ITMO UNIVERSITY

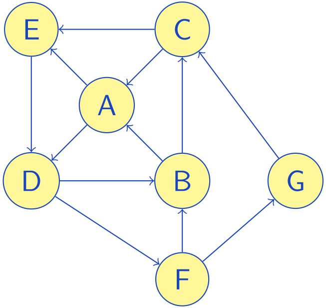
How to Win Coding Competitions: Secrets of Champions

Week 6: Algorithms on Graphs 2
Lecture 1: Eulerian paths and Eulerian tours

Maxim Buzdalov
Saint Petersburg 2016

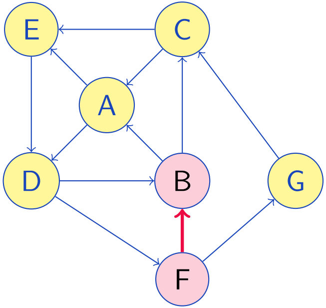
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



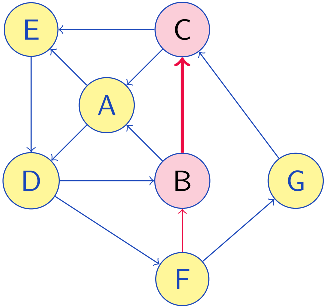
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



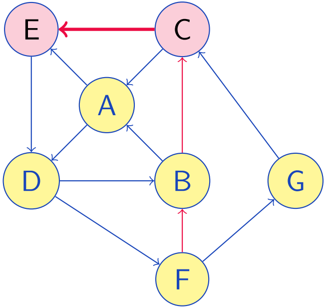
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



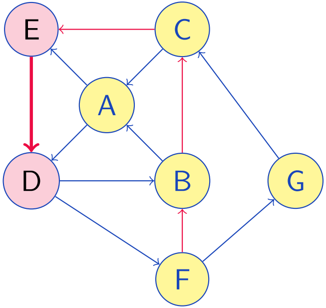
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



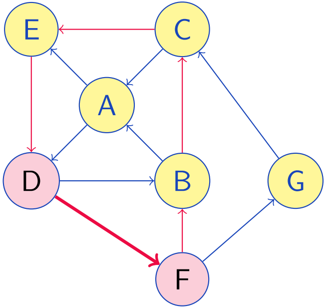
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



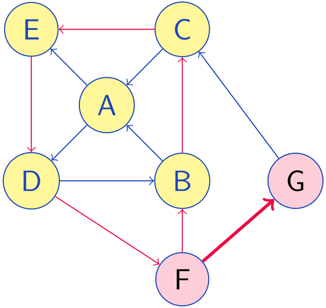
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



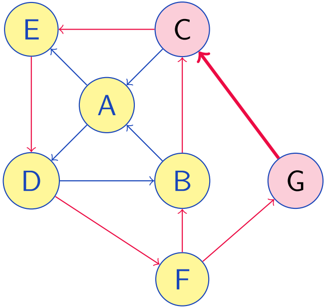
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



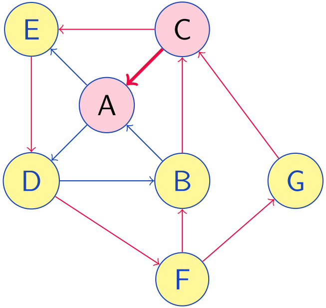
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



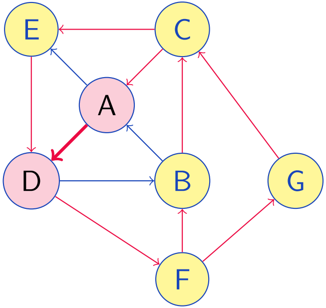
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



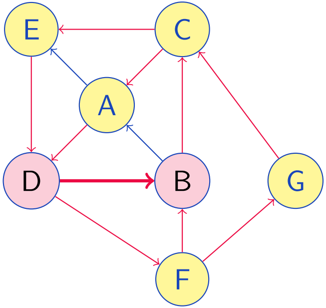
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



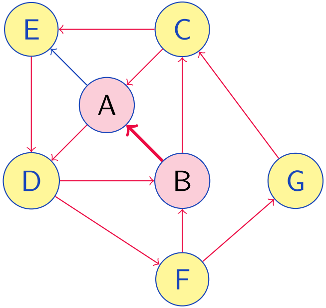
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



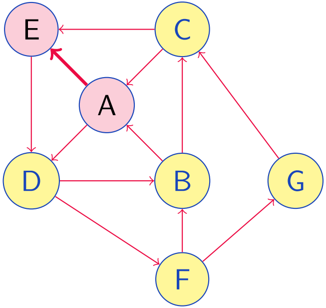
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



