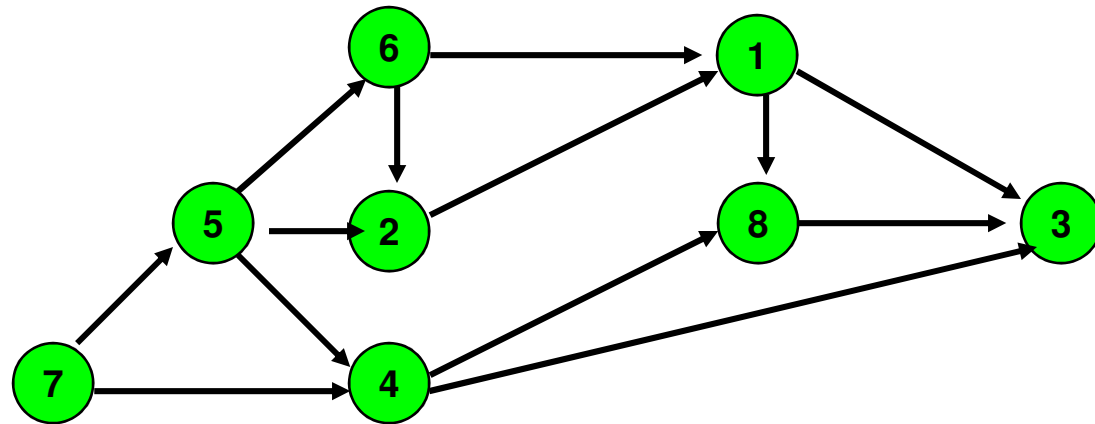
2.4 Topological Sorting – An Example

Fall 2010

Instructor: Dr. Masoud Yaghini

Initialization

“Next” will be the label of nodes in the topological order.

