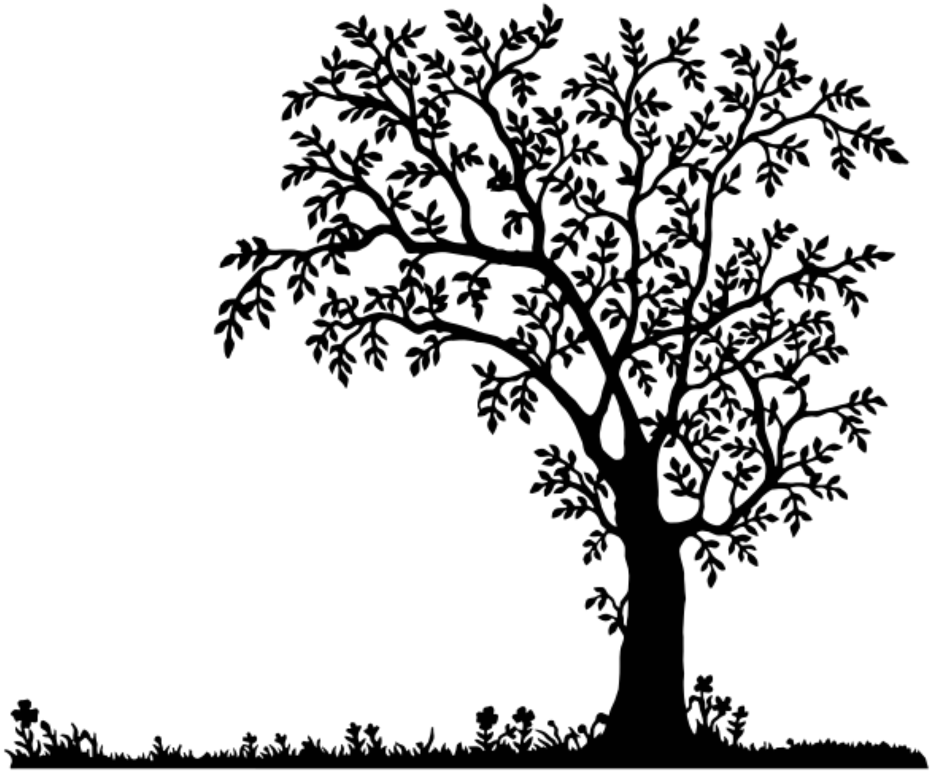
Data Structures and Algorithms (DSA) for AI

Fernanda Madeiral



Applications — BFS and DFS

BFS:

Finding the shortest path (next lecture)

Finding all neighboring locations

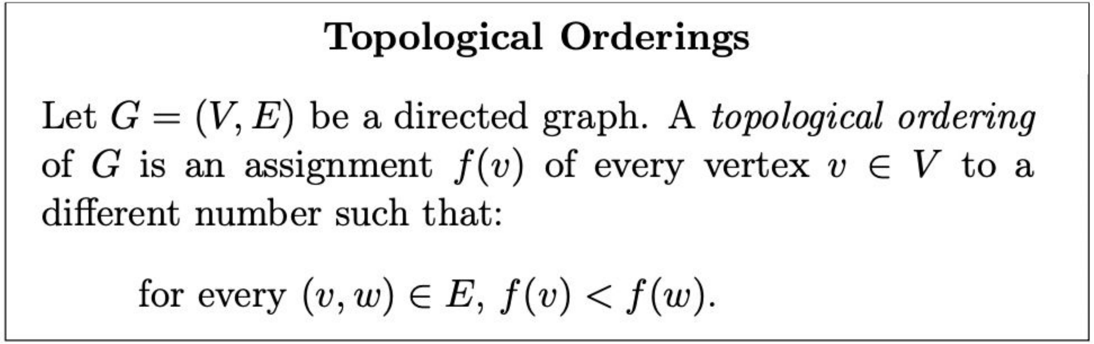
Broadcasting tasks, such as with networks

DFS:

Solving puzzles, such as mazes

Topological ordering (this lecture)

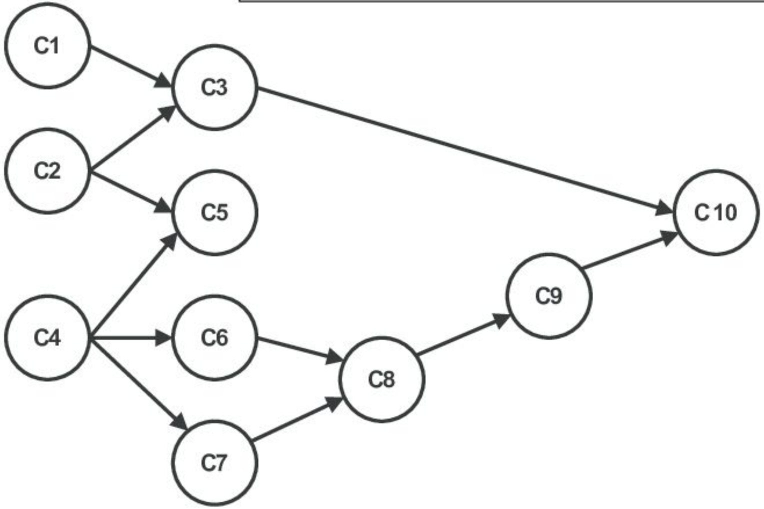
Topological Ordering



Topological Orderings

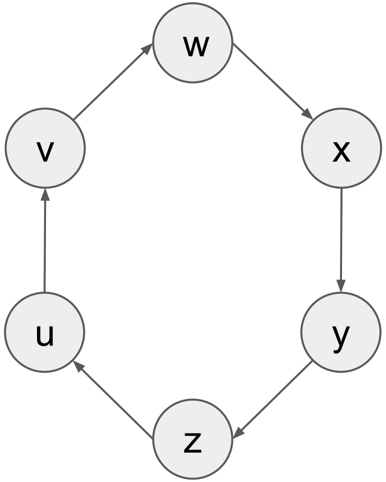
Let $G = (V, E)$ be a directed graph. A *topological ordering* of G is an assignment $f(v)$ of every vertex $v \in V$ to a different number such that:

$$\text{for every } (v, w) \in E,\ f(v) < f(w).$$

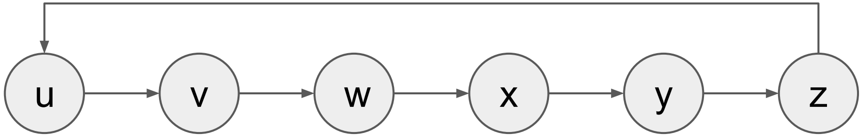


Example:
Problem of
pre-requirements
for courses

Example of a graph with no topological ordering

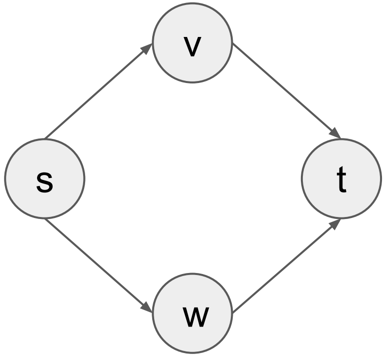


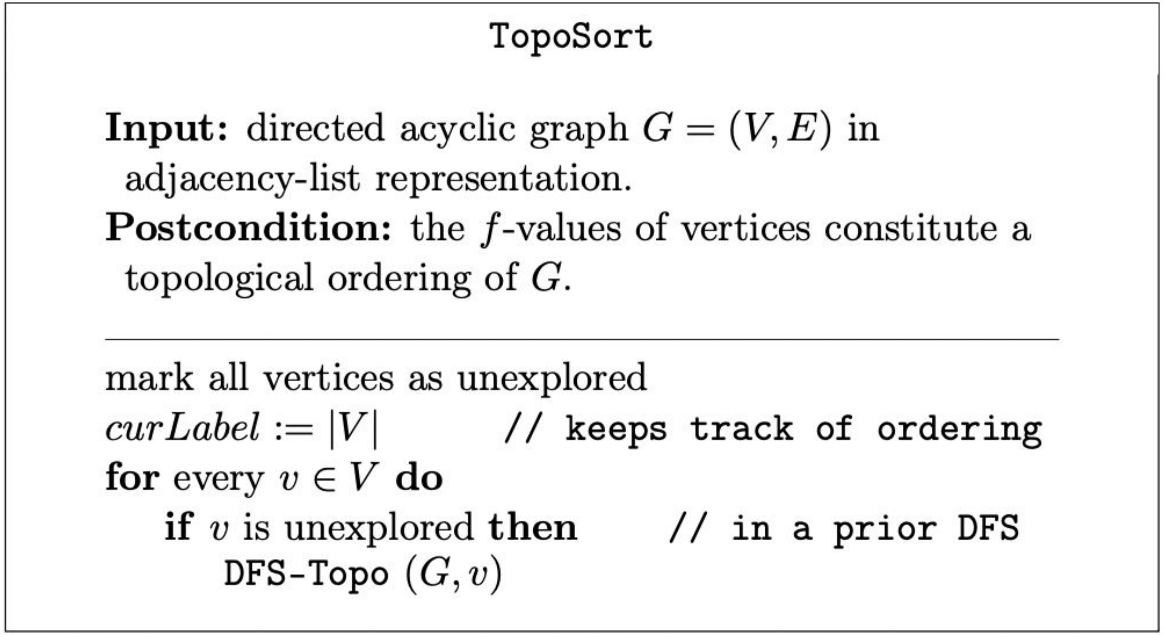
A graph with a directed cycle



A non-topological ordering

Topological orderings





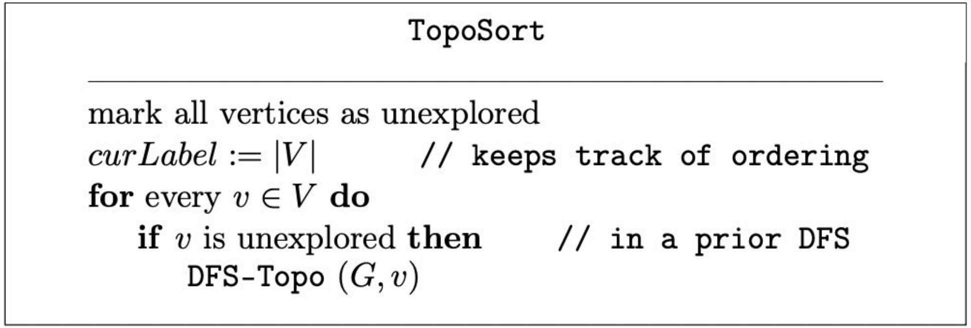
TopoSort

Input: directed acyclic graph $G = (V, E)$ in adjacency-list representation.