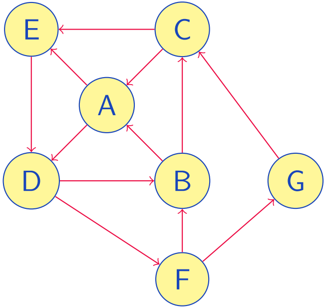
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



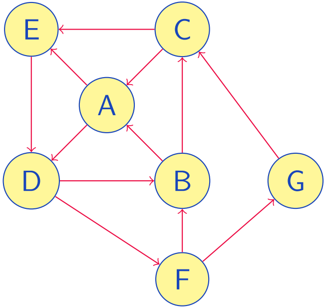
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



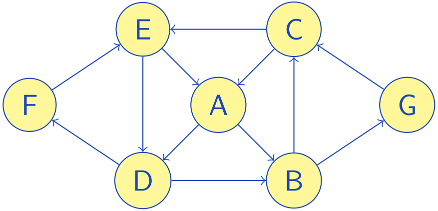
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



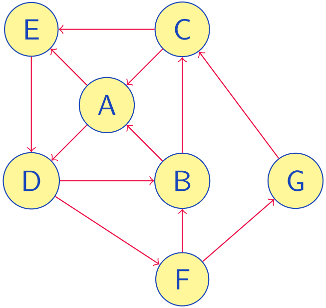
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



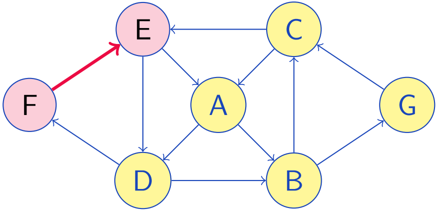
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



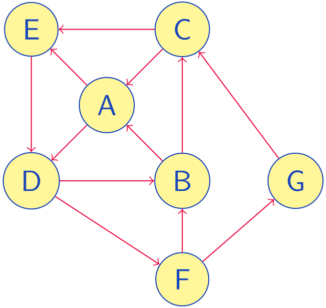
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



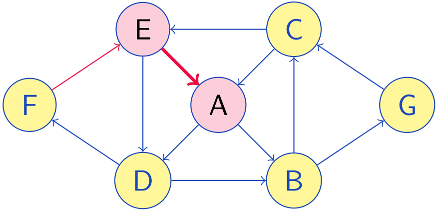
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



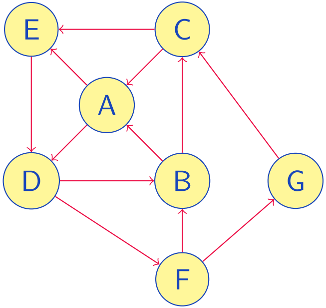
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



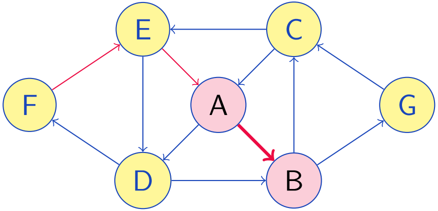
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



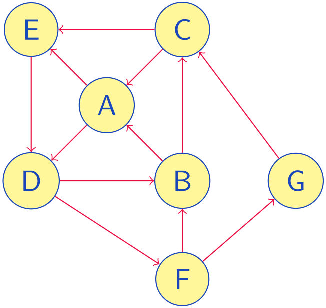
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



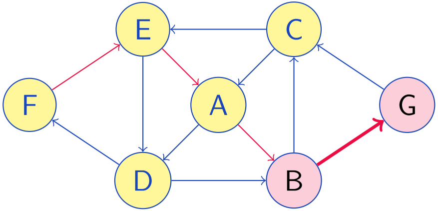
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



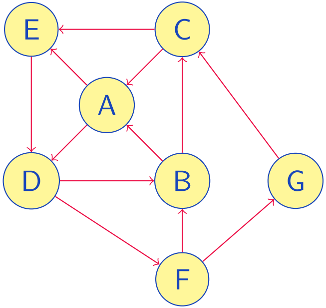
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



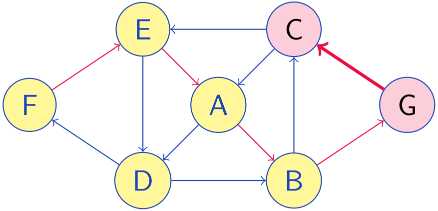
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