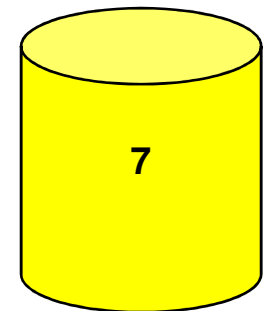


next

0

Node
Indegree

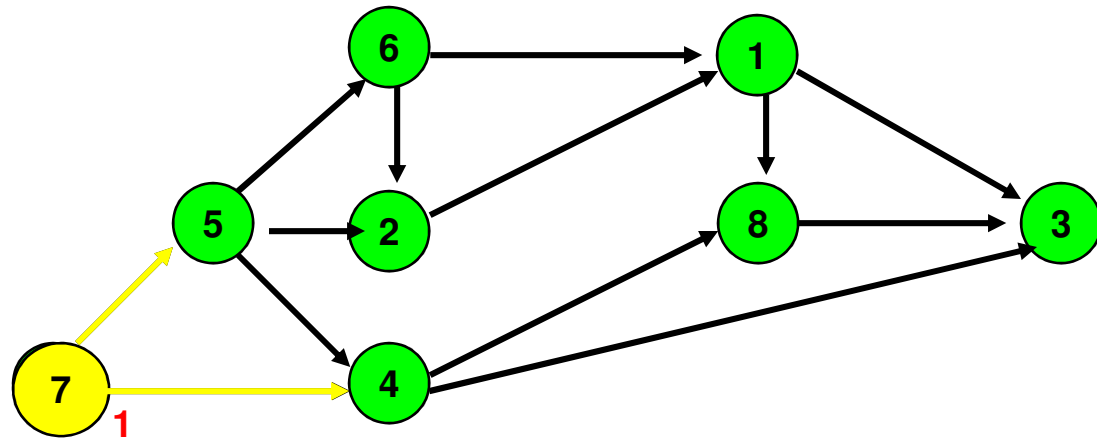
1	2	3	4	5	6	7	8
2	2	3	2	1	1	0	2



LIST

Select a node from LIST

next := next + 1
 order(i) := next;
 update indegrees
 update LIST

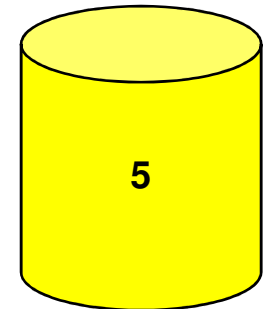


next

1

Node
 Indegree

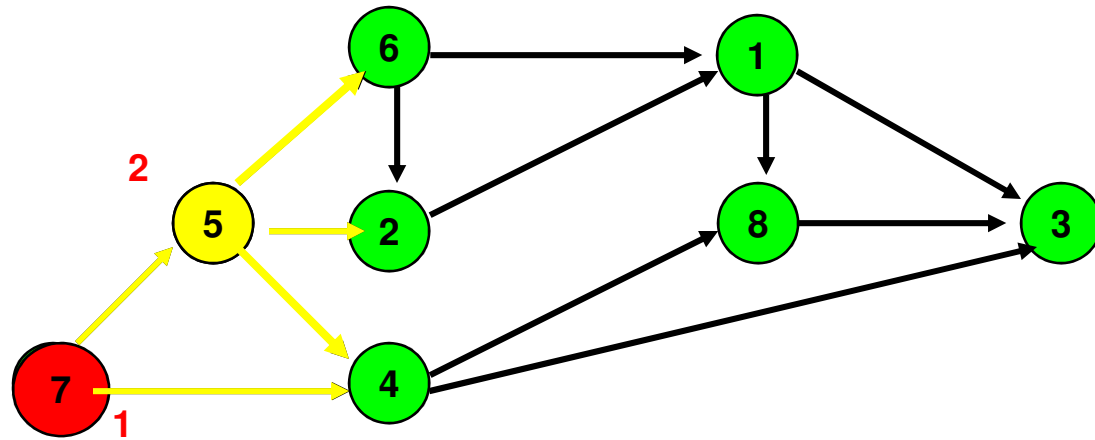
1	2	3	4	5	6	8
2	2	3	1	0	1	2



LIST

Select a node from LIST

next := next + 1
 order(i) := next;
 update indegrees
 update LIST

