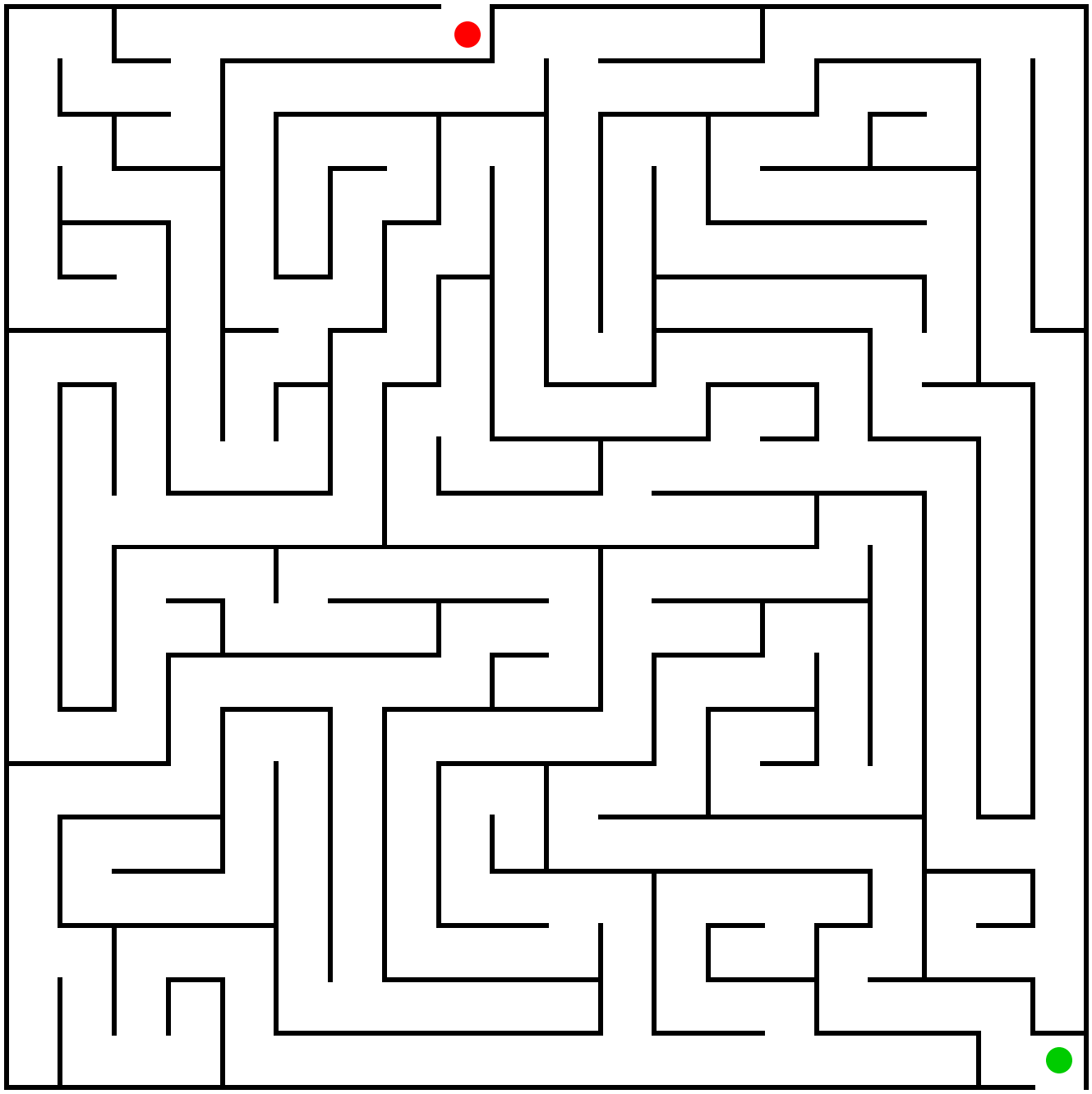
# Algoritmer og Datastrukturer

**Topologisk Sortering, Stærke Sammenhængskomponenter**
**[CLRS, kapitel 22.4-22.5]**



# Dybde Først Søgning (DFS)

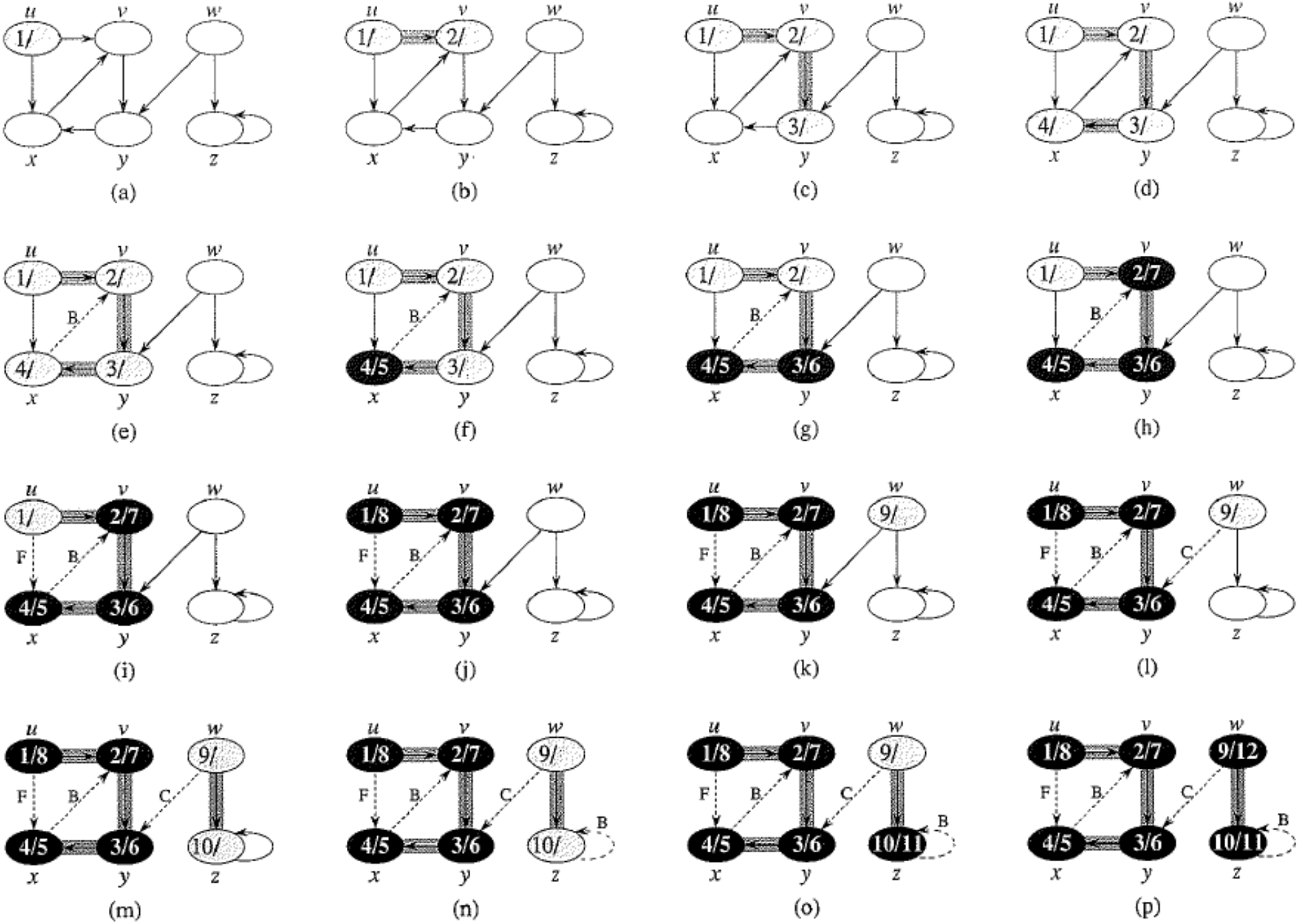
```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)

DFS-VISIT(G, u)
 1  time = time + 1                  // white vertex u has just been discovered
 2  u.d = time
 3  u.color = GRAY
 4  for each v ∈ G.Adj[u]            // explore edge (u, v)
