Lecturer: Andrew Parkes
http://www.cs.nott.ac.uk/~pszajp/

G52ACE 2017-18
Graph DFS for Cycle Detection

Simple DFS respecting the edge directions

DFS starting from vertex v:

```
create a stack S
mark v as visited and push v onto S
while S is non-empty
  peek at the top u of S
  if u has an (unvisited) neighbour w such that
   (u,w) is a (directed) edge
           mark w and push it onto S
  else pop S
```

NOTE: if store the directed edges as adjacency lists then the above is natural anyway

Crucial Property of DFS

- The stack always defines a (directed) path
  - because the next element is always a neighbour
  - this ability of DFS to find paths results in it often being used as the basis for other algorithms
    - Note: it finds "a path" but generally not the shortest path!!!

Modification of depth first search

- How to get DFS to detect cycles in a directed graph:

idea: if we encounter a vertex which is already on the stack, we found a loop (stack contains vertices on a path, and if we see the same vertex again, the path must contain a cycle).

- Instead of visited and unvisited, use three 'colours':
  - **white =** unvisited ("not done")
  - **grey =** on the stack ("half done")
  - **black =** finished (we backtracked from it, seen everywhere we can reach from it)

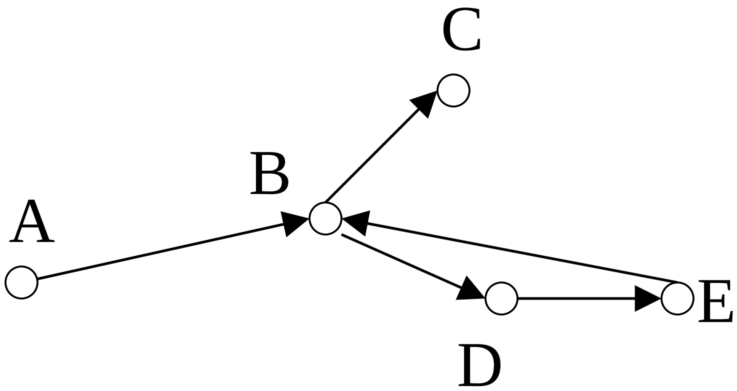
Modification of depth first search

Modified DFS starting from **`v`**:

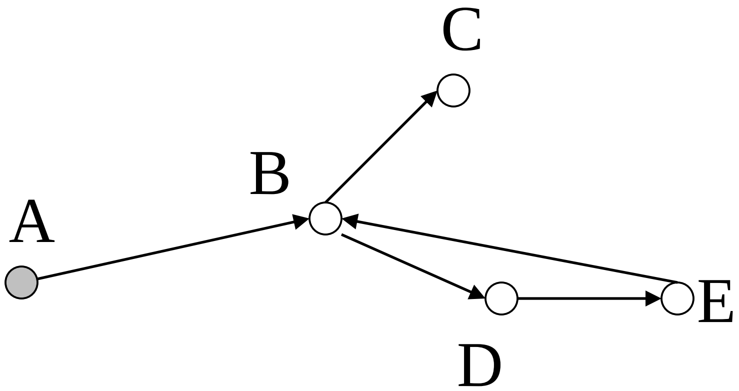
```
all vertices coloured white
create a stack S
colour v grey and push v onto S
while S is non-empty
  peek at the top u of S
  if u has a grey neighbour, found a cycle
  else if u has a white neighbour w,
          colour w grey and push it onto S
  else colour u black and pop S
```