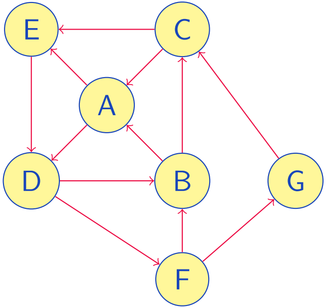
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



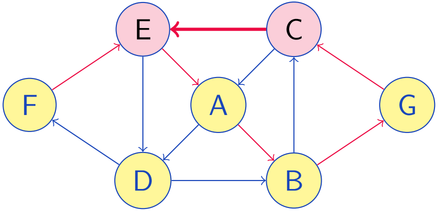
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



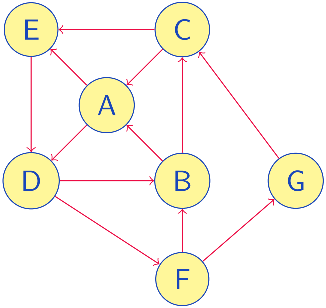
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



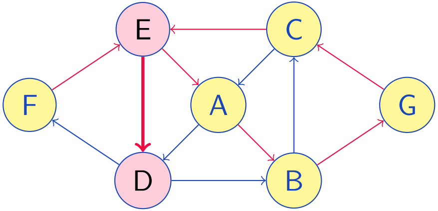
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



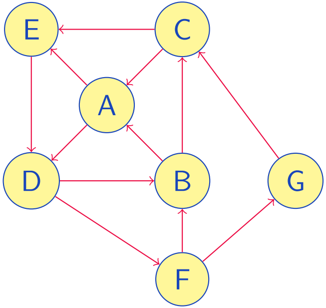
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



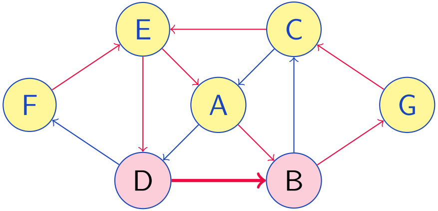
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



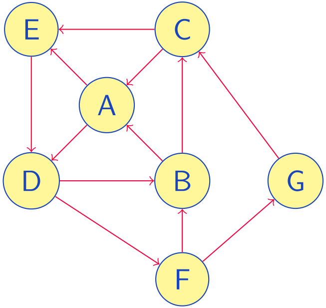
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



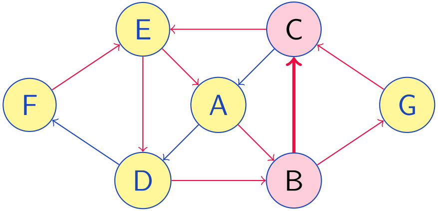
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



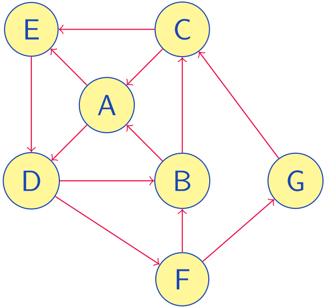
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



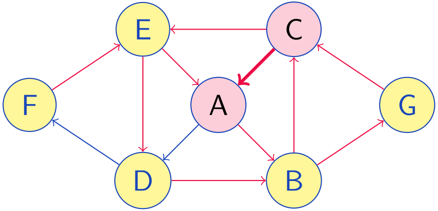
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



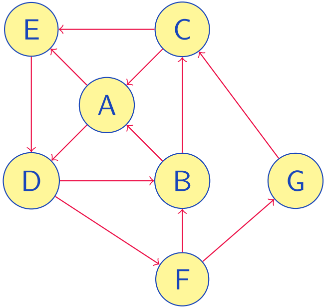
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



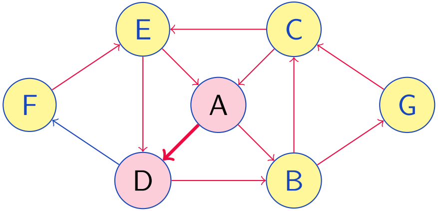
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



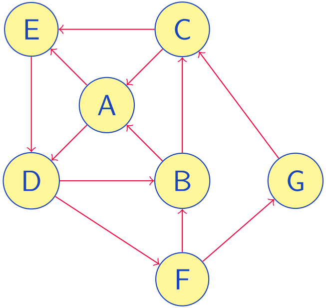
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



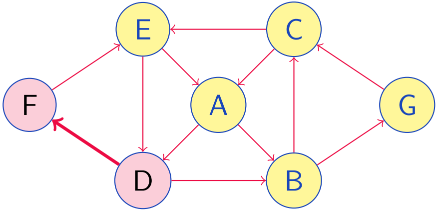
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