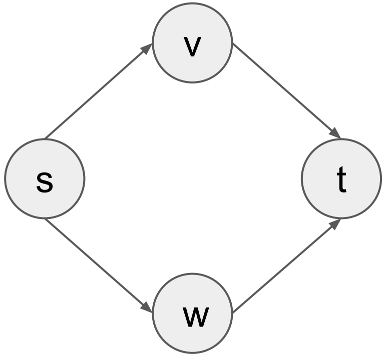
Postcondition: the f-values of vertices constitute a topological ordering of G.

```
mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

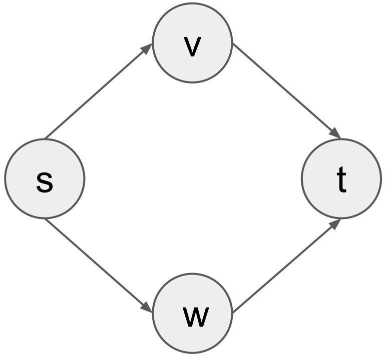
Topological orderings

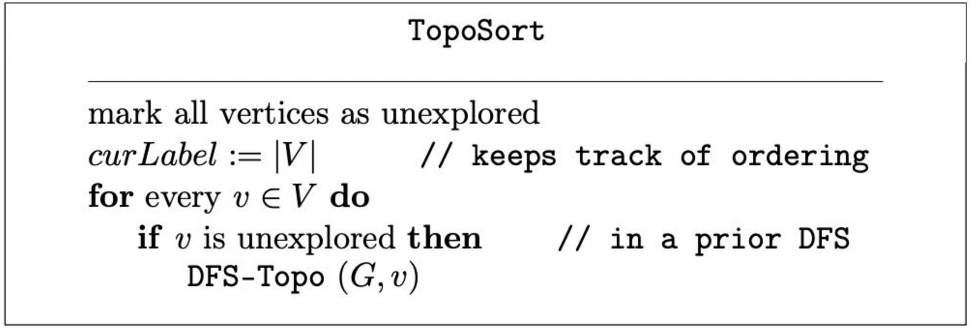


```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then     // in a prior DFS
        DFS-Topo $(G, v)$
```



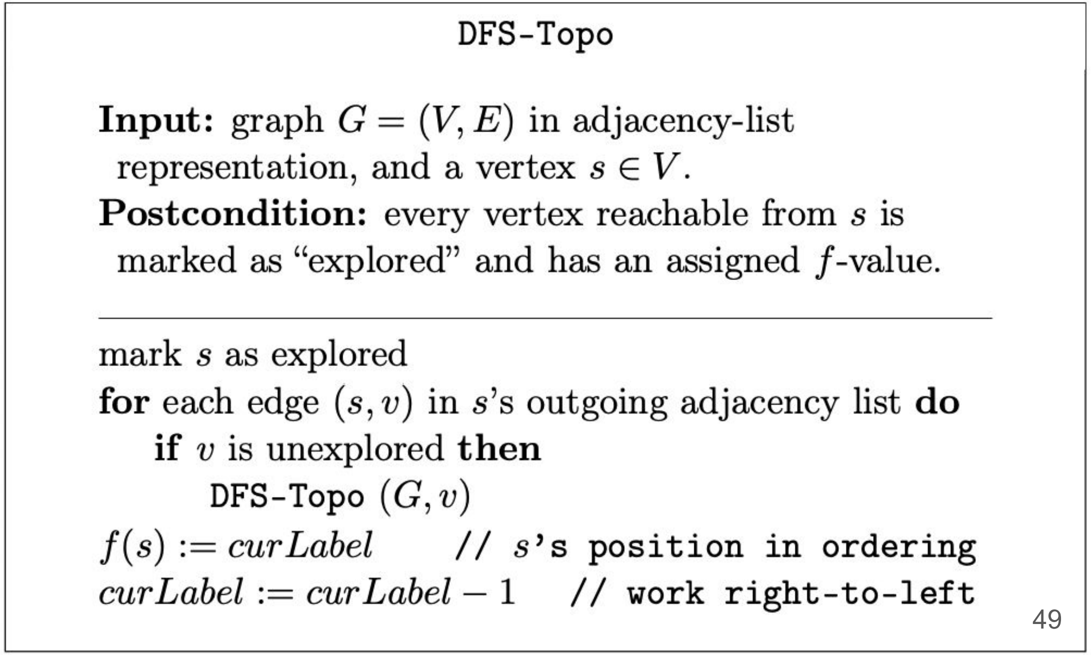
Topological orderings





TopoSort

```
mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

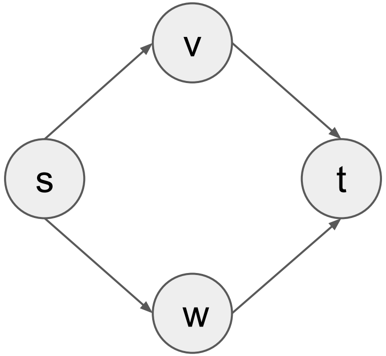


DFS-Topo

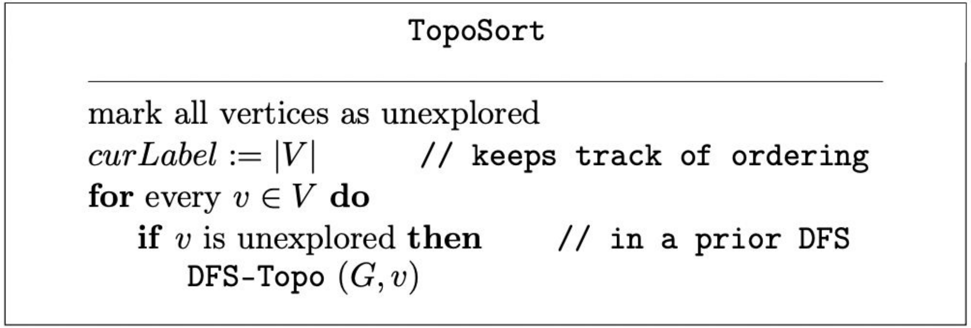
Input: graph $G = (V, E)$ in adjacency-list representation, and a vertex $s \in V$.

Postcondition: every vertex reachable from s is marked as "explored" and has an assigned f-value.

```
mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```

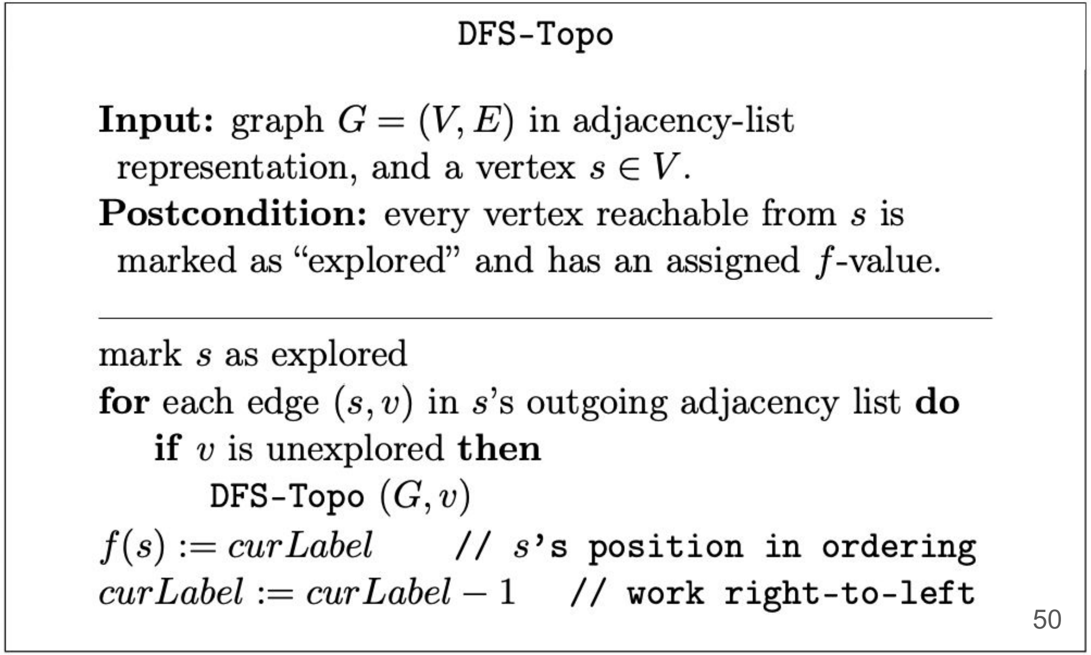


curLabel = 4
f(v) = ?
f(t) = ?
f(s) = ?
f(w) = ?



TopoSort

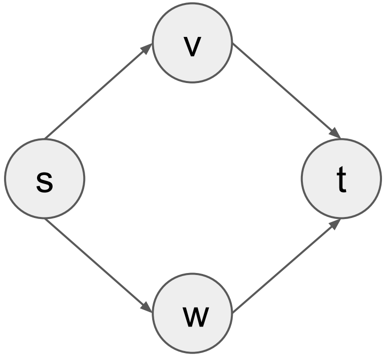
```
mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```



