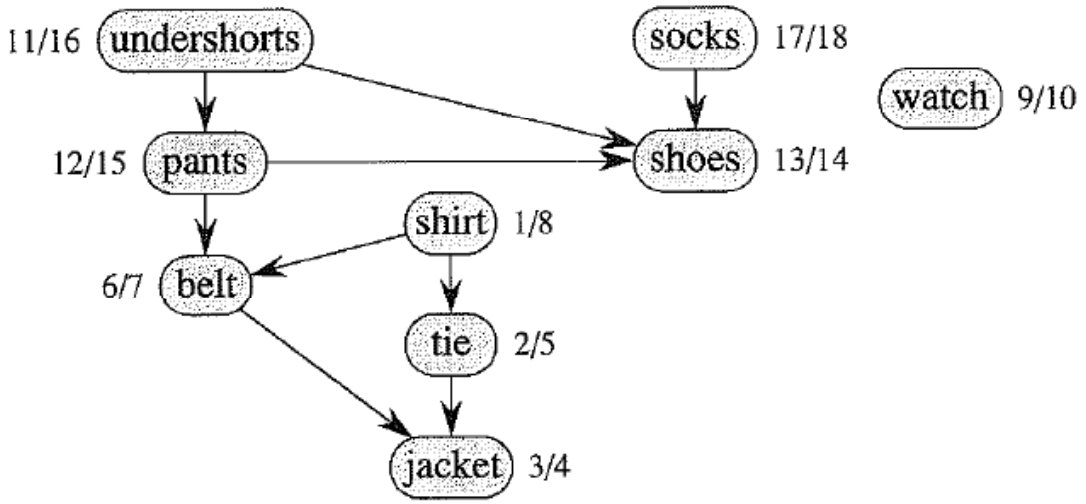
Algoritmer og Datastrukturer 2

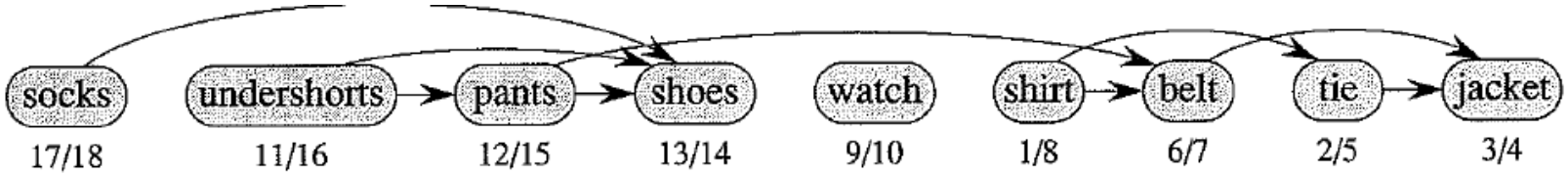
Gerth Stølting Brodal

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

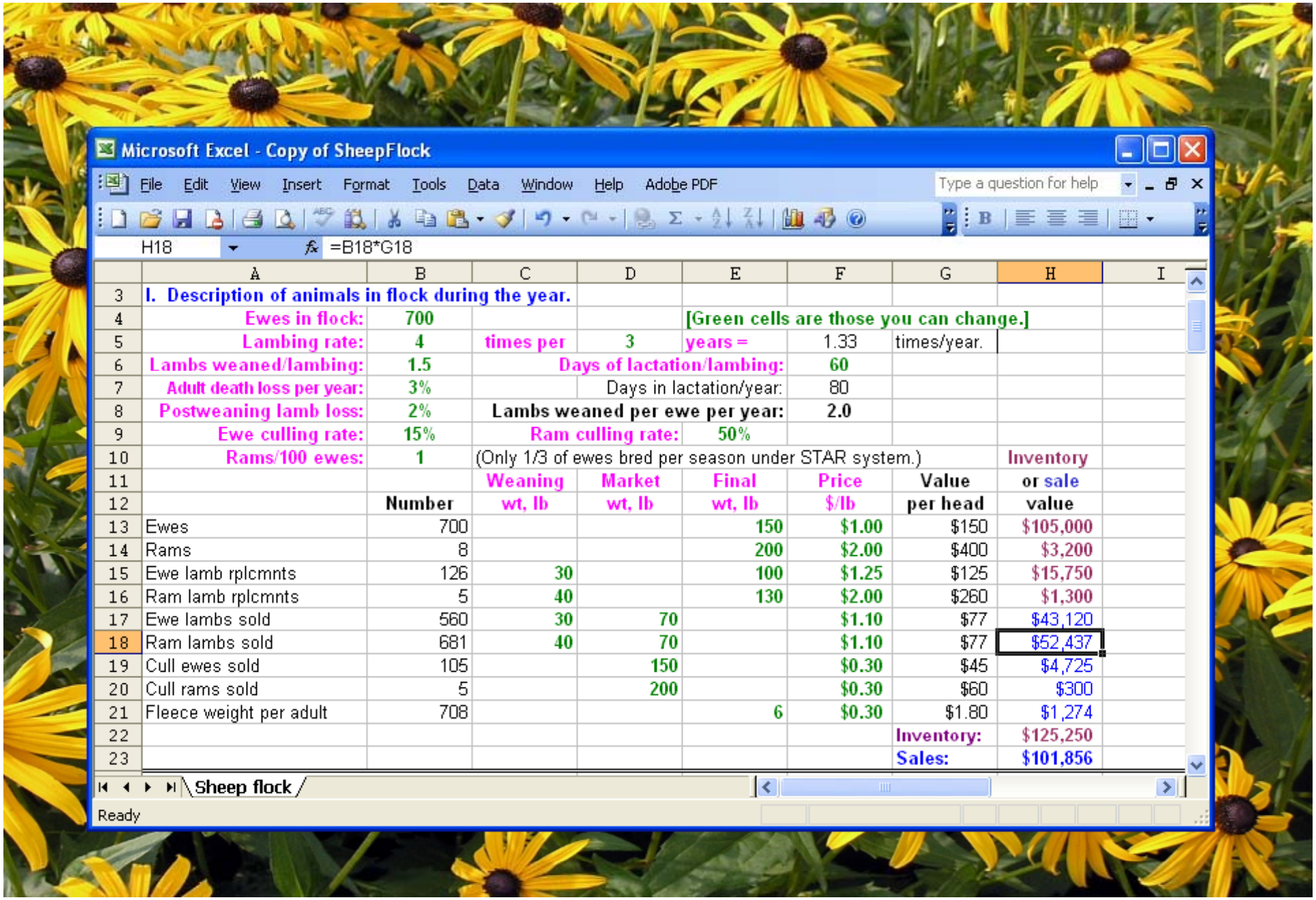