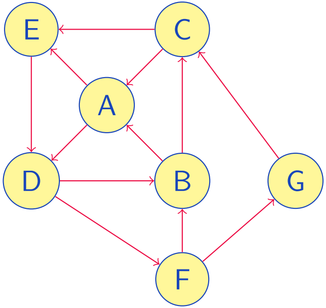
An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



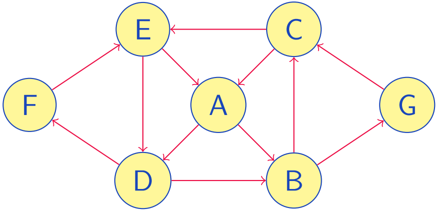
An Eulerian path is a path in a graph that contains each edge of the graph exactly once

FBCEDFGCADBAE



An Eulerian tour is a Eulerian path which starts and ends on the same vertex

FEABGCEDBCADF



When does Eulerian path exist?

When does Eulerian path exist?

- Undirected graph:
  - The graph is connected
  - There are at most two vertices with odd degree

When does Eulerian path exist?

- Undirected graph:
  - The graph is connected
  - There are at most two vertices with odd degree
- Directed graph:
  - The graph is connected (when directions are removed)
  - At most one vertex u has $deg^+(u) - deg^-(u) = +1$
  - At most one vertex v has $deg^+(v) - deg^-(v) = -1$
  - All other vertices have $deg^+(x) = deg^-(x)$

When does Eulerian path exist?

- Undirected graph:
  - The graph is connected
  - There are at most two vertices with odd degree
- Directed graph:
  - The graph is connected (when directions are removed)
  - At most one vertex u has $deg^+(u) - deg^-(u) = +1$
  - At most one vertex v has $deg^+(v) - deg^-(v) = -1$
  - All other vertices have $deg^+(x) = deg^-(x)$

When does Eulerian tour exist?

When does Eulerian path exist?

- Undirected graph:
  - The graph is connected
  - There are at most two vertices with odd degree
- Directed graph:
  - The graph is connected (when directions are removed)
  - At most one vertex u has $deg^+(u) - deg^-(u) = +1$
  - At most one vertex v has $deg^+(v) - deg^-(v) = -1$
  - All other vertices have $deg^+(x) = deg^-(x)$

When does Eulerian tour exist?

- Undirected graph:
  - The graph is connected
  - All vertex degrees are even

When does Eulerian path exist?

- Undirected graph:
  - The graph is connected
  - There are at most two vertices with odd degree
- Directed graph:
  - The graph is connected (when directions are removed)
  - At most one vertex u has $deg^+(u) - deg^-(u) = +1$
  - At most one vertex v has $deg^+(v) - deg^-(v) = -1$
  - All other vertices have $deg^+(x) = deg^-(x)$

When does Eulerian tour exist?

- Undirected graph:
  - The graph is connected
  - All vertex degrees are even
- Directed graph:
  - The graph is strongly connected
  - All vertices have $deg^+(x) = deg^-(x)$

- Induction: assume that your graph has $|E|$ edges,
  and the theorem was proven for all $e < |E|$ numbers of edges

- Induction: assume that your graph has $|E|$ edges,
  and the theorem was proven for all $e < |E|$ numbers of edges
- Pick a vertex
  - Random when you don't have "special" vertices
  - A "special" vertex if you have one
    - The one with $deg^+(v) > deg^-(v)$ if the graph is directed

- Induction: assume that your graph has $|E|$ edges,
  and the theorem was proven for all $e < |E|$ numbers of edges
- Pick a vertex
  - Random when you don't have "special" vertices
  - A "special" vertex if you have one
    - The one with $deg^+(v) > deg^-(v)$ if the graph is directed
- Traverse the graph and remove the traversed edges

- Induction: assume that your graph has $|E|$ edges, and the theorem was proven for all $e < |E|$ numbers of edges
- Pick a vertex
  - Random when you don't have "special" vertices
  - A "special" vertex if you have one
    - The one with $deg^{+}(v) > deg^{-}(v)$ if the graph is directed
- Traverse the graph and remove the traversed edges
- If you cannot do this anymore, what happens?
  - Either there are no more edges $\rightarrow$ path/tour is found
  - Some edges remain
    - There are (maybe several) connected subgraphs
    - Find the paths/tours in them (can do this by induction)
    - Connect them with the path/tour constructed from removed edges

- Induction: assume that your graph has $|E|$ edges, and the theorem was proven for all $e < |E|$ numbers of edges
- Pick a vertex
  - Random when you don't have "special" vertices
  - A "special" vertex if you have one
    - The one with $deg^+(v) > deg^-(v)$ if the graph is directed
- Traverse the graph and remove the traversed edges
- If you cannot do this anymore, what happens?
  - Either there are no more edges $\rightarrow$ path/tour is found
  - Some edges remain
    - There are (maybe several) connected subgraphs
    - Find the paths/tours in them (can do this by induction)
    - Connect them with the path/tour constructed from removed edges
- That's what Depth First Search can do!