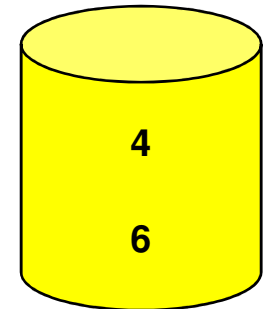


next

2

Node
 Indegree

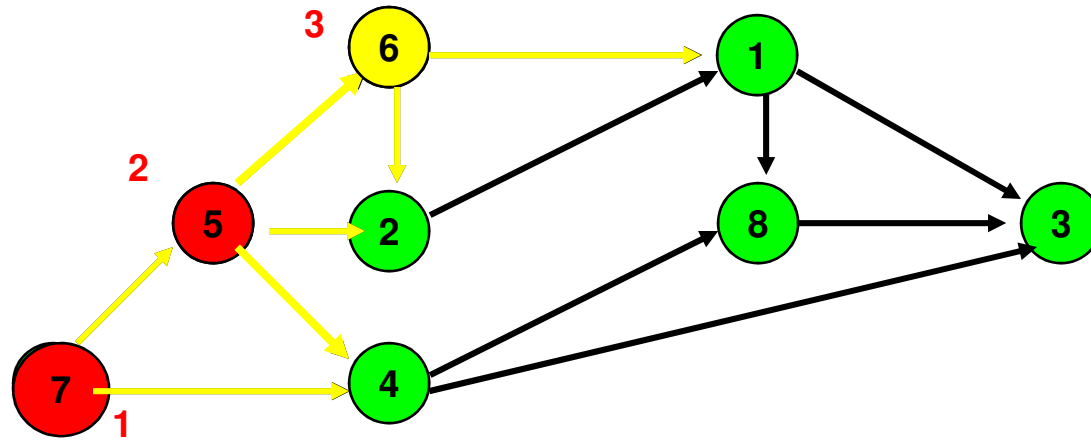
1	2	3	4	6	8
2	1	3	0	0	2



LIST

Select a node from LIST

next := next + 1
 order(i) := next;
 update indegrees
 update LIST



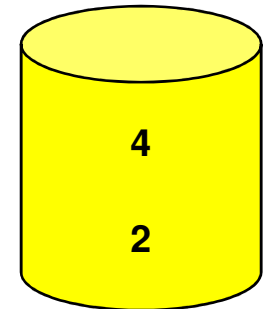
next

3

Node
 Indegree

1	2	3	4
1	0	3	0

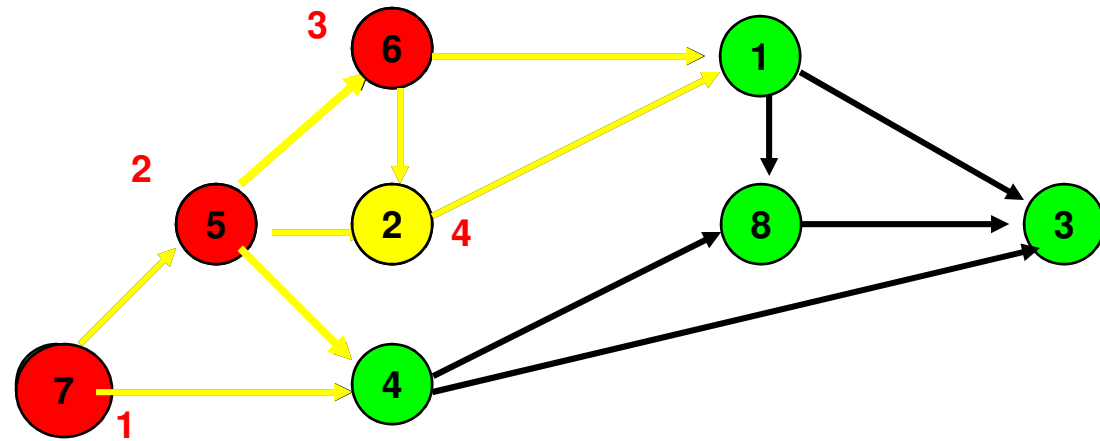
8
2



LIST

Select a node from LIST

next := next + 1
 order(i) := next;
 update indegrees
 update LIST



next