Tracing modified DFS from A

S = {}



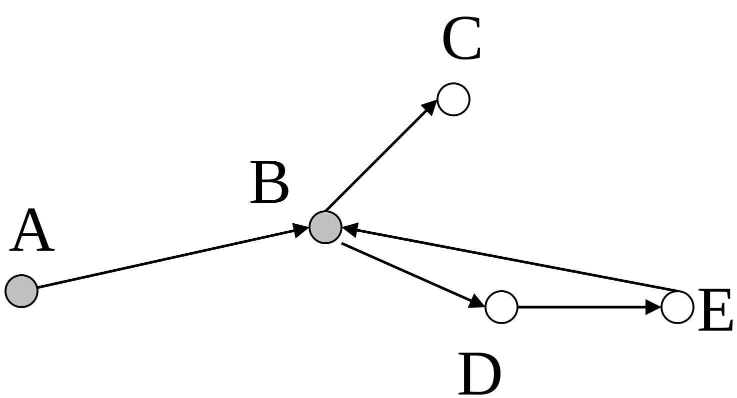
Tracing modified DFS from A



S = {}

S = A

Tracing modified DFS from A

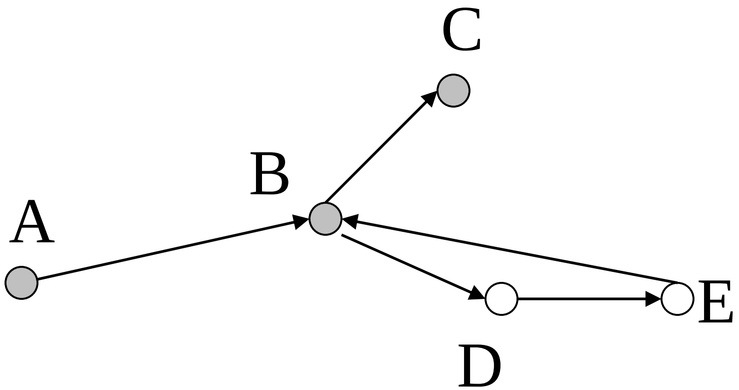


S = {}

S = A

S = A B

Tracing modified DFS from A



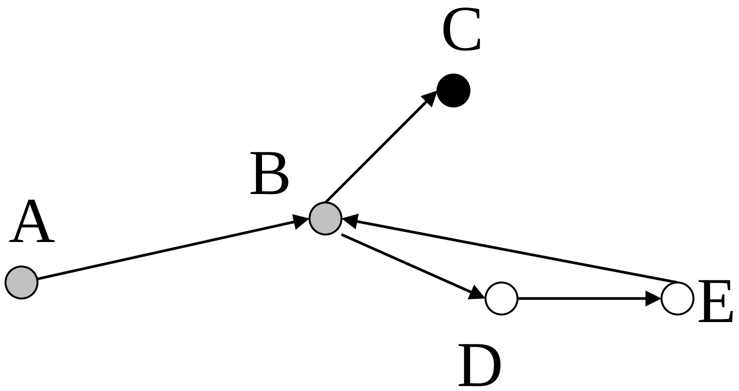
S = {}

S = A

S = A B

S = A B C

Tracing modified DFS from A



S = {}

S = A

S = A B

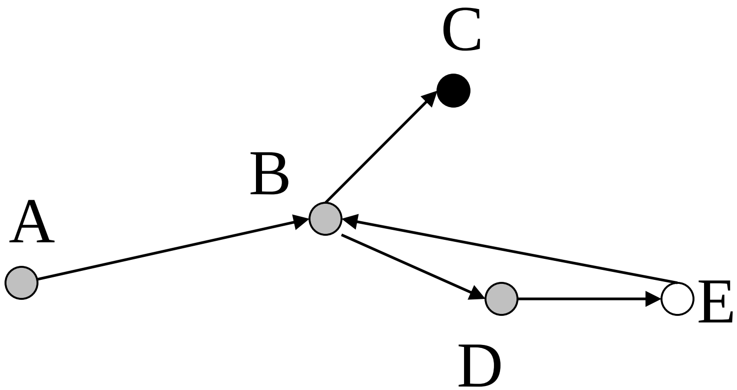
S = A B C

pop: S = A B

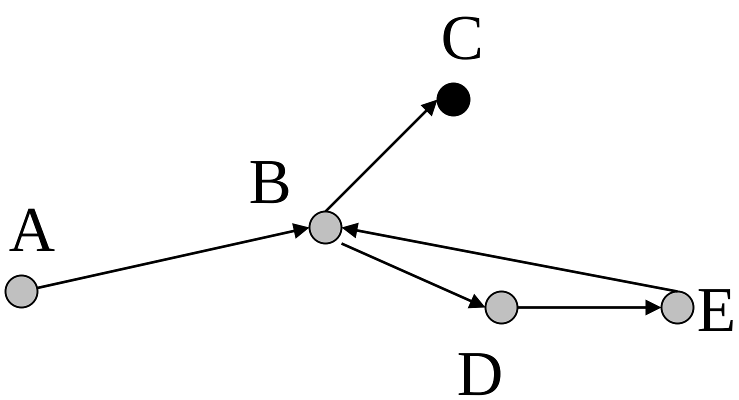
Tracing modified DFS from A

push:

S = A B D



Tracing modified DFS from A

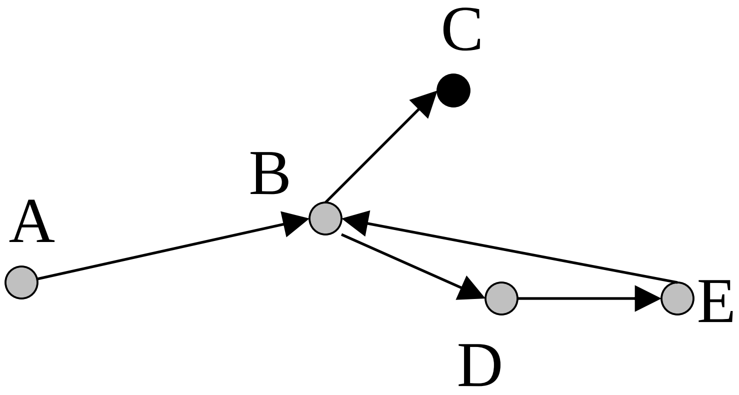


push:

S = A B D

S = A B D E

Tracing modified DFS from A



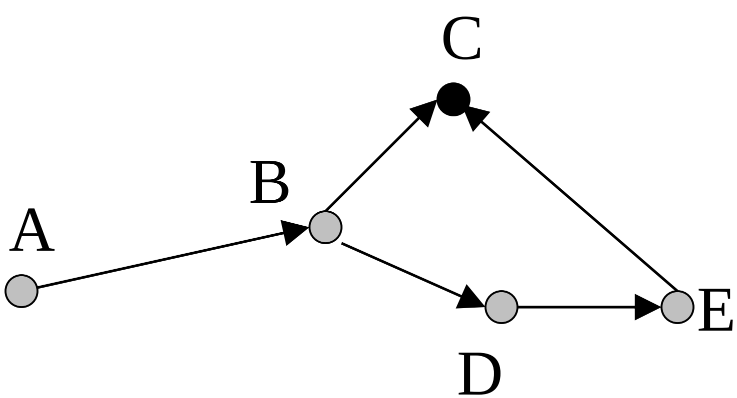
push:

S = A B D

S = A B D E

E has a grey neighbour: B!
Found a loop!

Another example



Suppose were doing a similar graph with no edge EB, but instead an edge EC.

Then the algorithm does not find a cycle. But this is correct as this graph is acyclic.

A cycle needs to respect the directions of the arrows – it is a path back to itself.

If think of nodes as webpages, and directed edges as links, then a cycle means: can just follow links and end up back on the starting web page.

Summary

- DFS can be modified to detect cycles in directed graphs