AARHUS UNIVERSITET

Acykliske Grafer: Topologisk Sortering



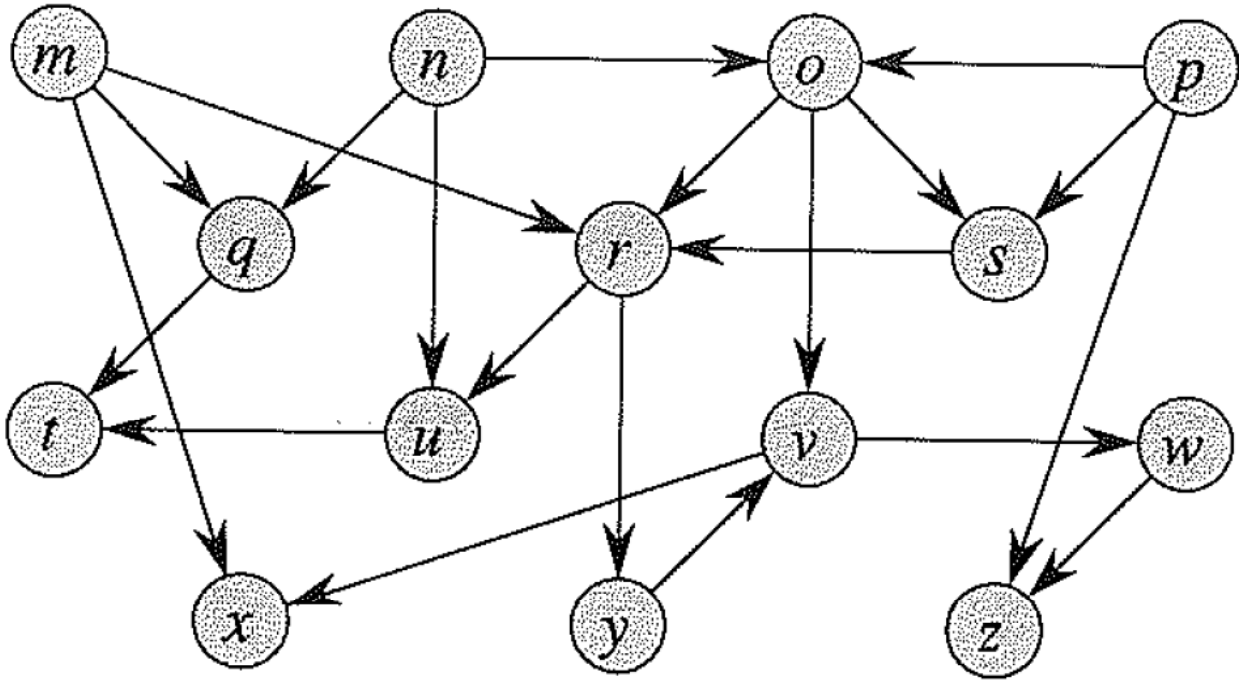


Alle kanter går fra venstre-mod-højre



Topologisk sortering = en rækkefølge hvor vi kan beregne cellernes indhold

Topologisk Sortering