DFS-Topo

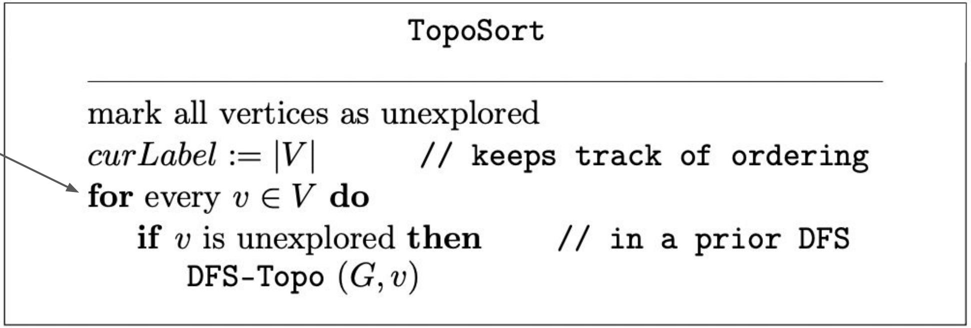
Input: graph $G = (V, E)$ in adjacency-list representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from s is marked as "explored" and has an assigned f-value.

```
mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // s's position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```



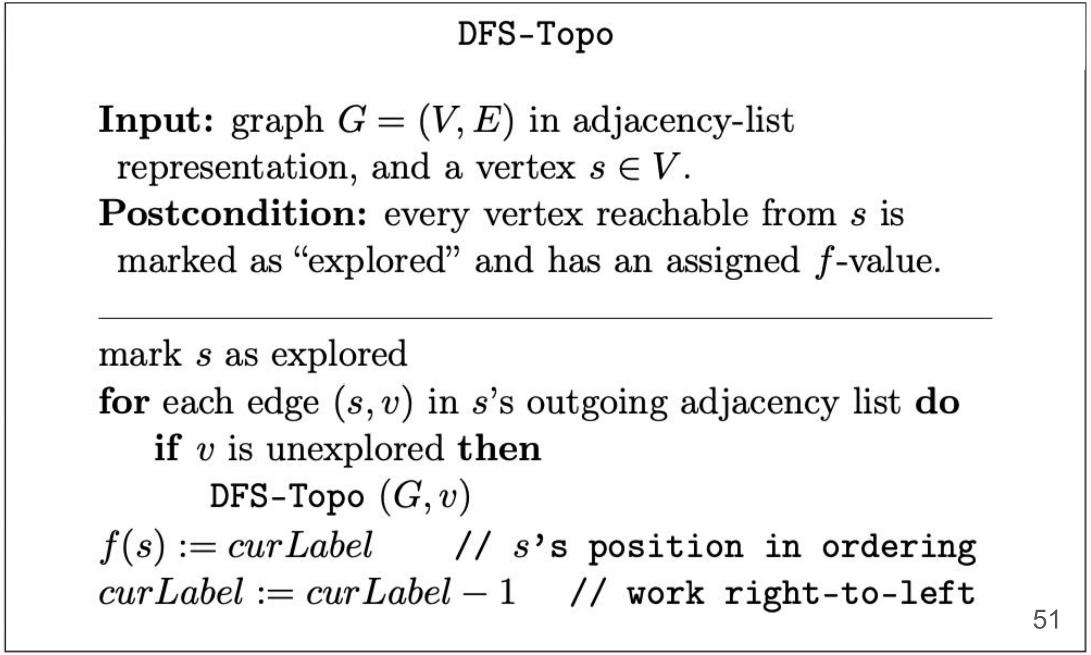
curLabel = 4
f(v) = ?
f(t) = ?
f(s) = ?
f(w) = ?

Let's assume the order "v, t, s, w"



TopoSort

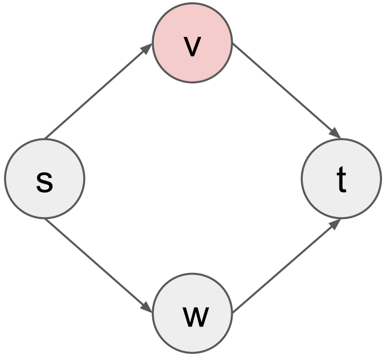
```
mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then        // in a prior DFS
        DFS-Topo $(G, v)$
```



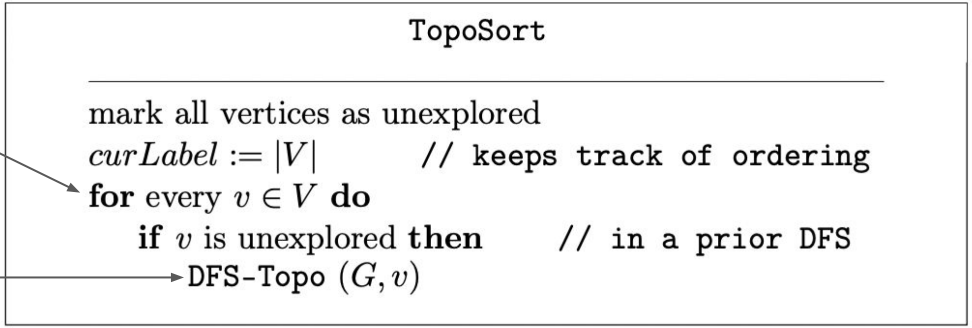
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from s is marked as "explored" and has an assigned f-value.

```
mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```



Let's assume the order "**v**, t, s, w"



```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

"v"

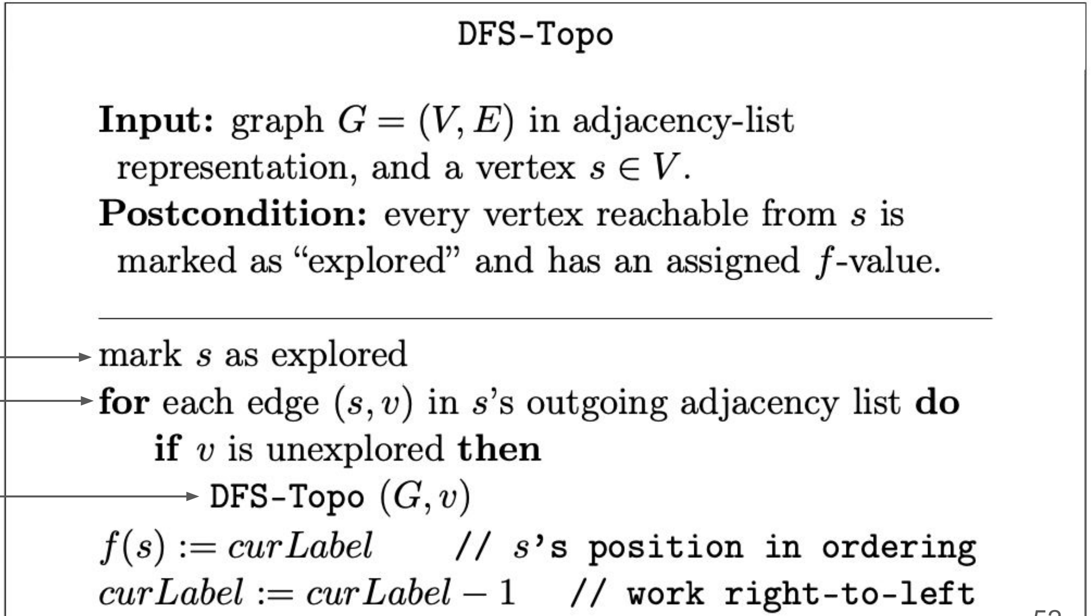
curLabel = 4

f(v) = ?

f(t) = ?

f(s) = ?

f(w) = ?

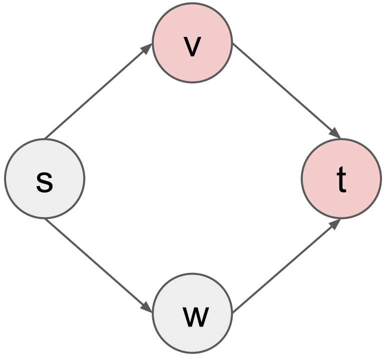


```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```

"v"

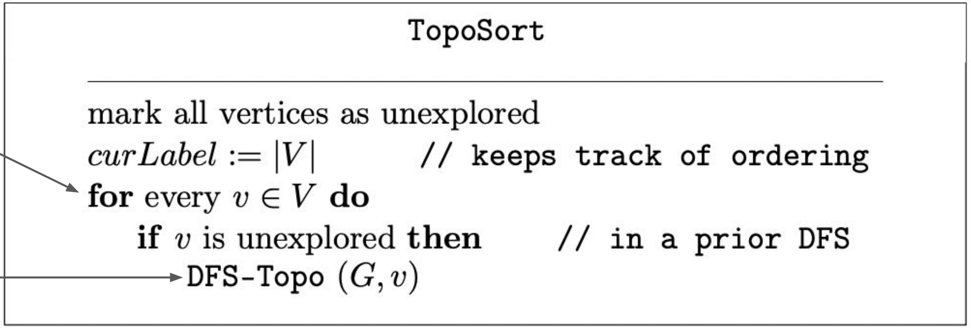
Only (v, t)

Call DFS-Topo on "t"



curLabel = 4
f(v) = ?
f(t) = ?
f(s) = ?
f(w) = ?