$G = \langle V, E \rangle$
$A(v) = \{u \mid (v, u) \in E\}$
$R \leftarrow [\,]$
procedure $\mathrm{DFS}(v)$
    for $u \in A(v)$ **do**
        Remove u from $A(v)$
        if graph is undirected **then**
            Remove v from $A(u)$
        end if
        $\mathrm{DFS}(u)$
    end for
    $R \leftarrow [v] + R$
end procedure

$G = \langle V, E \rangle$
$A(v) = \{u \mid (v, u) \in E\}$ $\triangleright$ No U set is used: can visit a vertex more than once!
$R \leftarrow [\,]$
procedure $\mathrm{DFS}(v)$
  for $u \in A(v)$ **do**
    Remove u from $A(v)$
    if graph is undirected **then**
      Remove v from $A(u)$
    end if
    $\mathrm{DFS}(u)$
  end for
  $R \leftarrow [v] + R$
end procedure

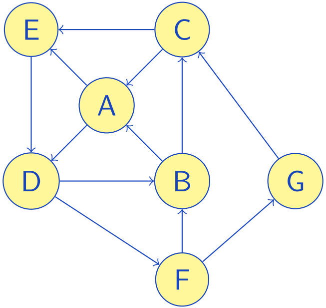
$G = \langle V, E \rangle$
$A(v) = \{u \mid (v, u) \in E\}$ ▷ No U set is used: can visit a vertex more than once!
$R \leftarrow [\,]$ ▷ The result storage
procedure $\mathrm{DFS}(v)$
    for $u \in A(v)$ **do**
        Remove u from $A(v)$
        if graph is undirected **then**
            Remove v from $A(u)$
        end if
        $\mathrm{DFS}(u)$
    end for
    $R \leftarrow [v] + R$
end procedure

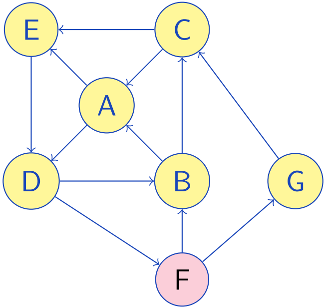
$G = \langle V, E \rangle$
$A(v) = \{u \mid (v, u) \in E\}$ ▷ No U set is used: can visit a vertex more than once!
$R \leftarrow [\,]$ ▷ The result storage
procedure $\mathrm{DFS}(v)$ ▷ Should be called on a non-regular vertex, if any
    for $u \in A(v)$ **do**
        Remove u from $A(v)$
        if graph is undirected **then**
            Remove v from $A(u)$
        end if
        $\mathrm{DFS}(u)$
    end for
    $R \leftarrow [v] + R$
end procedure

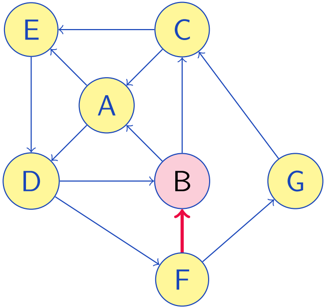
$G = \langle V, E \rangle$
$A(v) = \{u \mid (v, u) \in E\}$ ▷ No U set is used: can visit a vertex more than once!
$R \leftarrow [\,]$ ▷ The result storage
procedure $\mathrm{DFS}(v)$ ▷ Should be called on a non-regular vertex, if any
    for $u \in A(v)$ **do**
        Remove u from $A(v)$ ▷ Will never follow the (v, u) edge anymore
        if graph is undirected **then**
            Remove v from $A(u)$
        end if
        $\mathrm{DFS}(u)$
    end for
    $R \leftarrow [v] + R$
end procedure

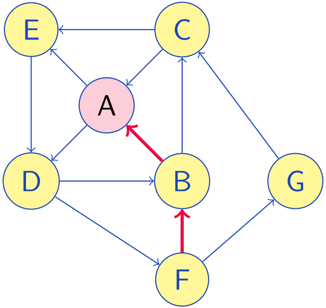
$G = \langle V, E \rangle$
$A(v) = \{u \mid (v, u) \in E\}$ ▷ No U set is used: can visit a vertex more than once!
$R \leftarrow [\,]$ ▷ The result storage
procedure DFS(v) ▷ Should be called on a non-regular vertex, if any
    for $u \in A(v)$ **do**
        Remove u from $A(v)$ ▷ Will never follow the (v, u) edge anymore
        if graph is undirected **then**
            Remove v from $A(u)$ ▷ If undirected, the anti-edge must be removed
        end if
        DFS(u)
    end for
    $R \leftarrow [v] + R$
end procedure