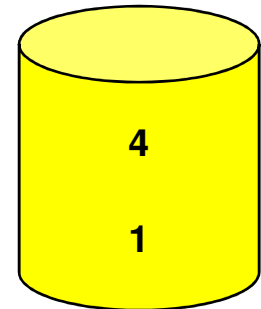
4

Node
 Indegree

1
0

3	4
3	0

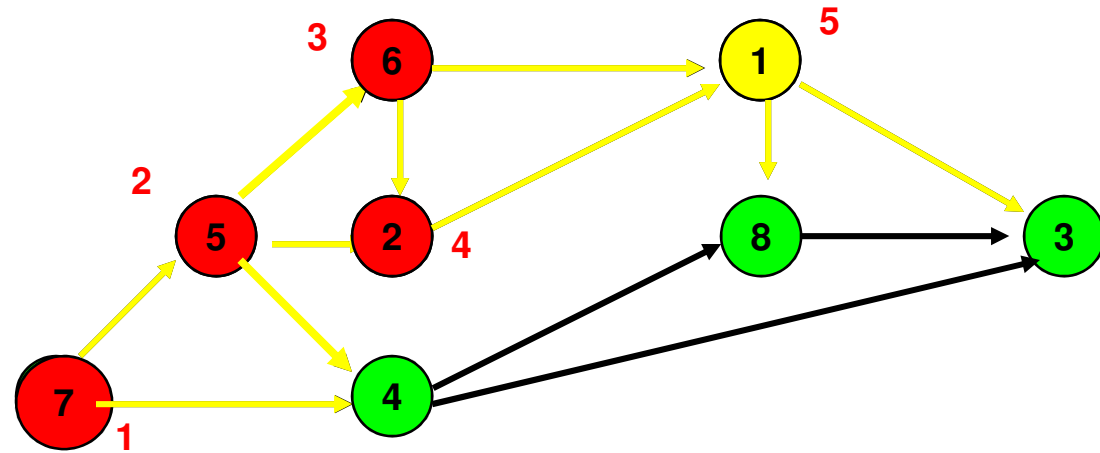
8
2



LIST

Select a node from LIST

next := next + 1
 order(i) := next;
 update indegrees
 update LIST



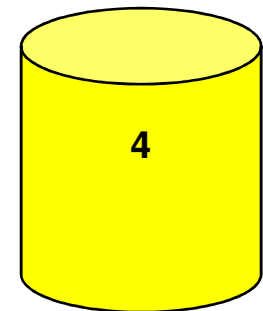
next

5

Node
 Indegree

3	4
2	0

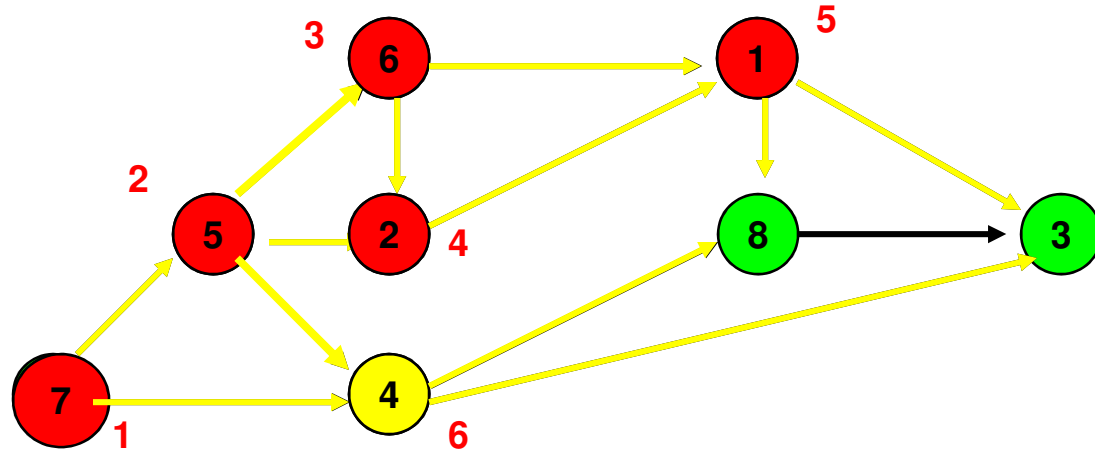
8
1



LIST

Select a node from LIST

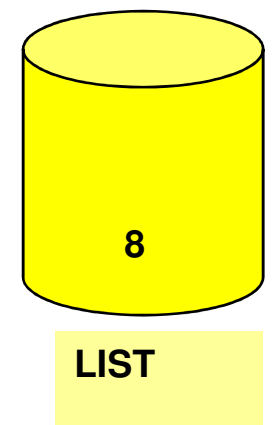
`next := next + 1`
`order(i) := next;`
 update indegrees
 update LIST