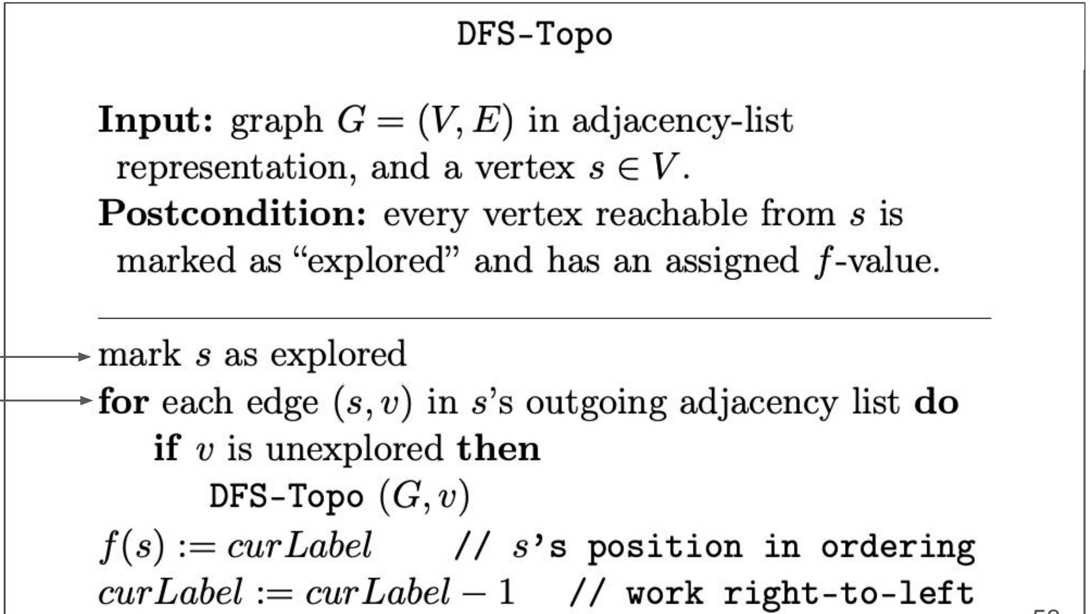
Let's assume the order "**v**, t, s, w"



TopoSort

```
mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

"v"

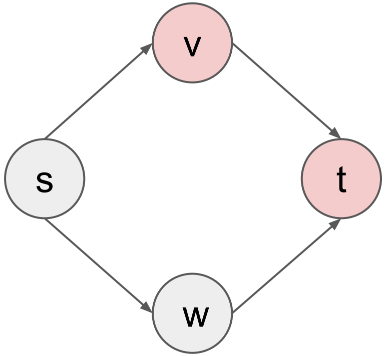


DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from s is marked as “explored” and has an assigned f-value.

```
mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // s's position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```

"t"
No edge



curLabel = ~~4~~ 3
f(v) = ?
f(t) = 4
f(s) = ?
f(w) = ?

Let's assume the order "**v**, t, s, w"

"v"

```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

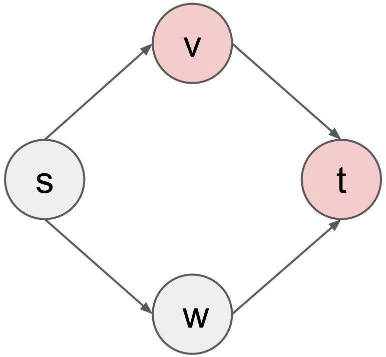
"t"

No edge

f(t) = 4

curLabel = 3

```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$          // s's position in ordering
$curLabel := curLabel - 1$      // work right-to-left
```



curLabel = ~~4~~ 3
f(v) = ?
f(t) = 4
f(s) = ?
f(w) = ?

Let's assume the order "**v**, t, s, w"

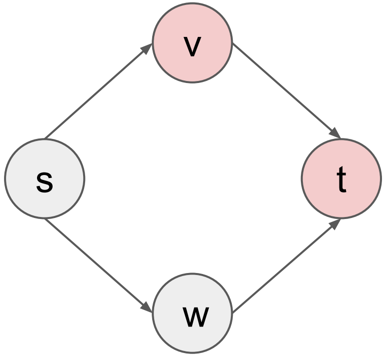
"v"

```
TopoSort
mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

"v"

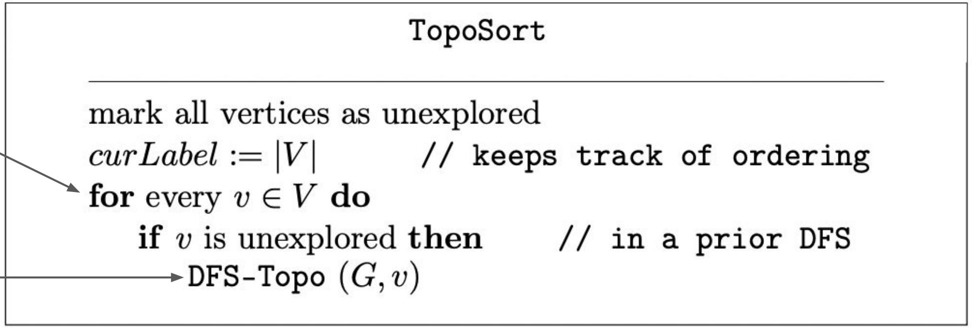
No other edge

```
DFS-Topo
Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$      // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```



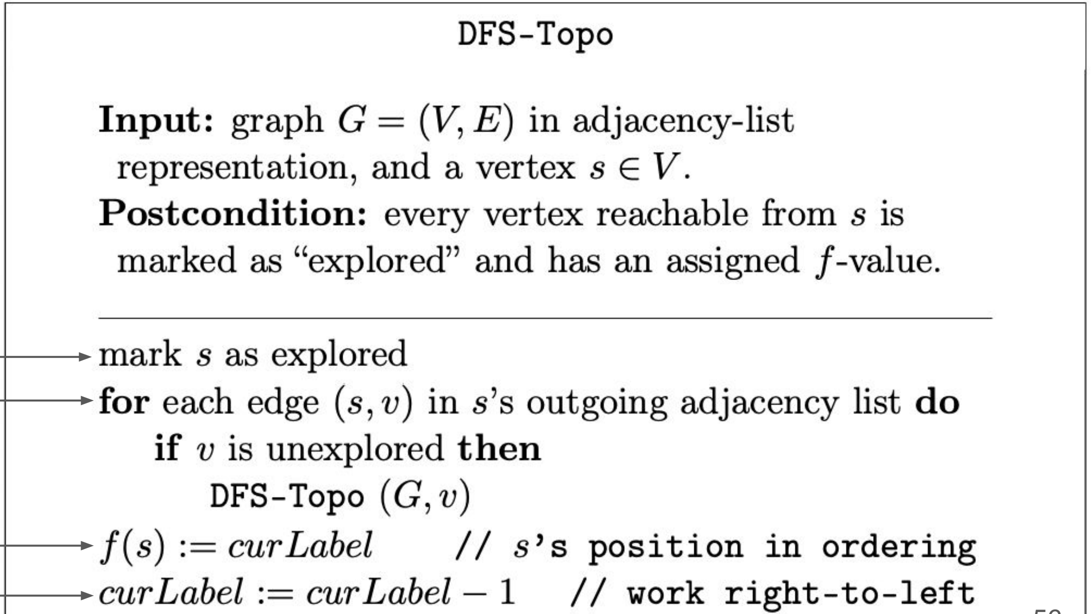
Let's assume the order "**v**, t, s, w"

"v"



```
TopoSort
mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then        // in a prior DFS
        DFS-Topo $(G, v)$
```

curLabel = ~~4~~ ~~3~~ 2
f(v) = 3
f(t) = 4
f(s) = ?
f(w) = ?



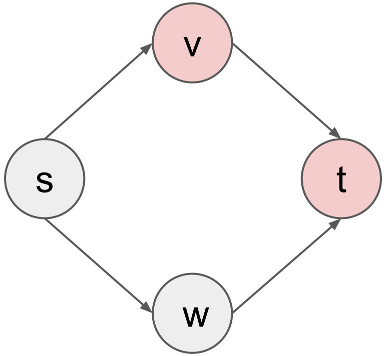
```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // s's position in ordering
$curLabel := curLabel - 1$        // work right-to-left
```

"v"

No other edge

f(v) = 3

curLabel = 2



curLabel = ~~4~~ ~~3~~ 2

f(v) = 3

f(t) = 4

f(s) = ?

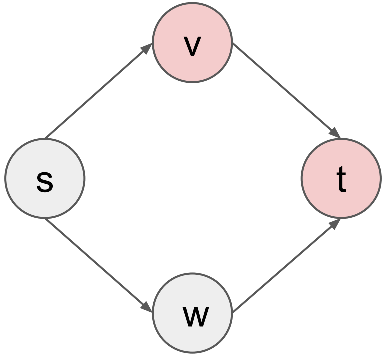
f(w) = ?

Let's assume the order "v, **t**, s, w"

"t"

```
TopoSort
-----------------------------------------------------------
mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.
-----------------------------------------------------------
mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$          // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```



curLabel = ~~4~~ ~~3~~ 2

f(v) = 3

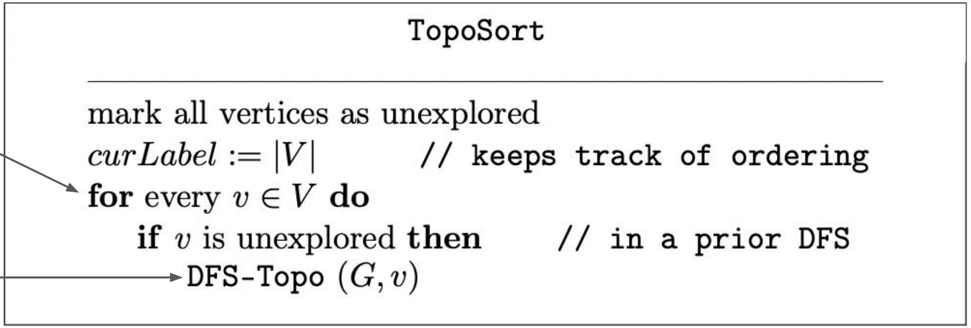
f(t) = 4

f(s) = ?

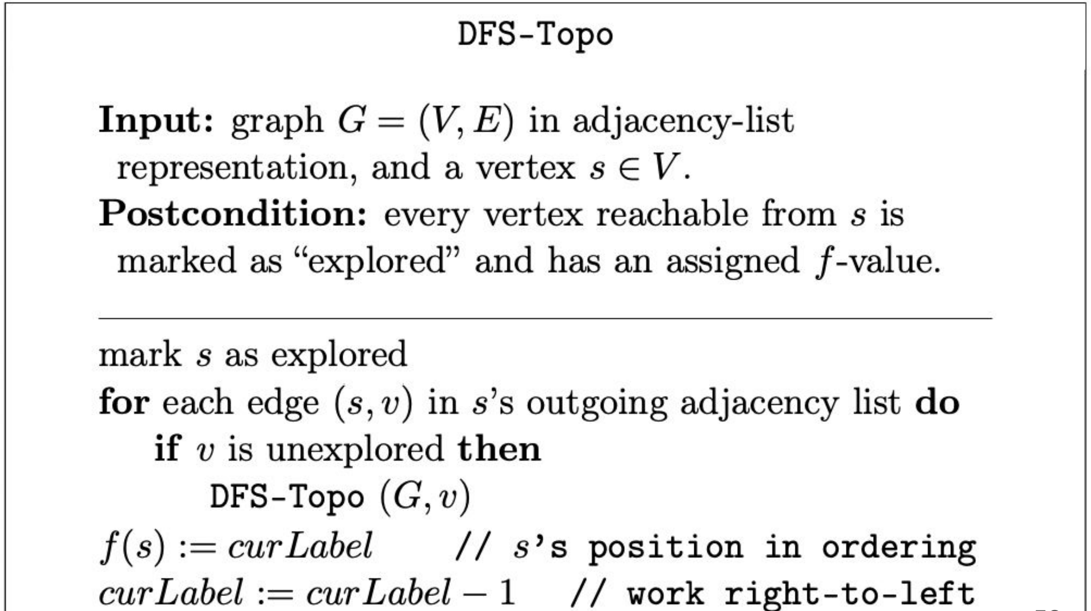
f(w) = ?