 5      if v.color == WHITE
 6          v.π = u
 7          DFS-VISIT(G, v)
 8  u.color = BLACK                  // blacken u; it is finished
 9  time = time + 1
10  u.f = time
```

*u.color*
WHITE = knuderne endnu ikke besøgt
GRAY = knuder på rekursionsstakken
BLACK = knuderne besøgt

$u.\pi$ = faderen til $u$ i DFS træet

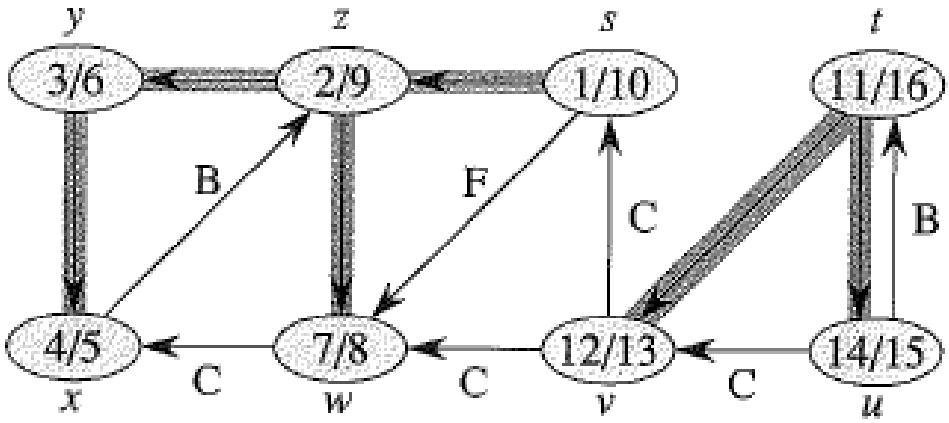
$u.d$ = "discover time" for $u$
$u.f$ = "finishing time" for $u$