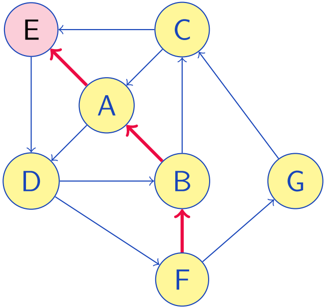
$G = \langle V, E \rangle$
$A(v) = \{u \mid (v, u) \in E\}$ ▷ No U set is used: can visit a vertex more than once!
$R \leftarrow [\,]$ ▷ The result storage
procedure $\mathrm{DFS}(v)$ ▷ Should be called on a non-regular vertex, if any
    for $u \in A(v)$ **do**
        Remove u from $A(v)$ ▷ Will never follow the (v, u) edge anymore
        if graph is undirected **then**
            Remove v from $A(u)$ ▷ If undirected, the anti-edge must be removed
        end if
        $\mathrm{DFS}(u)$
    end for
    $R \leftarrow [v] + R$ ▷ Prepend v to the answer
end procedure

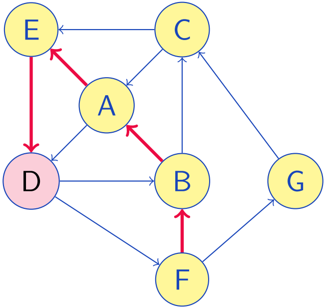

E
C
A
D
B
G
F

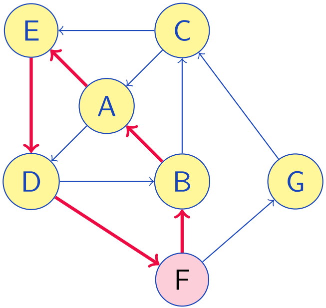

E
C
A
D
B
G
F


E
C
A
D
B
G
F

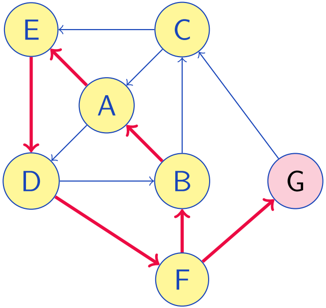

E
C
A
D
B
G
F

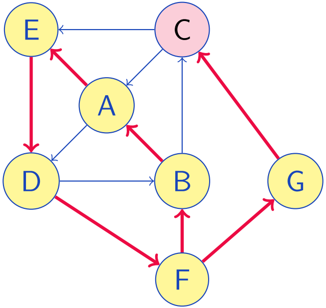