Let's assume the order "v, t, **s**, w"

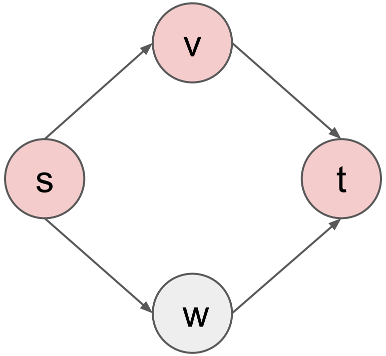
"s"



```
TopoSort
mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```



```
DFS-Topo
Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$       // s's position in ordering
$curLabel := curLabel - 1$     // work right-to-left
```



curLabel = ~~4~~ ~~3~~ 2

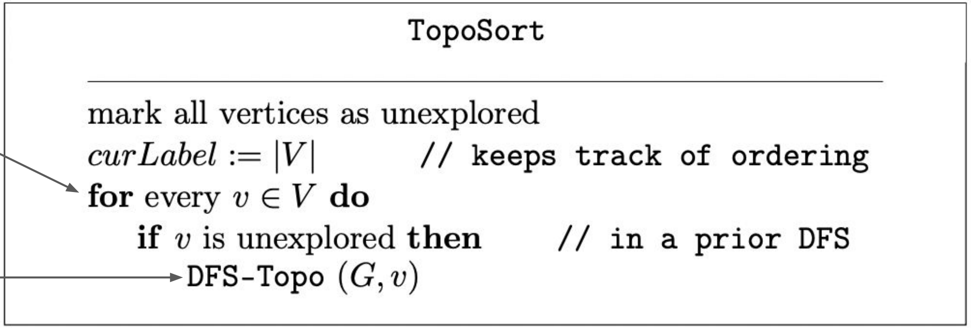
f(v) = 3

f(t) = 4

f(s) = ?

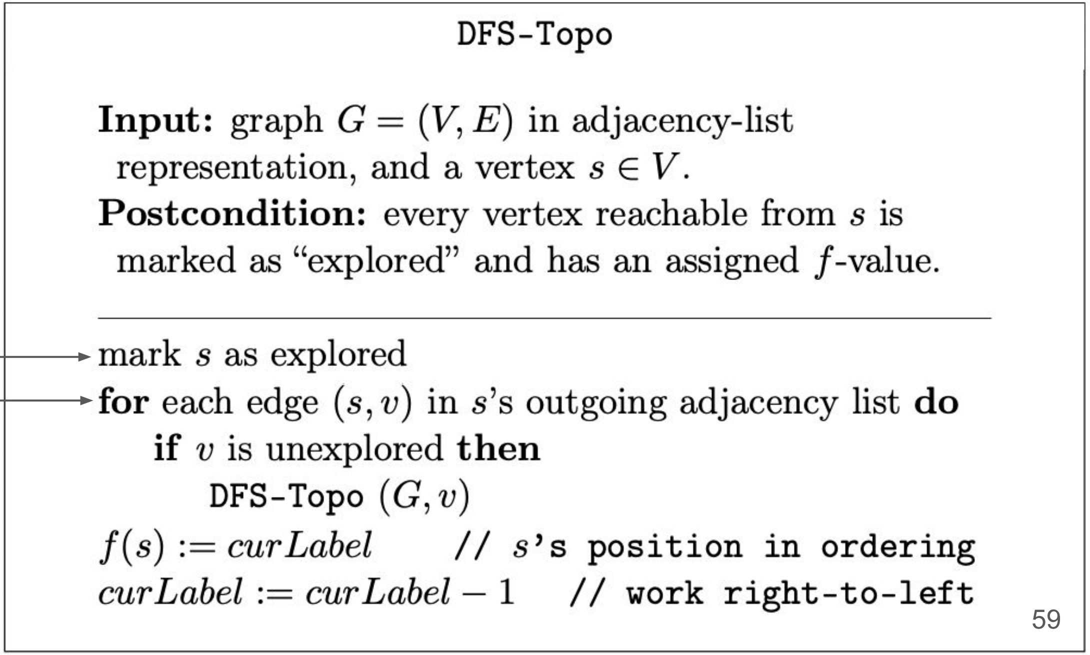
f(w) = ?

Let's assume the order "v, t, **s**, w"



```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

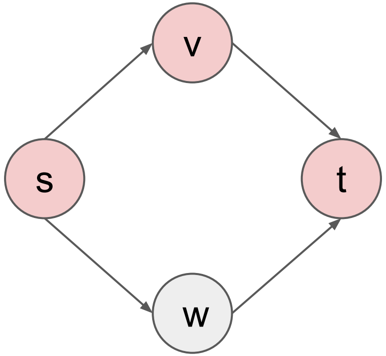
"s"



```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$      // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```

"s"

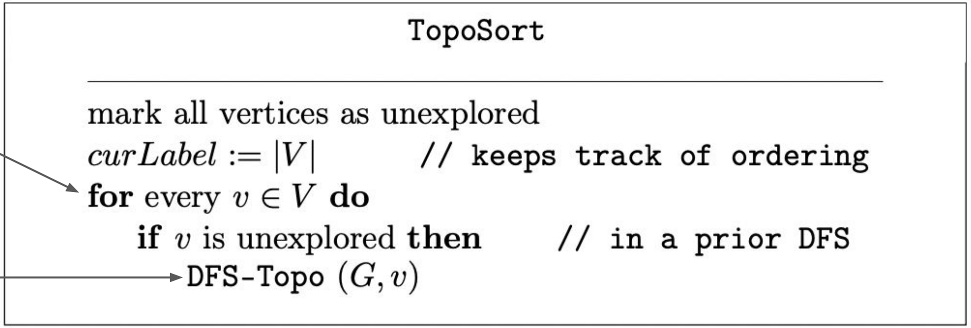
(s, v), (s, w)



curLabel = ~~4~~ ~~3~~ 2
f(v) = 3
f(t) = 4
f(s) = ?
f(w) = ?

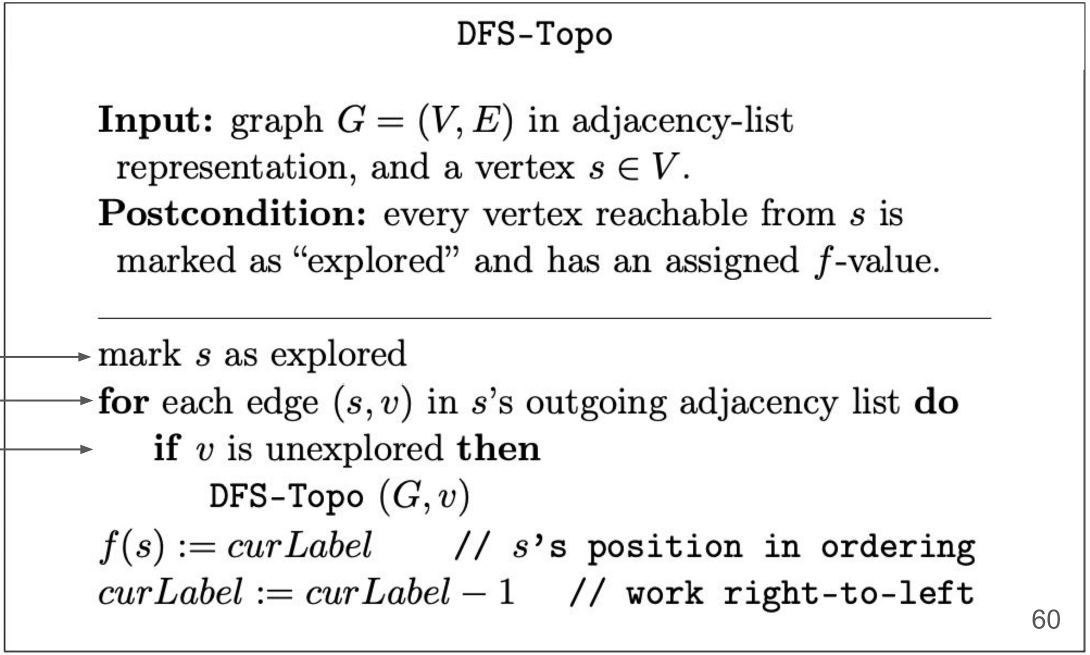
Let's assume the order "v, t, **s**, w"

"s"