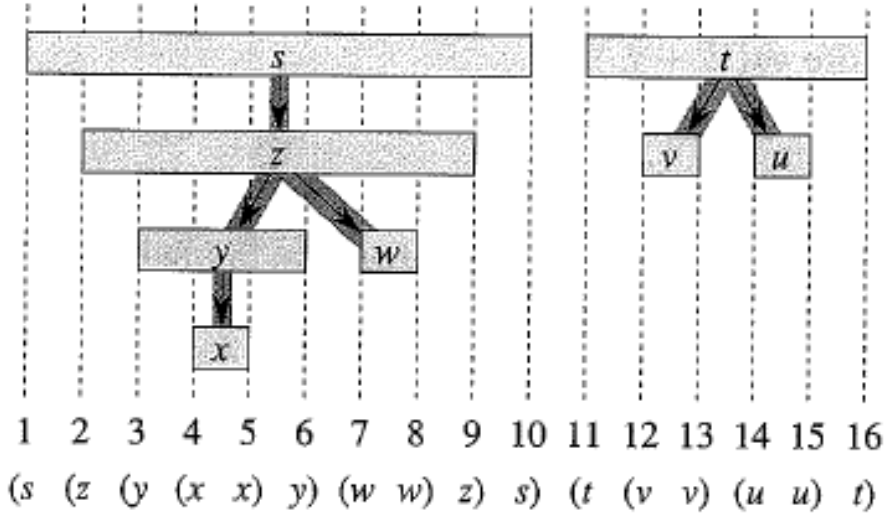
**Tid** $O(n+m)$

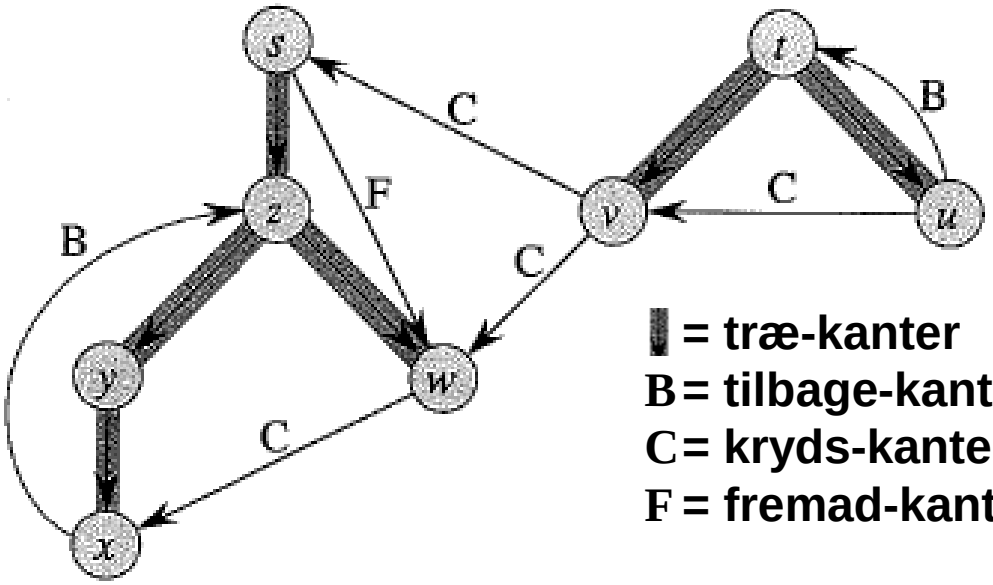

u v w
1/
x y z
(a)
1/ 2/
(b)
1/ 2/ 3/
(c)
1/ 2/ 4/ 3/
(d)
B
(e)
4/5 3/
(f)
4/5 3/6
(g)
2/7
(h)
F
(i)
1/8 2/7 4/5 3/6
(j)
9/
(k)
C
(l)
10/
(m)
(n)
10/11
(o)
9/12
(p)

# Kan en knude have 13/17 som DFS discover/finishing tider ?

a) Ja
b) Nej
c) Ved ikke



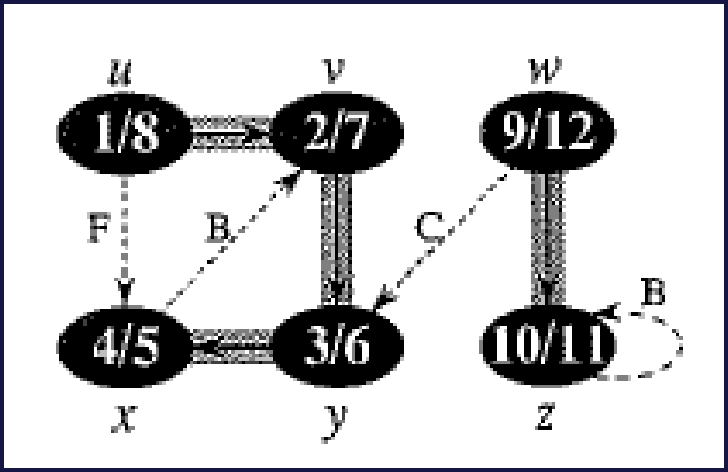




**▮ = træ-kanter**
**B = tilbage-kanter**
**C = kryds-kanter**
**F = fremad-kanter**

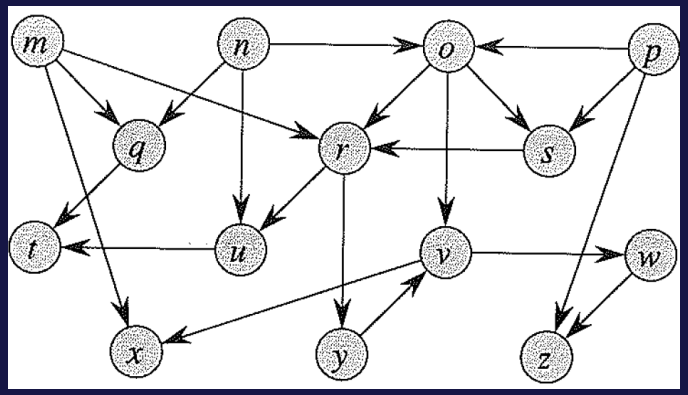
Antag der findes sti $a \rightsquigarrow b$ i en orienteret graf.
Gælder der altid at $a.f > b.f$ ?