next 6

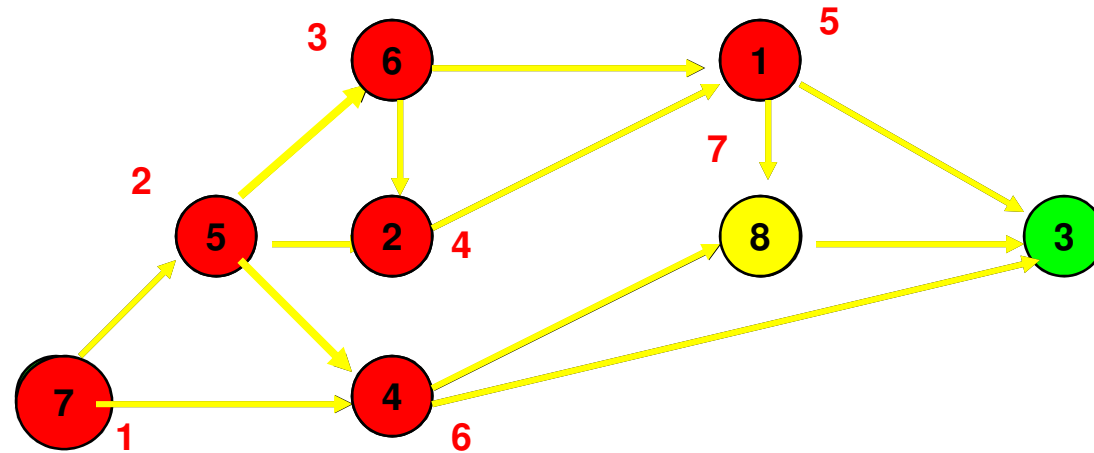
Node	3
Indegree	1

Node	8
Indegree	0



Select a node from LIST

next := next + 1
order(i) := next;
update indegrees
update LIST



next

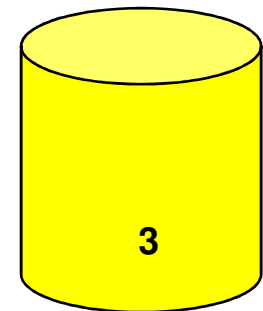
7

Node

3

Indegree

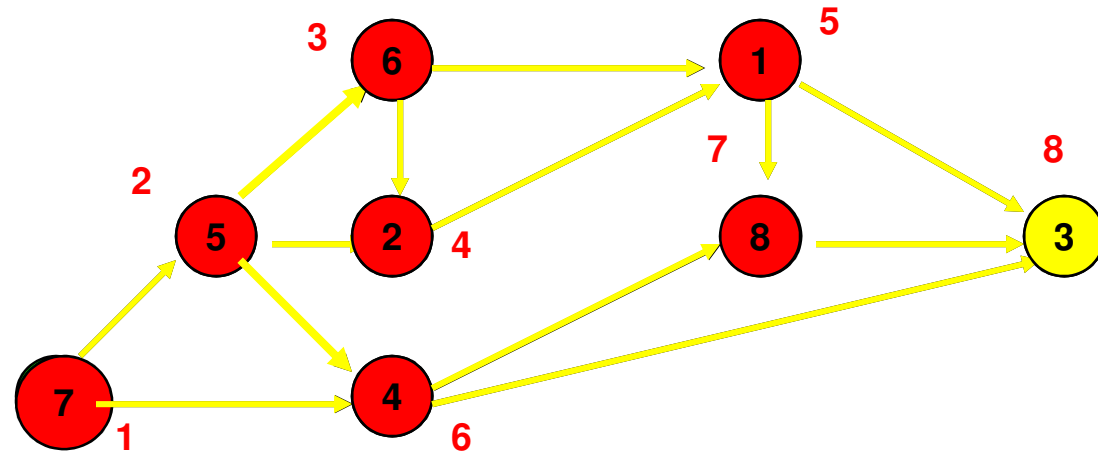
0



LIST

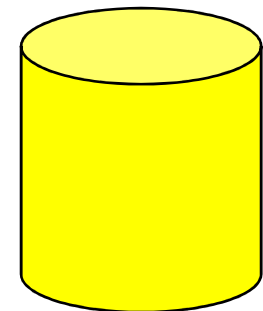
Select a node from LIST

```
next := next + 1  
order(i) := next;  
update indegrees  
update LIST
```



next

8



LIST

List is empty.

The algorithm terminates with a topological order of the nodes

Node

Indegree



The End