```
TopoSort
mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

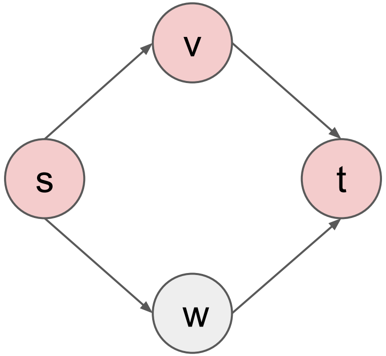
"s"
(s, v), (s, w)
"v" already explored



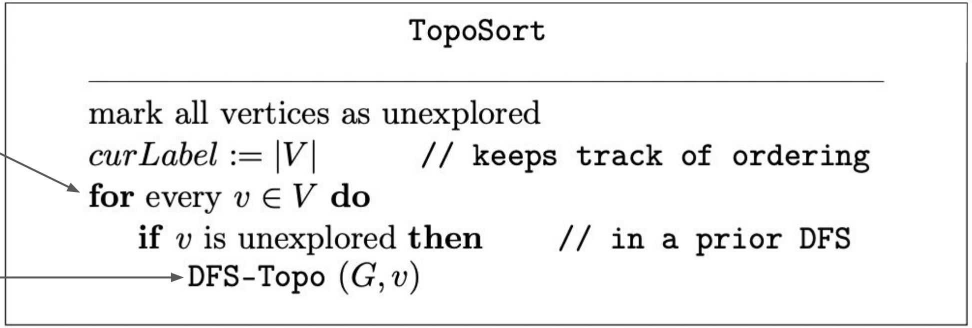
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from s is marked as "explored" and has an assigned f-value.

```
mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // s's position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```



Let's assume the order "v, t, **s**, w"



```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

"s"

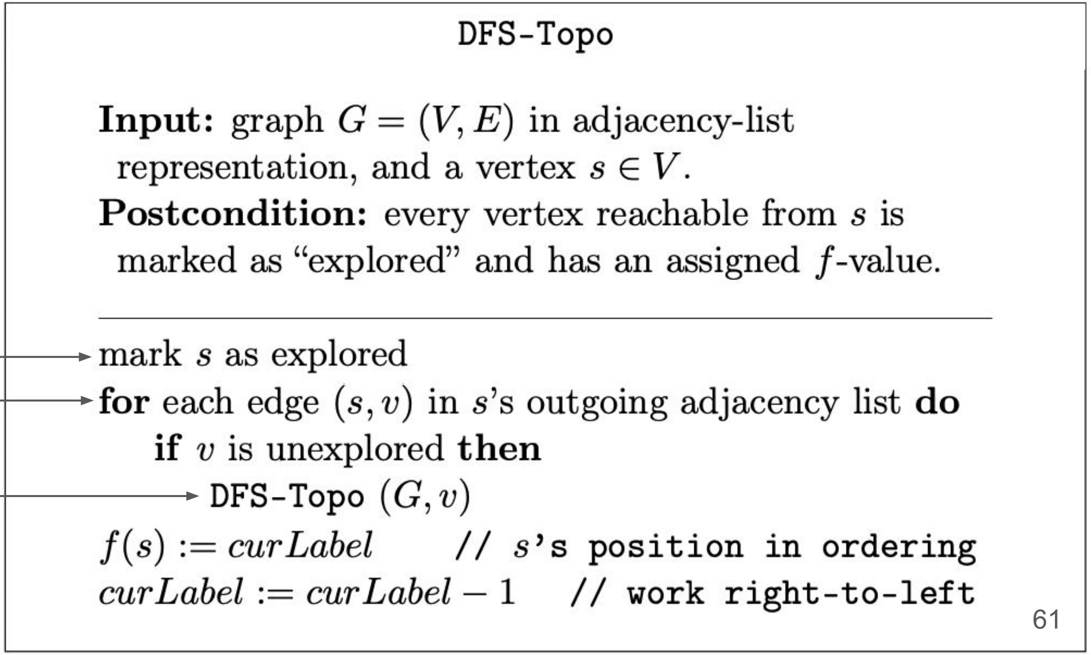
curLabel = ~~4~~ ~~3~~ 2

f(v) = 3

f(t) = 4

f(s) = ?

f(w) = ?

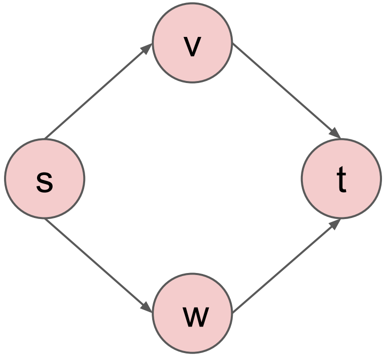


```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
  representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
  marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // s's position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```

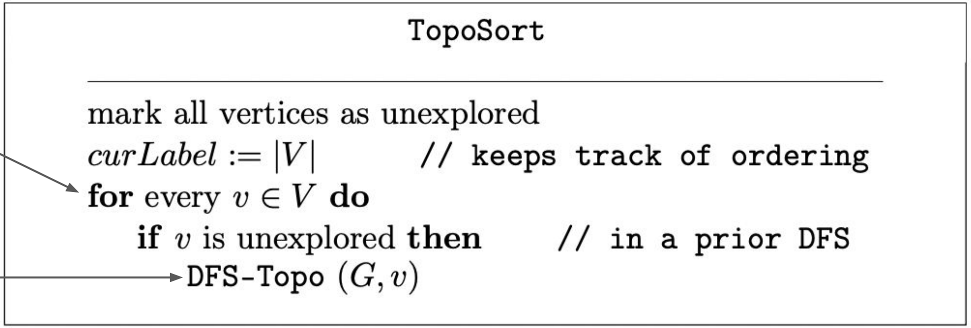
"s"

(s, v), **(s, w)**

Call DFS-Topo on "w"



Let's assume the order "v, t, **s**, w"



```
TopoSort
mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

"s"

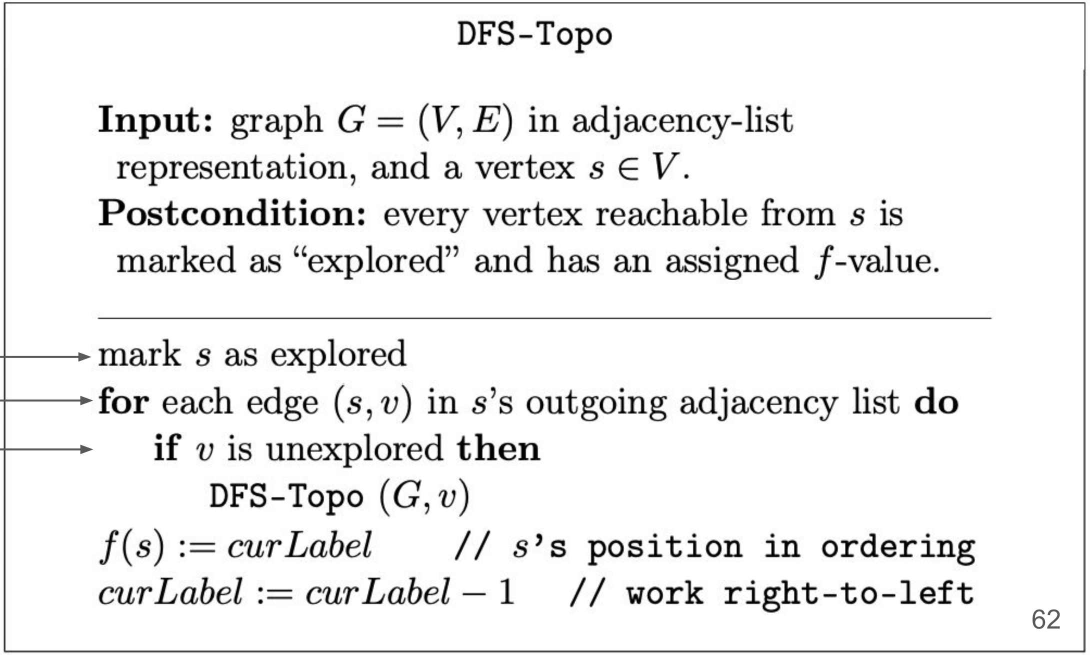
curLabel = ~~4~~ ~~3~~ 2

f(v) = 3

f(t) = 4

f(s) = ?

f(w) = ?



DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list representation, and a vertex $s \in V$.

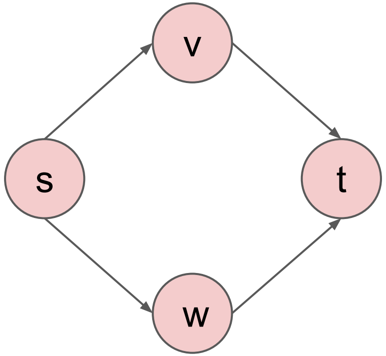
Postcondition: every vertex reachable from s is marked as “explored” and has an assigned f-value.

```
mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$          // s's position in ordering
$curLabel := curLabel - 1$   // work right-to-left
```

"w"

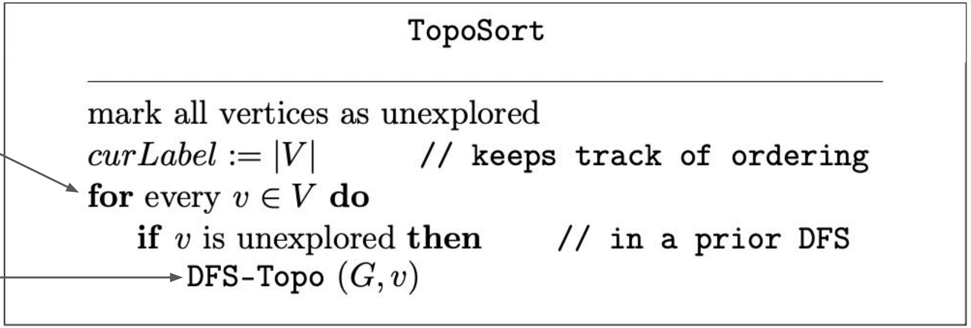
(w, t)