a) Ja
b) Nej
c) Ved ikke

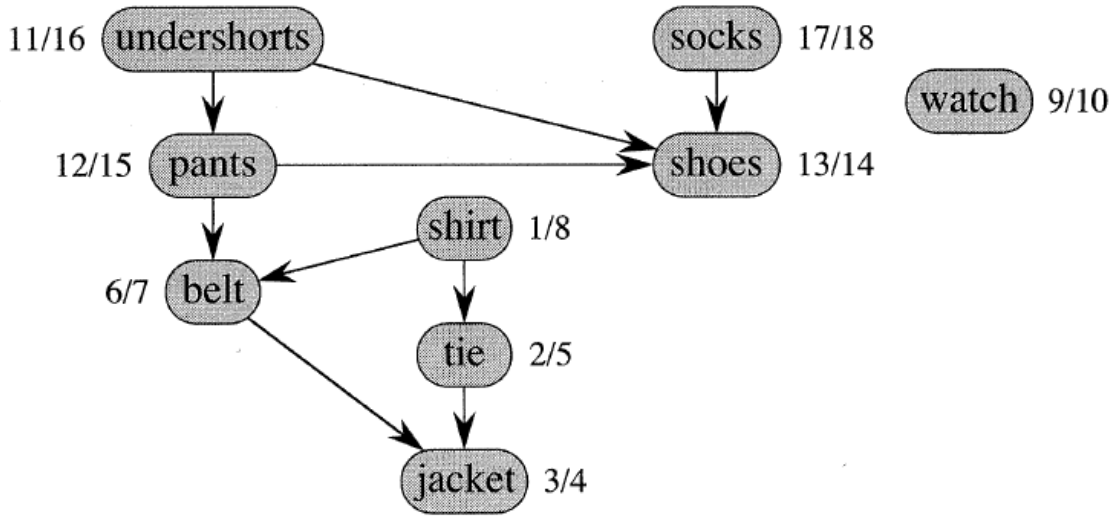


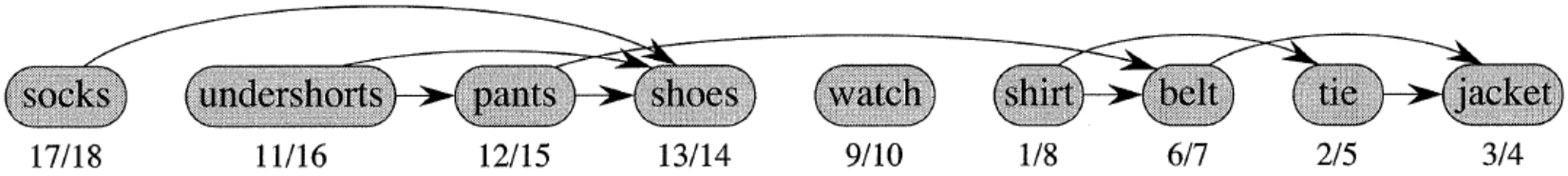
Antag der findes sti $a \rightsquigarrow b$ i en orienteret graf **uden cykler**.
Gælder der altid at $a.f > b.f$ ?