TOPOLOGICAL-SORT(G)

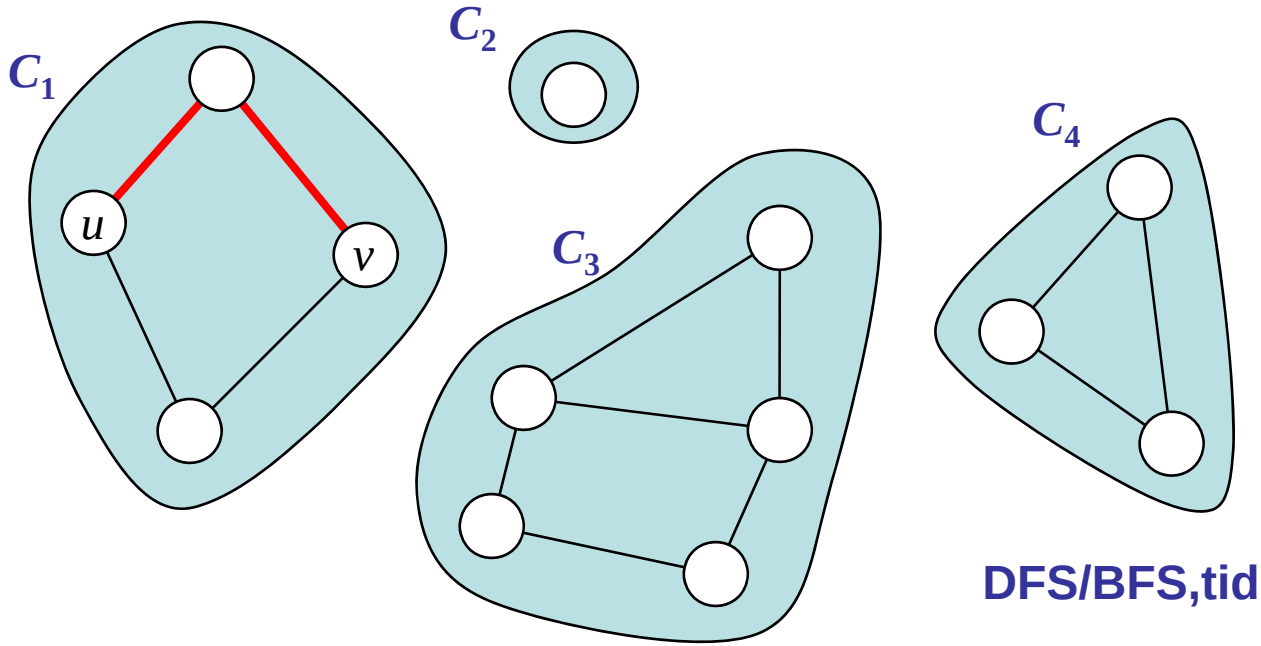
1 call DFS(G) to compute finishing times $f[v]$ for each vertex v
2 as each vertex is finished, insert it onto the front of a linked list
3 **return** the linked list of vertices