"t" already explored



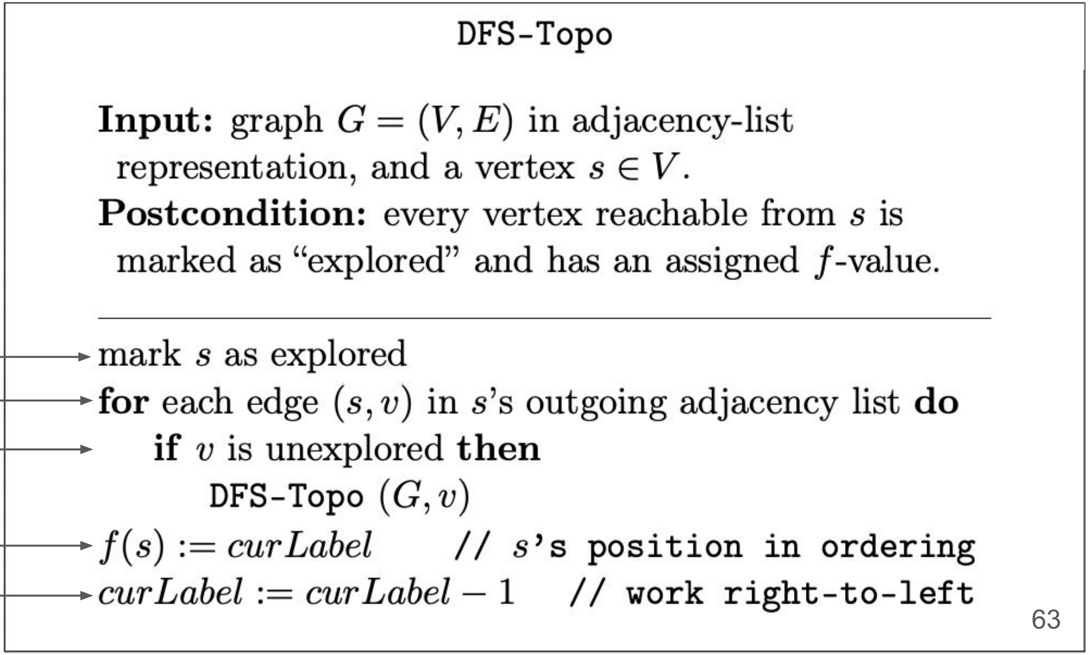
curLabel = ~~4~~ ~~3~~ ~~2~~ 1
f(v) = 3
f(t) = 4
f(s) = ?
f(w) = 2

Let's assume the order "v, t, **s**, w"



```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

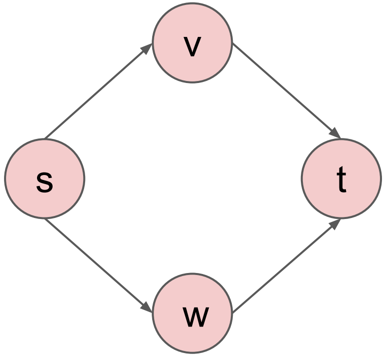
"s"



```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$       // s's position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```

"w"
(w, t)
"t" already explored

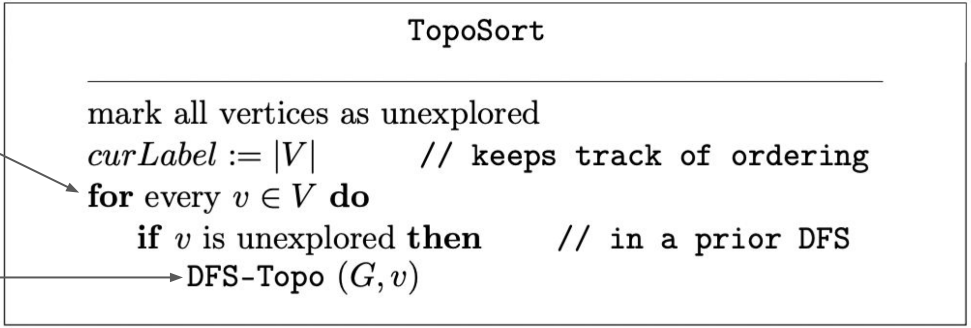
f(w) = 2
curLabel = 1



curLabel = ~~4~~ ~~3~~ ~~2~~ 1
f(v) = 3
f(t) = 4
f(s) = ?
f(w) = 2

Let's assume the order "v, t, **s**, w"

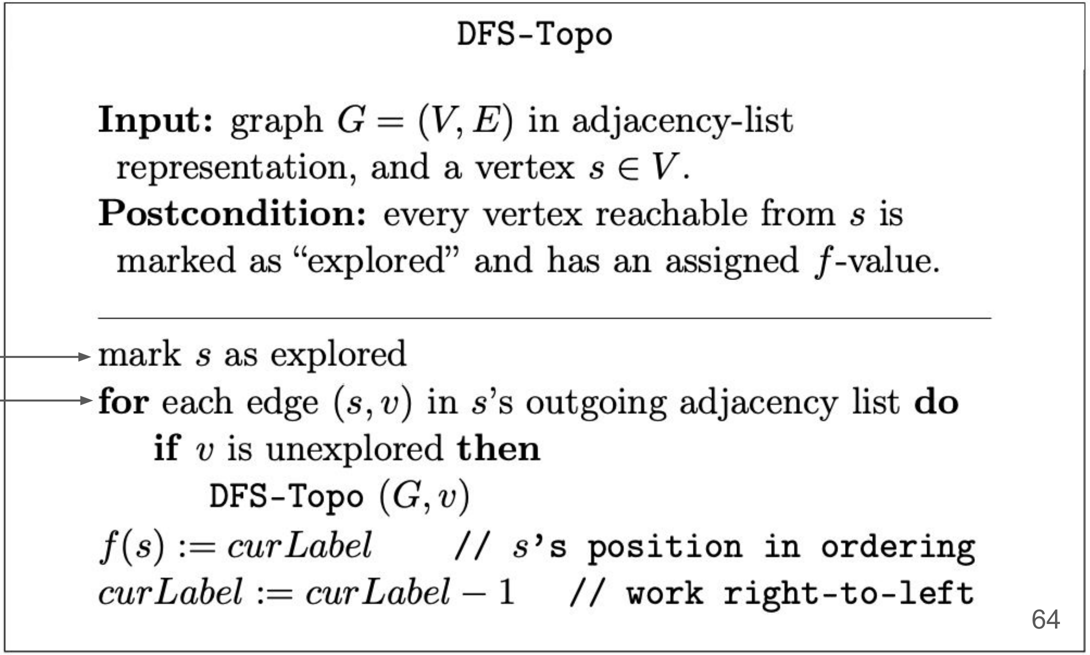
"s"



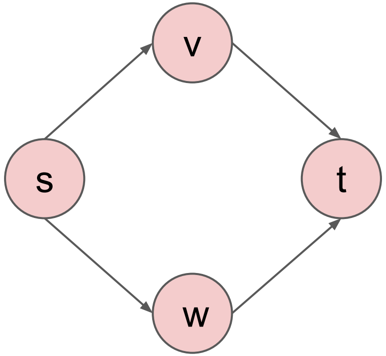
```
TopoSort
mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then       // in a prior DFS
        DFS-Topo $(G, v)$
```

"s"

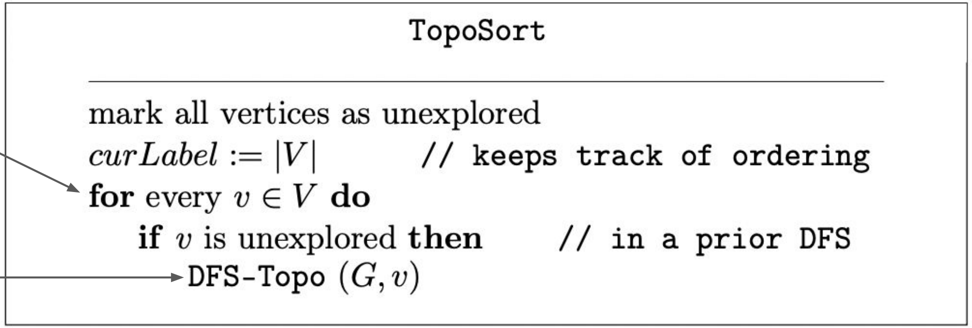
No other edge



```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$       // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```



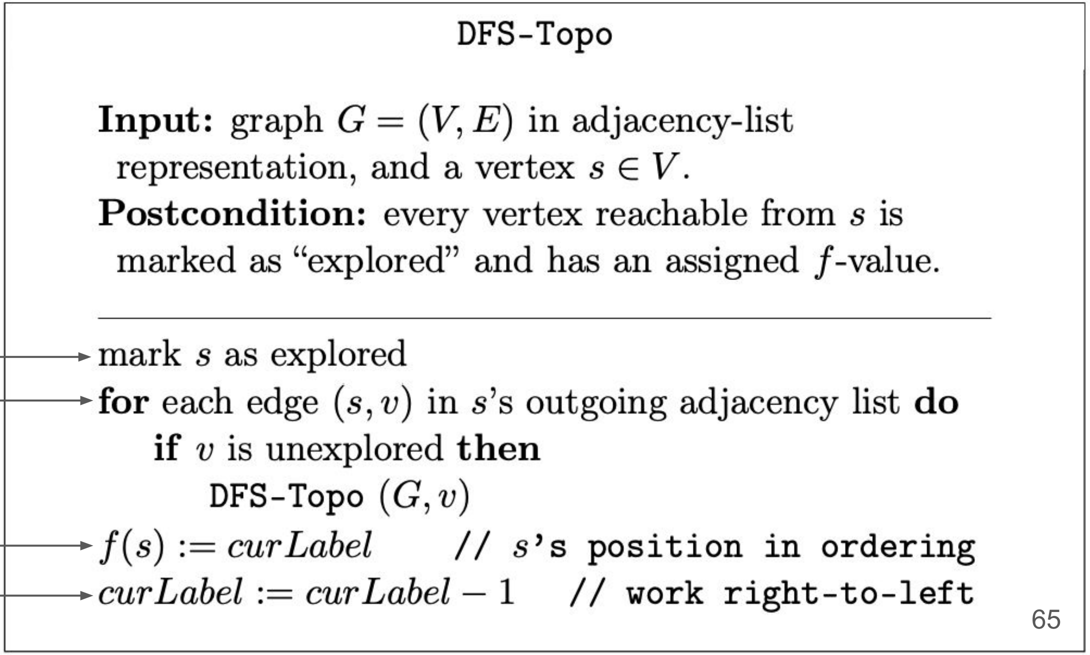
Let's assume the order "v, t, **s**, w"