a) Ja
b) Nej
c) Ved ikke

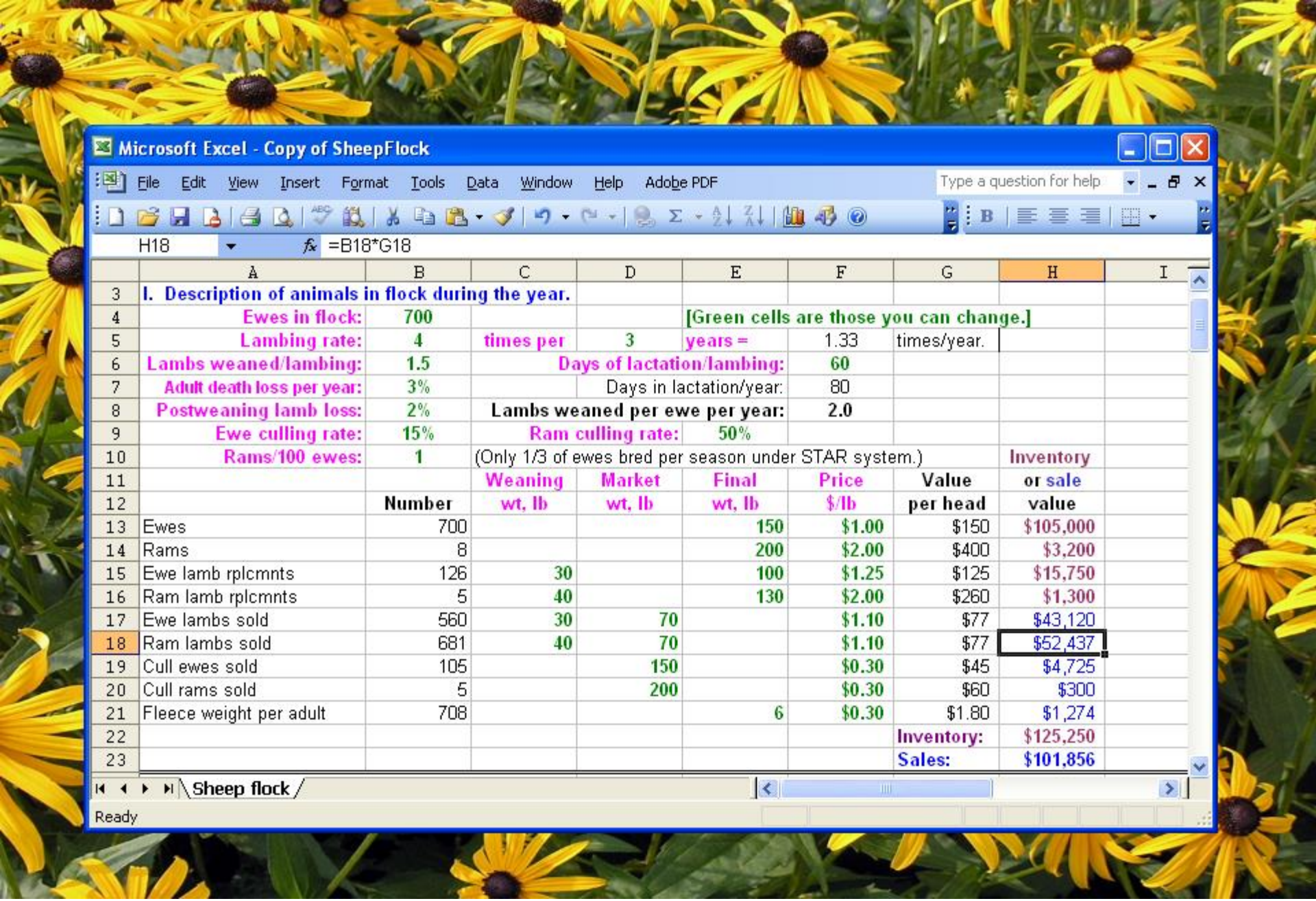


# Acykliske Grafer: Topologisk Sortering



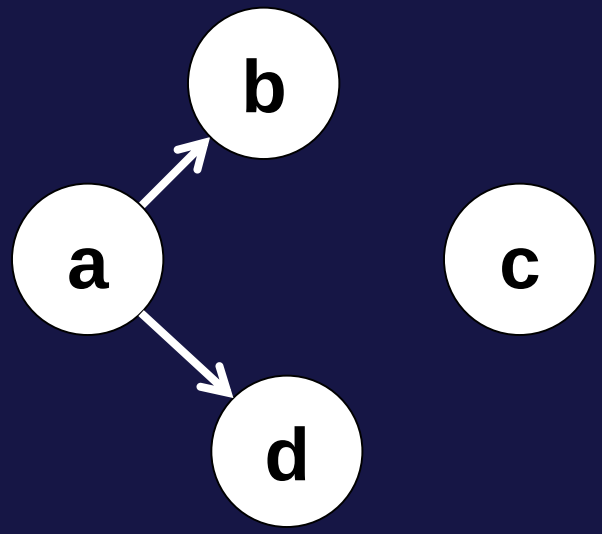

**Alle kanter går fra venstre-mod-højre**




**Topologisk sortering = en rækkefølge hvor vi kan beregne cellernes indhold**

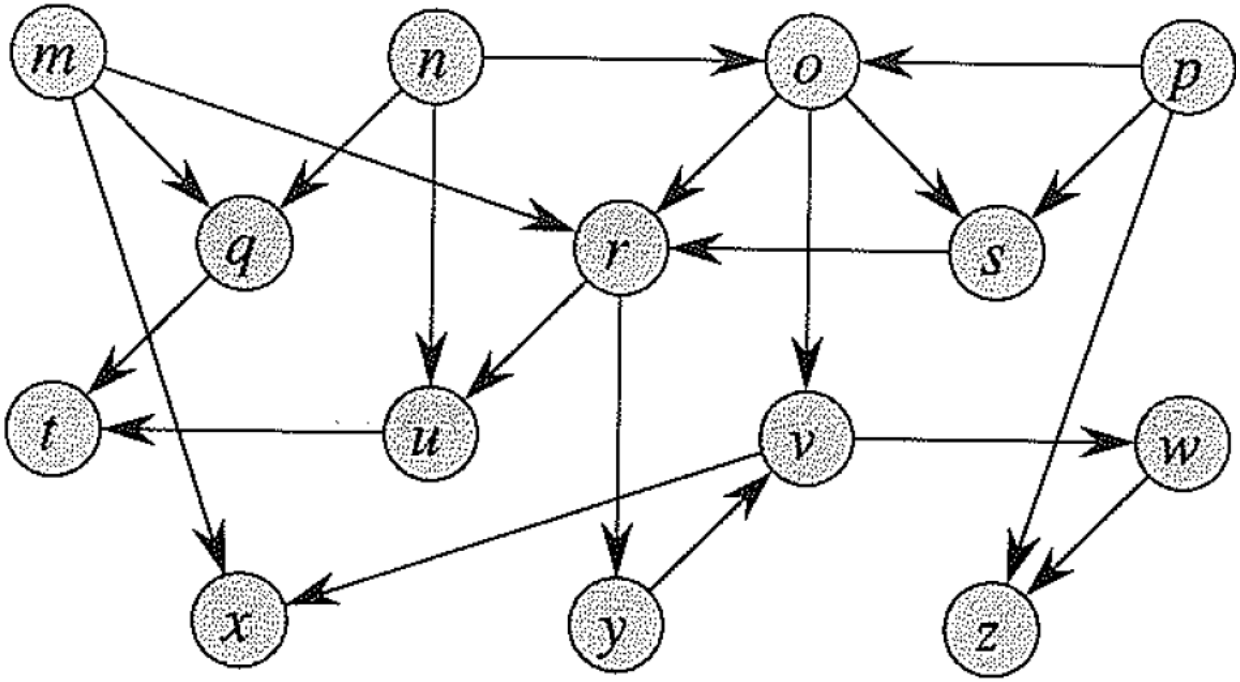
# Hvor mange mulige måder kan man topologisk sortere grafen ?