E
C
A
D
B
G
F

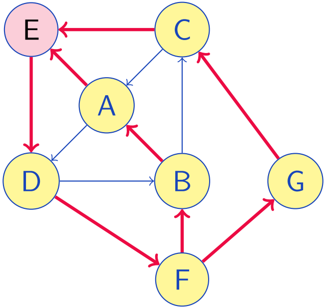

E
C
A
D
B
G
F


E
C
A
D
B
G
F

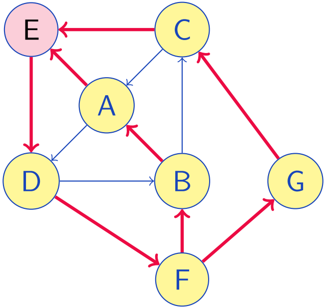
Example: DFS finds Eulerian path



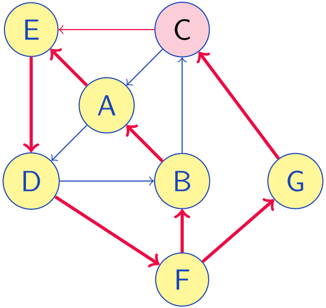

E
C
A
D
B
G
F


E
C
A
D
B
G
F

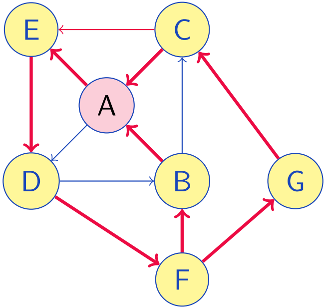
E



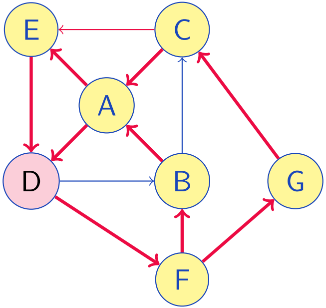
E



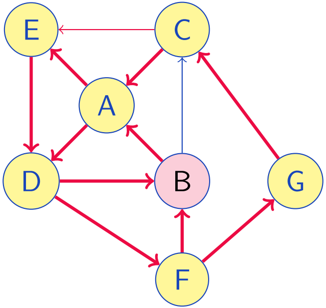
E



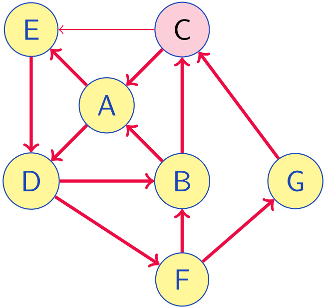
E



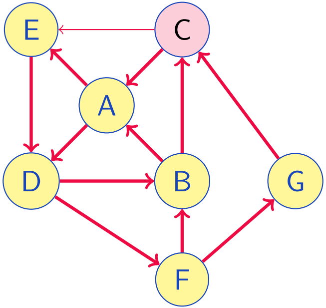
E



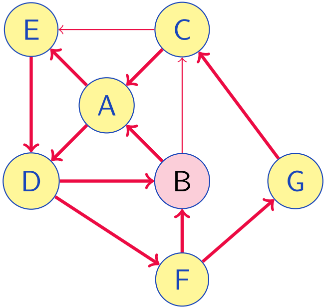
E



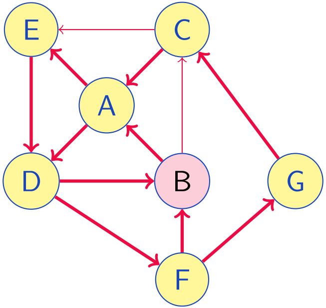
CE



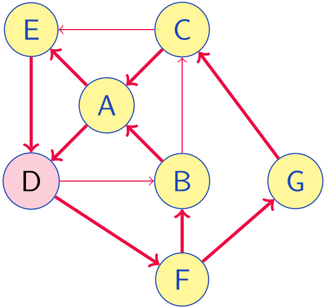
CE



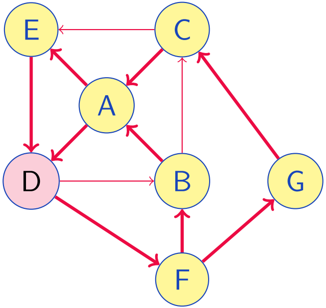
BCE



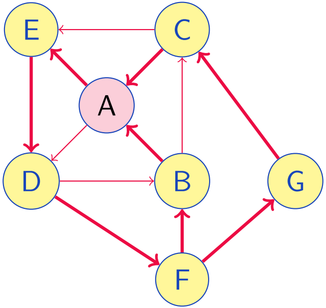
BCE



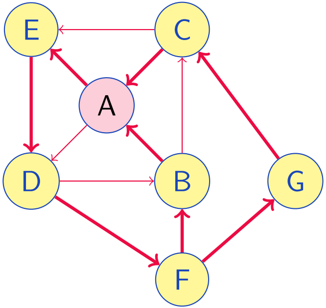
DBCE



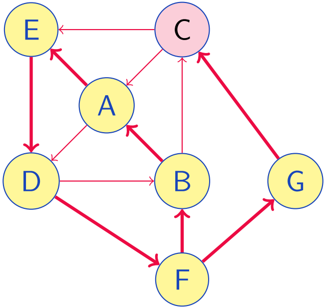
DBCE