```
TopoSort
mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

"s"

curLabel = ~~4~~ ~~3~~ ~~2~~ ~~1~~ 0
f(v) = 3
f(t) = 4
f(s) = 1
f(w) = 2



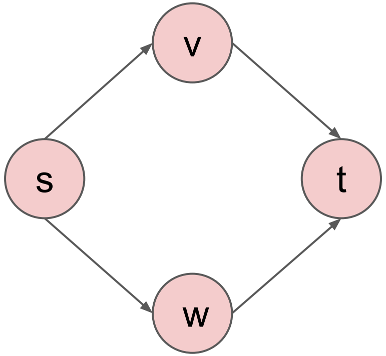
```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$      // s's position in ordering
$curLabel := curLabel - 1$     // work right-to-left
```

"s"

No other edge

f(s) = 1

curLabel = 0



curLabel = ~~4~~ ~~3~~ ~~2~~ ~~1~~ 0

f(v) = 3

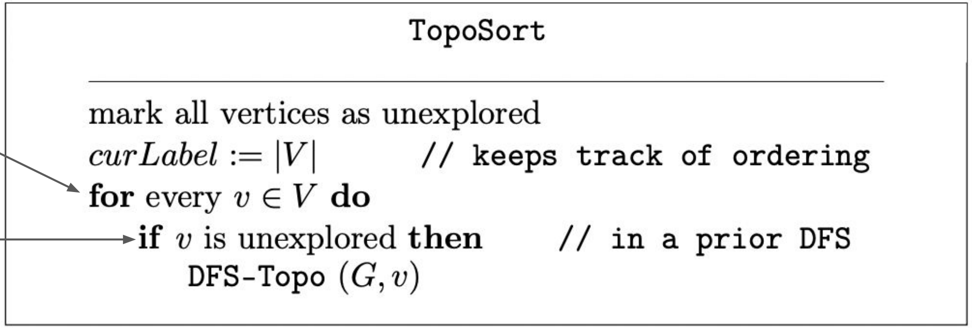
f(t) = 4

f(s) = 1

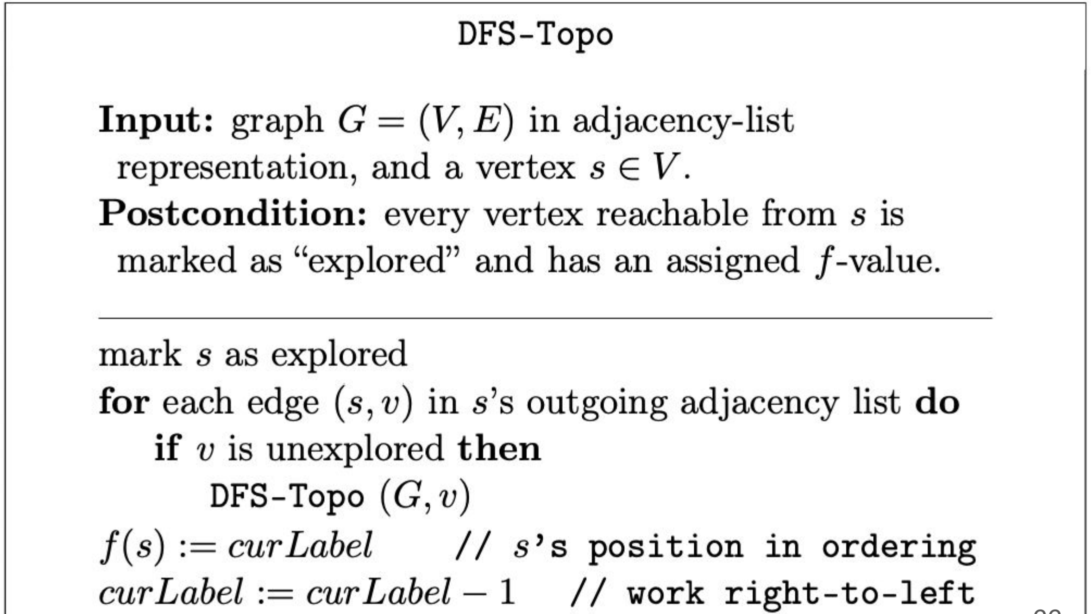
f(w) = 2

Let's assume the order "v, t, s, **w**"

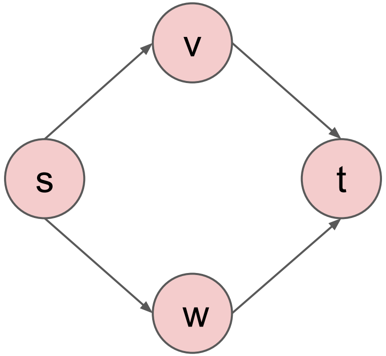
"w"



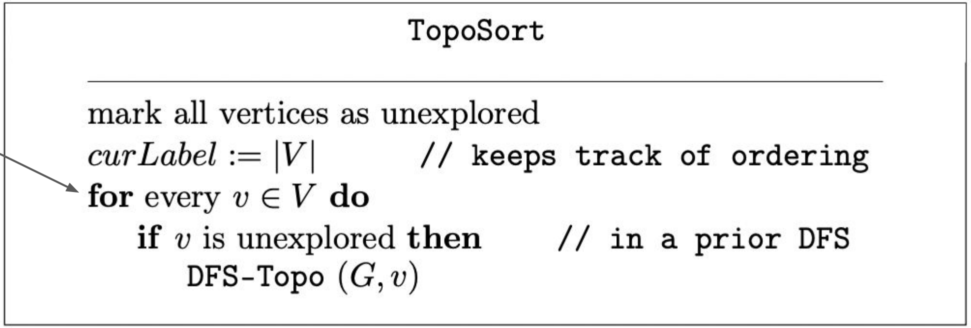
```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then       // in a prior DFS
        DFS-Topo $(G, v)$
```



```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
    representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
    marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$        // s's position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```



Let's assume the order "v, t, s, w"



```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$          // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then       // in a prior DFS
        DFS-Topo $(G, v)$
```

curLabel = ~~4~~ ~~3~~ ~~2~~ ~~1~~ 0

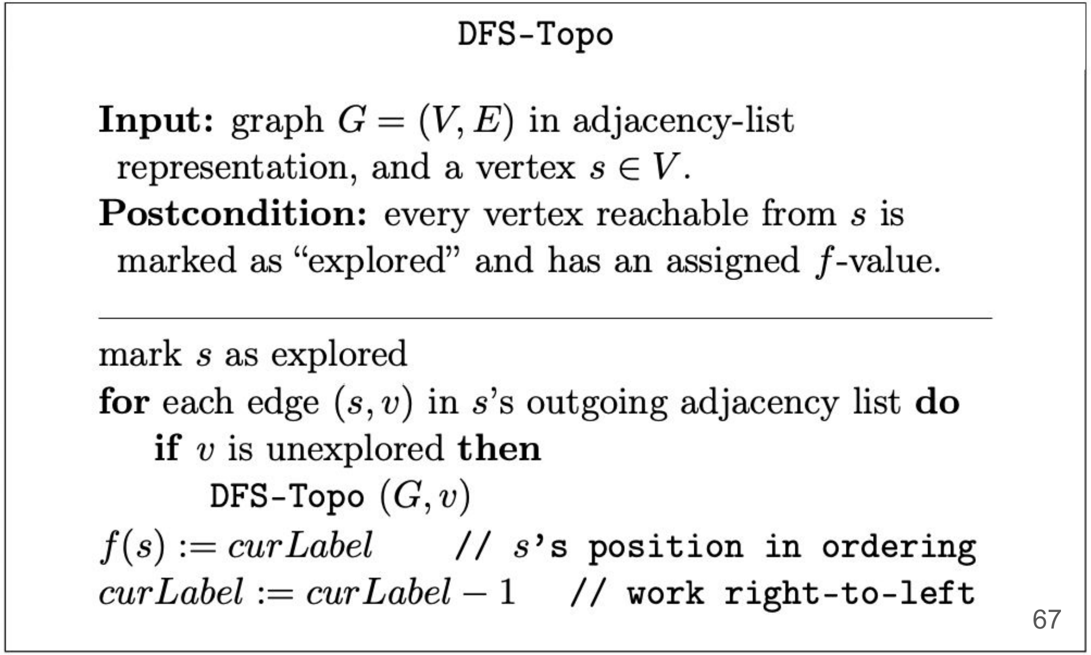
f(v) = 3

f(t) = 4

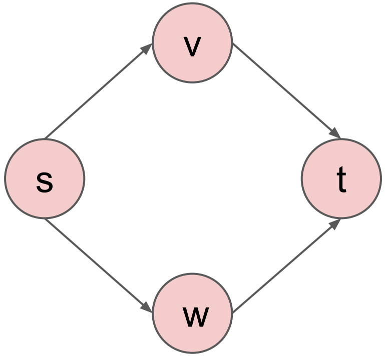
f(s) = 1

f(w) = 2

End.



```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$       // $s$'s position in ordering
$curLabel := curLabel - 1$     // work right-to-left
```

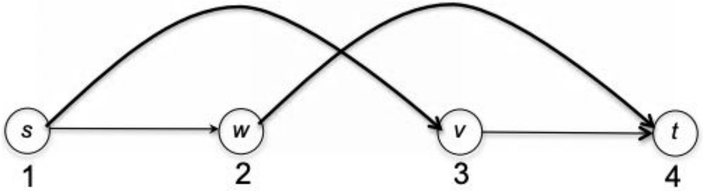


Let's assume the order "v, t, s, w"

```
TopoSort

mark all vertices as unexplored
$curLabel := |V|$        // keeps track of ordering
for every $v \in V$ do
    if $v$ is unexplored then      // in a prior DFS
        DFS-Topo $(G, v)$
```

curLabel = ~~4~~ ~~3~~ ~~2~~ ~~1~~ 0
f(v) = 3
f(t) = 4
f(s) = 1
f(w) = 2

End.



```
DFS-Topo

Input: graph $G = (V, E)$ in adjacency-list
 representation, and a vertex $s \in V$.
Postcondition: every vertex reachable from $s$ is
 marked as "explored" and has an assigned $f$-value.

mark $s$ as explored
for each edge $(s, v)$ in $s$'s outgoing adjacency list do
    if $v$ is unexplored then
        DFS-Topo $(G, v)$
$f(s) := curLabel$       // $s$'s position in ordering
$curLabel := curLabel - 1$    // work right-to-left
```