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



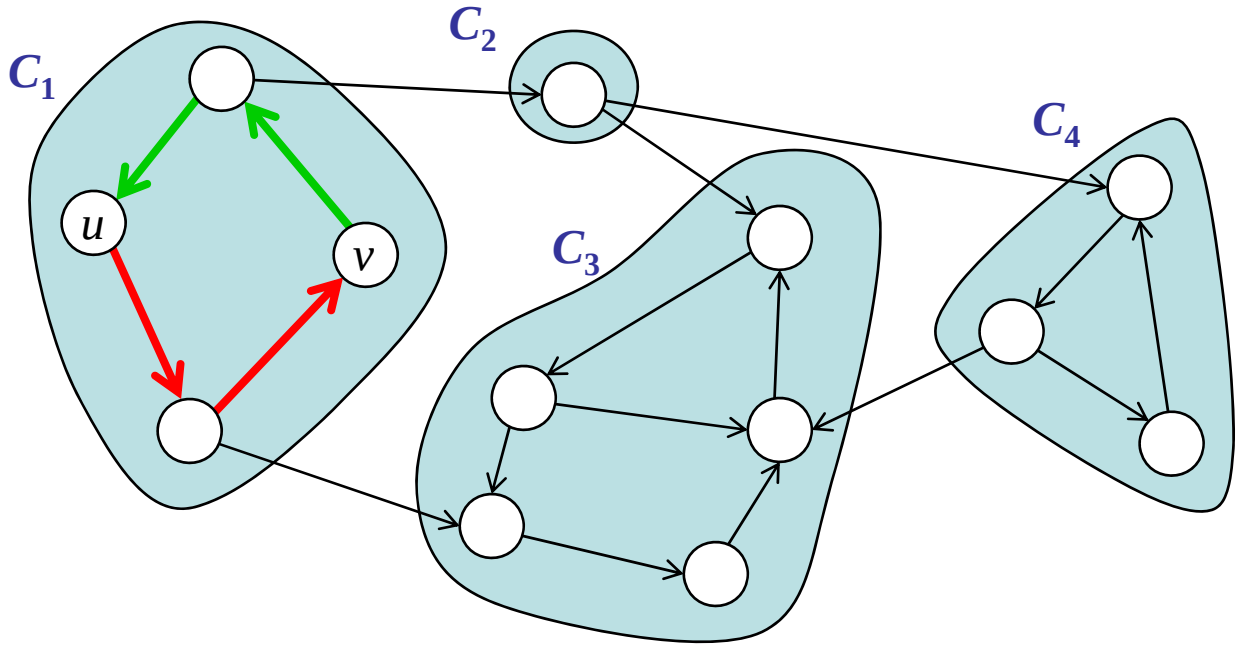
DFS/BFS,tid $O(m+n)$

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



Stærke Sammenhængskomponenter

STRONGLY-CONNECTED-COMPONENTS(G)

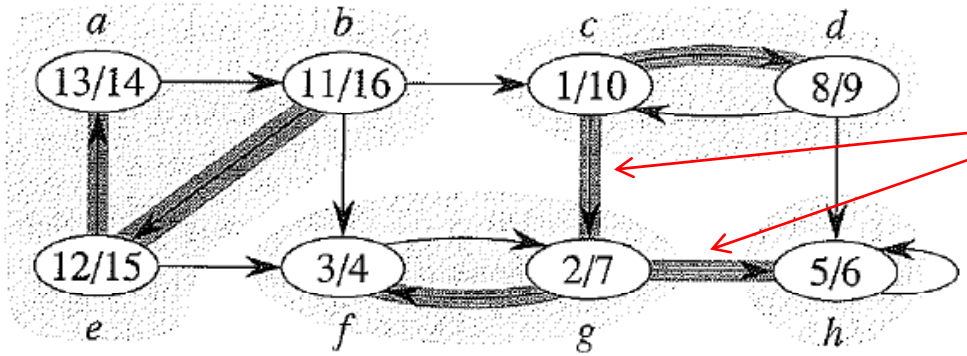
1 call DFS(G) to compute finishing times $f[u]$ for each vertex u
2 compute G^T
3 call DFS(G^T), but in the main loop of DFS, consider the vertices in order of decreasing $f[u]$ (as computed in line 1)
4 output the vertices of each tree in the depth-first forest formed in line 3 as a separate strongly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

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

DFS-VISIT(u)
1  color[u] ← GRAY        ▷ White vertex u has just been discovered.
2  time ← time + 1
3  d[u] ← time
4  for each v ∈ Adj[u]    ▷ Explore edge (u, v).
5       do if color[v] = WHITE
6             then π[v] ← u
7                  DFS-VISIT(v)
8  color[u] ← BLACK       ▷ Blacken u; it is finished.
9  f[u] ← time ← time + 1
```

Tid $O(m+n)$

Stærke Sammenhængskomponenter



DFS trækanter mellem to stærke sammenhængs-komponenter