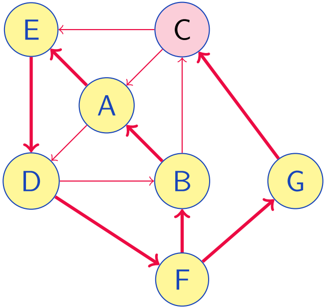
ADBCE



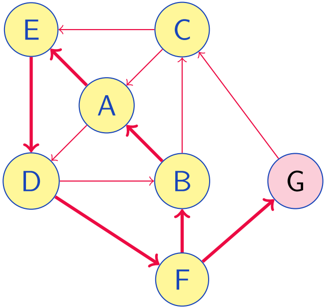
ADBCE



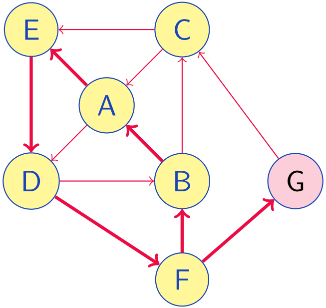
CADBCE



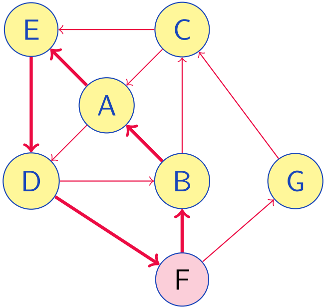
CADBCE



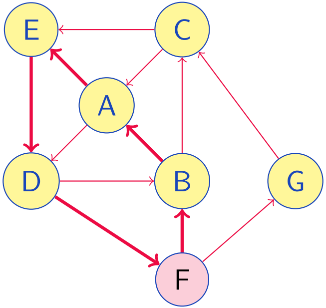
GCADBCE



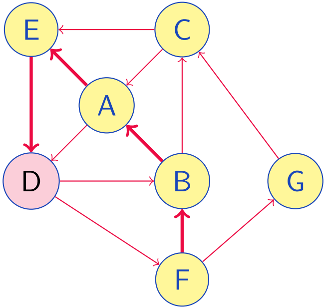
GCADBCE



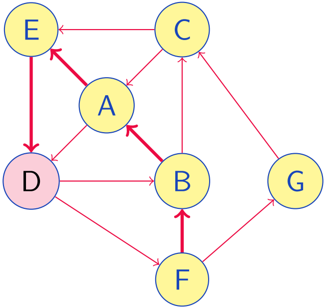
FGCADBCE



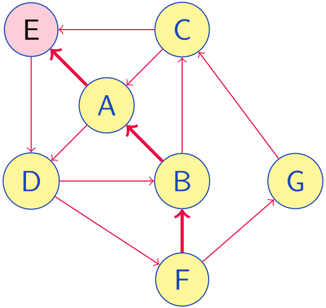
FGCADBCE



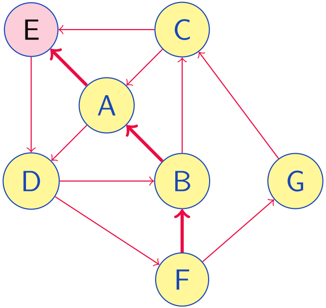
DFGCADBCE



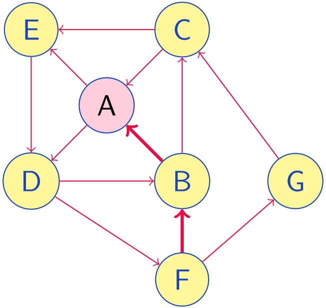
DFGCADBCE



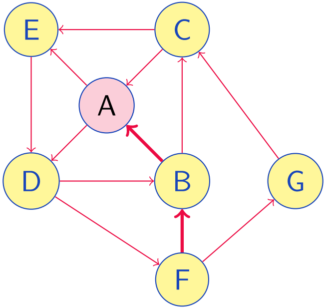
EDFGCADBCE



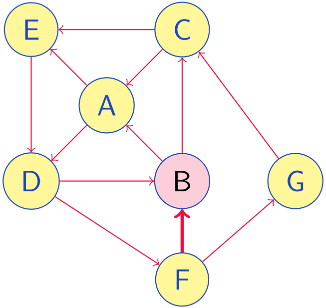
EDFGCADBCE



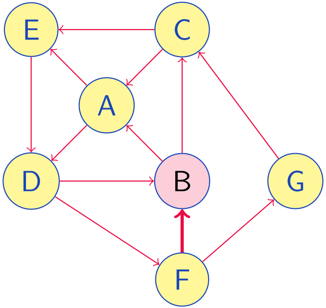
AEDFGCADBCE



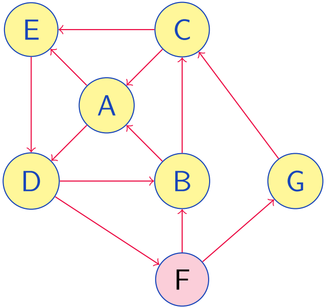
AEDFGCADBCE



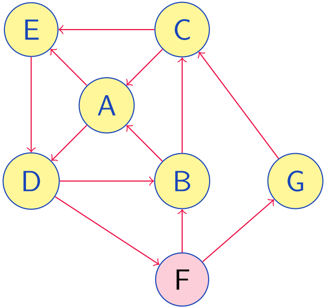
BAEDFGCADBCE



BAEDFGCADBCE



FBAEDFGCADBCE



FBAEDFGCADBCE