DEPTH-FIRST SEARCH-BASED ALGORITHM

→ based on Murphy's law: "whatever you're starting to do, you realize something else"

Input:
G - directed graph
Output:
sorted: a list of vertices in topological sorting order, or NULL, if G has cycles

```
Subalgorithm TOPOSORTDFS(Graph G, Vertex x, List sorted,
                         set FullyProcessed, set inProcess)
    inProcess.add(x)
    for y in Nin(x)
        if y in inProcess then //there is a cycle
            return false
        else if y not in FullyProcessed then
            ok=topoSortDFS(G,y,sorted,fullyProcessed,
                           inProcess)
            if not ok then
                return false
    inProcess.remove(x)
    sorted.append(x)
    fullyProcessed.add(x)
    return true

Algorithm:
    sorted = emptyList
    fullyProcessed = emptySet
    inProcess = emptySet
    for x in X do
        if x not in fullyProcessed then
            ok=topoSortDFS(G,x,sorted,fullyProcessed,inProcess)
            if not ok then
                sorted=null
                return
```

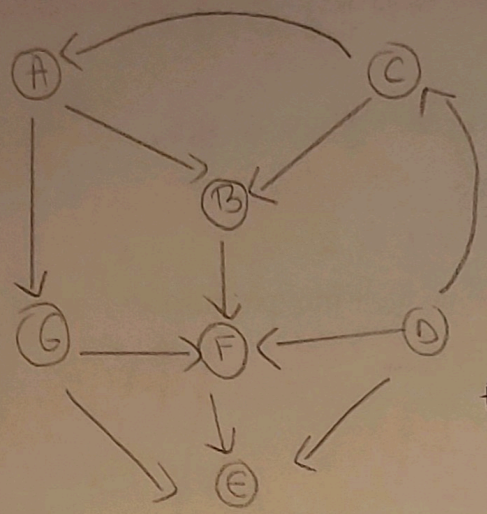
Highest cost path between two vertices in $O(V+E)$

Generalized: longest path cost in graph G

Topologically sort G
for each vertex $v \in V$ in linearized order
do $dist(v) = \max_{(u,v)\in E}\{dist(u)+w(u,v)\}$
return $\max_{v\in V}\{dist\ v\}$

```
dist[] = {-INF, -INF, ...}
dist[c] = 0 ; // source vertex
  for every vertex u in topological order
        if (u == z) break; // destination vertex
        for every adjacent vertex v of u
            if ((dist[u] + weight(u,v)) > dist[v])
                dist[v] = dist[u] + weight(u,v)
return dist[z]
```

TOPOSORTDFS



Topological sorting of the DAG using the DFS method.

For each entry and exit of the recursive function show the execution stack and the sorted list so far.

In the end: the final list of vertices.

A topological sorting: D C A B G F E

calls	x,y	unProcess	FullyProcessed	sorted	ok
initialization		{}	{}	[]	
TopoSortDFS(G,A,[],{},{})	x=A y=C	{} {A}	{}	[]	
TopoSortDFS(G,C,[],{},{A})	x=C y=D	{A} {A,C}	{}	[]	
TopoSortDFS(G,D,[],{},{A,C})	x=D ←	{A,C,D} {A,C}	{D}	[D]	true
	x=C	{A}	{D,C}	[D,C]	true
	x=A	{}	{D,C,A}	[D,C,A]	true
TopoSortDFS(G,B,[D,C,A],{B,C,A},{})	x=B y=A y=C	{} {B} {}	{D,C,A} {D,C,A,B}	[D,C,A] [D,C,A,B]	true
TopoSortDFS(G,G,[D,C,A,B],{D,C,A,B},{})	x=G y=A	{} {G} {}	{D,C,A,B} {D,C,A,B,G}	[D,C,A,B] [D,C,A,B,G]	true
TopoSortDFS(G,E,[D,C,A,B,G],{D,C,A,B,G},{})	x=E y=D y=G y=F	{} {E}	{D,C,A,B,G}	[D,C,A,B,G]	
TopoSortDFS(G,F,...)	x=F y=B y=G y=D	{E} {E,F} {E}	{D,C,A,B,G} {D,C,A,B,G,F}	[D,F,A,B,G] [D,C,A,B,G,F]	true
		{}	{D,C,A,B,G,F,E}	[D,C,A,B,G,F,E]	true