a) 1
b) 2
c) 3
d) 4
e) 5
f) 6
g) 7
h) 8
i) 9
j) Ved ikke



Kahn 1962

# Topologisk Sortering (I)

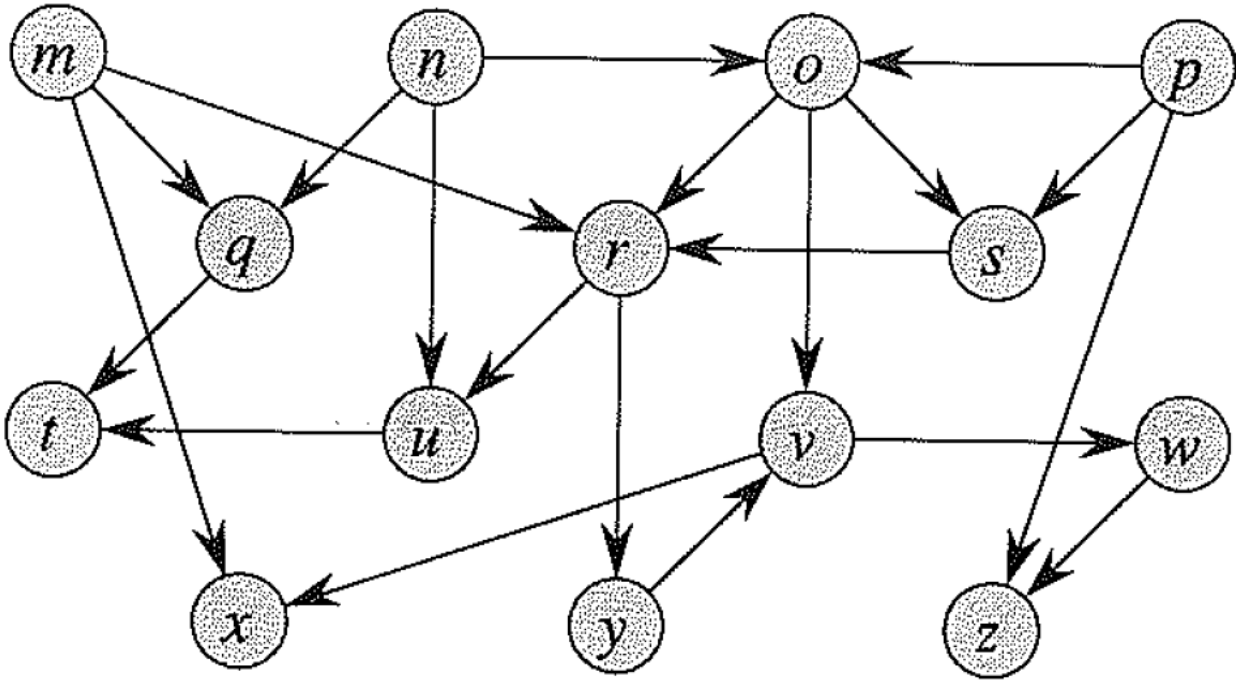



**Algoritme**: Grådigt slet en knude med indgrad 0 (og udgående kanter), og tilføj knuden sidst i den topologiske orden

**Tid** $O(m+n)$

Tarjan 1976

# Topologisk Sortering (II)




TOPOLOGICAL-SORT($G$)

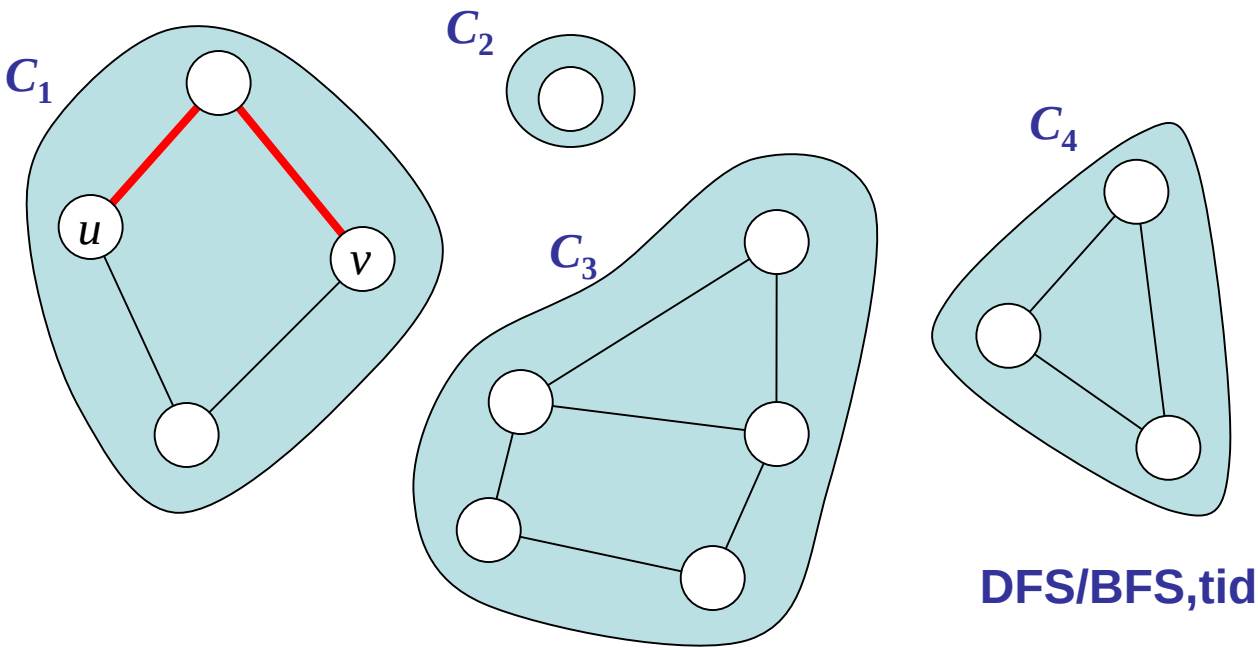
1 call DFS($G$) to compute finishing times $v.f$ for each vertex $v$
2 as each vertex is finished, insert it onto the front of a linked list
3 **return** the linked list of vertices

**Tid** O($m$+$n$)

# Sammenhængskomponenter

Opdeling af knuderne i en uorienteret graf i **komponenter $C_1,\ldots,C_k$**, således at

$u$ og $v$ er i $C_i$ hvis og kun hvis der er en sti mellem $u$ og $v$



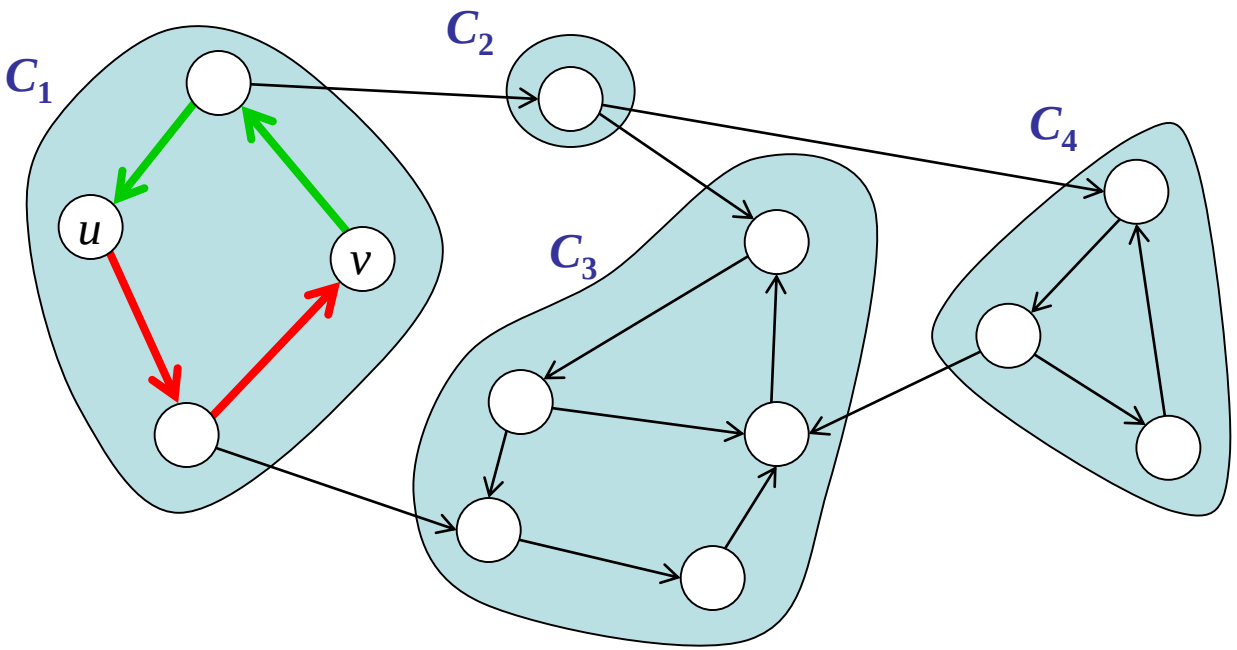

**DFS/BFS,tid** $O(m+n)$

# Stærke Sammenhængskomponenter

Opdeling af knuderne i en orienteret graf i **komponenter $C_1,\ldots,C_k$**, således at

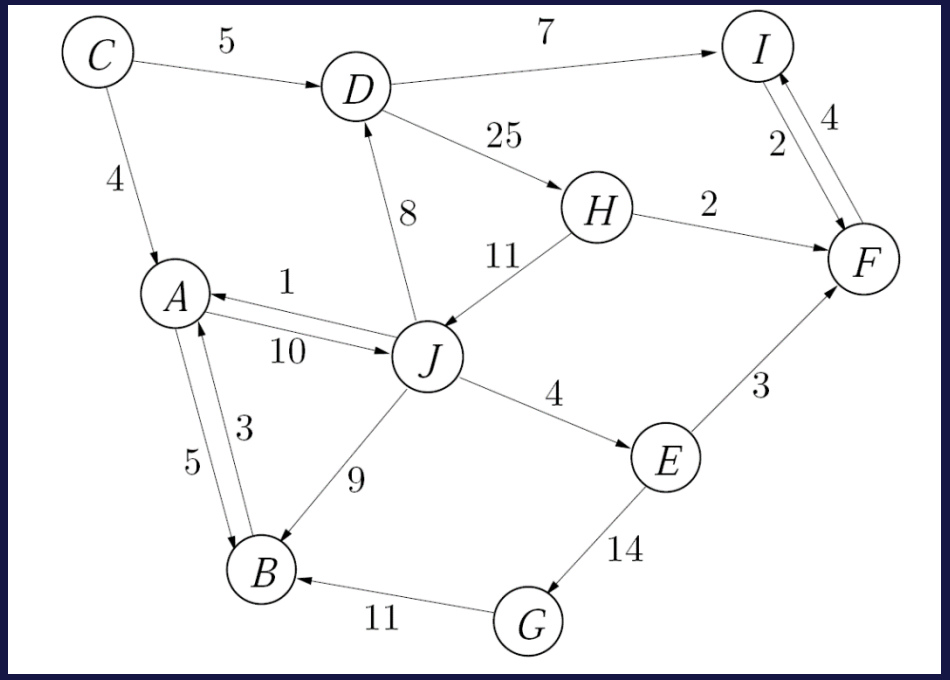
$u$ og $v$ er i $C_i$ hvis og kun hvis der både er

- en sti fra $u$ til $v$ og
- en sti fra $v$ til $u$



# Antal stærke sammenhængskomponenter ?

a) 1
b) 2
c) 3
d) 4
e) 5
f) Ved ikke




Eksamensopgave 1(d), sommeren 2009

{A,B,D,E,G,H,J}, {C}, {F, I}

Tarjan 1972

# Stærke Sammenhængskomponenter

STRONGLY-CONNECTED-COMPONENTS($G$)

1 call DFS($G$) to compute finishing times $u.f$ for each vertex $u$
2 compute $G^T$
3 call DFS($G^T$), but in the main loop of DFS, consider the vertices in order of decreasing $u.f$ (as computed in line 1)
4 output the vertices of each tree in the depth-first forest formed in line 3 as separate strongly connected component

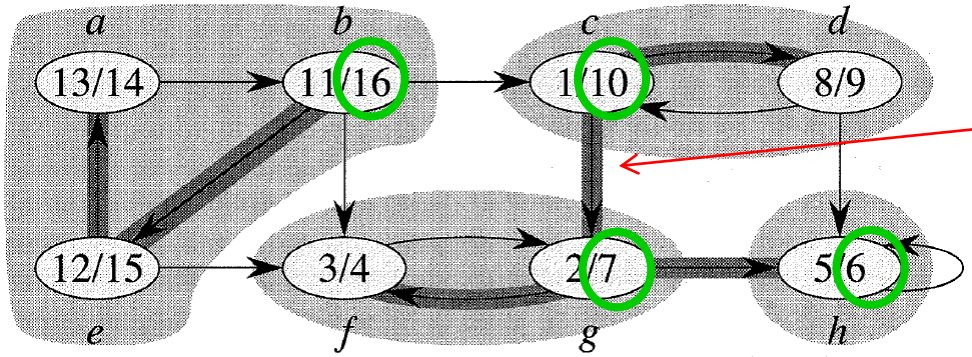
```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)
```

```
DFS-VISIT(G, u)
 1  time = time + 1            // white vertex u has just been discovered
 2  u.d = time
 3  u.color = GRAY
 4  for each v ∈ G.Adj[u]      // explore edge (u, v)
 5      if v.color == WHITE
 6          v.π = u
 7          DFS-VISIT(G, v)
 8  u.color = BLACK            // blacken u; it is finished
 9  time = time + 1
10  u.f = time
```

**Tid** $O(m+n)$

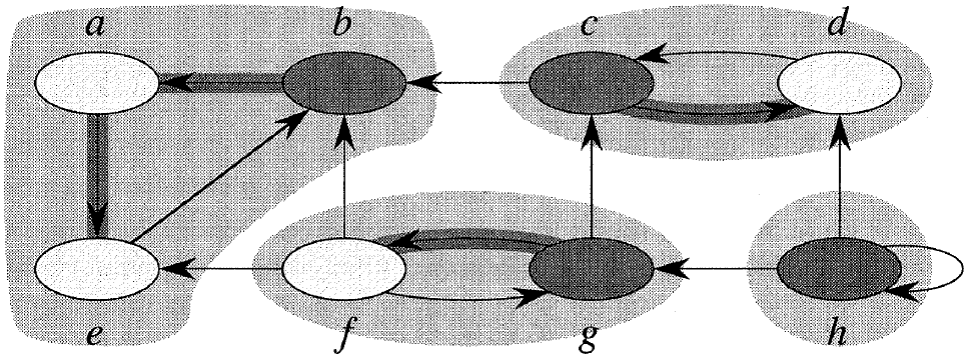
# Stærke Sammenhængskomponenter

$G$

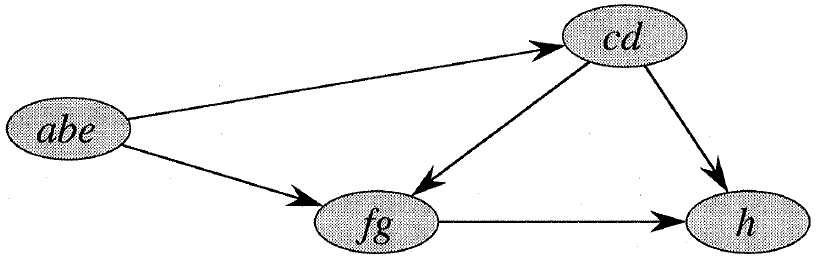



DFS trækanter mellem to stærke sammenhængs-komponenter

$G^T$




De største finising tider i hver komponent udgør en (omvendt) topologisk sortering af komponenterne




**Tid** $O(m+n)$