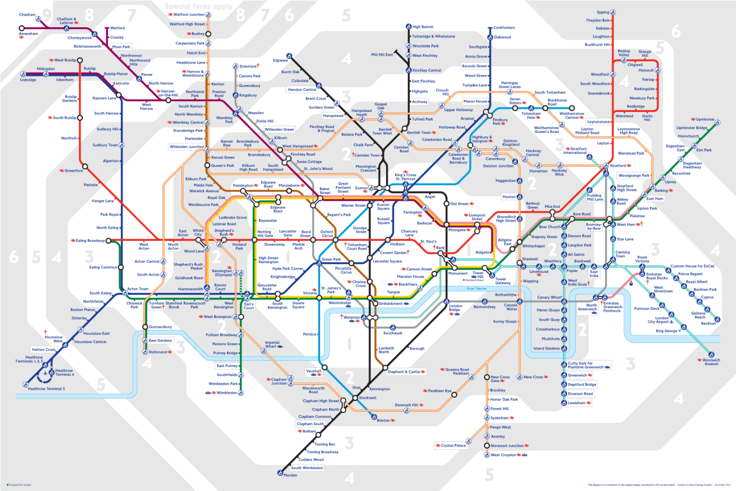
Rettede grafer

- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt sammenhængende komponenter
- Implicitte grafer

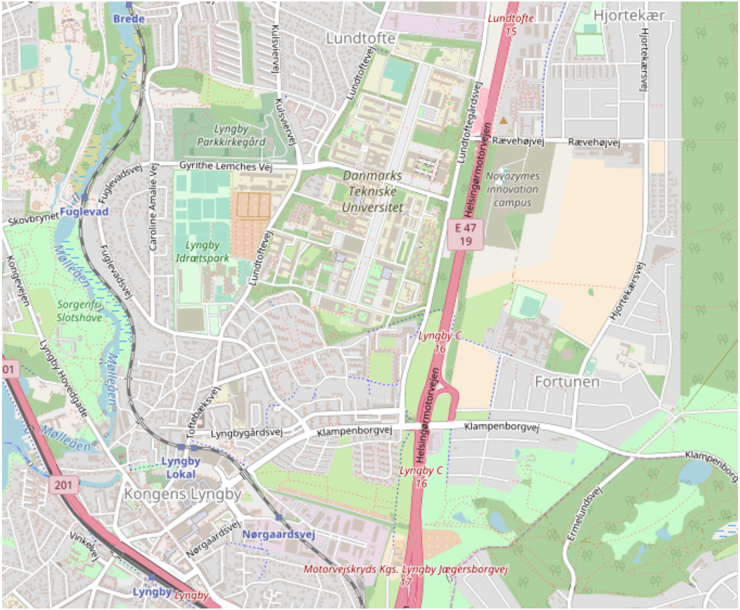
Rettede grafer

- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt sammenhængende komponenter
- Implicitte grafer

U-rettet graf eksempel: Transport



Eksempel: Vejnet – en graf?



Eksempel: Vejnet – en graf?

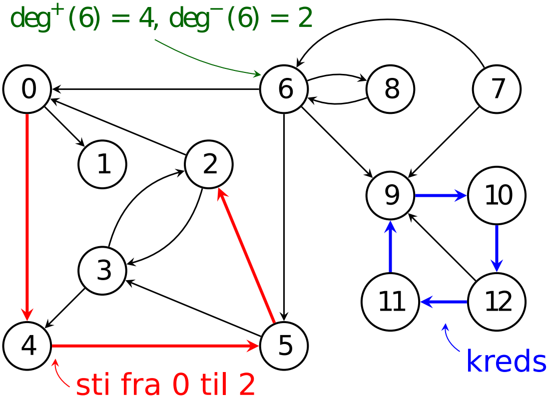
Eksempel: Vejnet – en graf?

Nogen veje er *ensrettede* . Dette modelleres ved en rettet graf.

Rettet graf

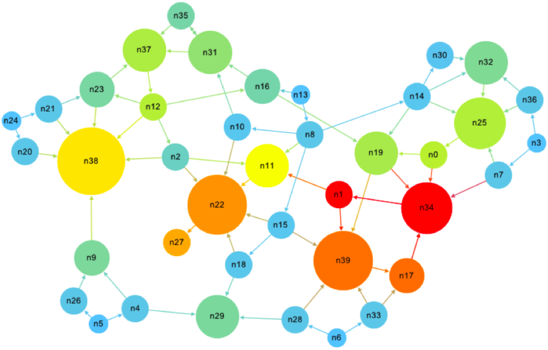
Definition (rettet graf)

Mængde af knuder, parvist forbundet af *rettede* kanter.



Anvendelse: WWW

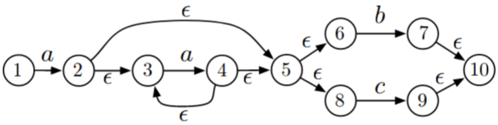
- Knude: Hjemmeside. Kant: Hyperlink.
- Webcrawling. Page rank.



http://computationalculture.net/what_is_in_pagerank/

Anvendelse: Automater, regulære udtryk

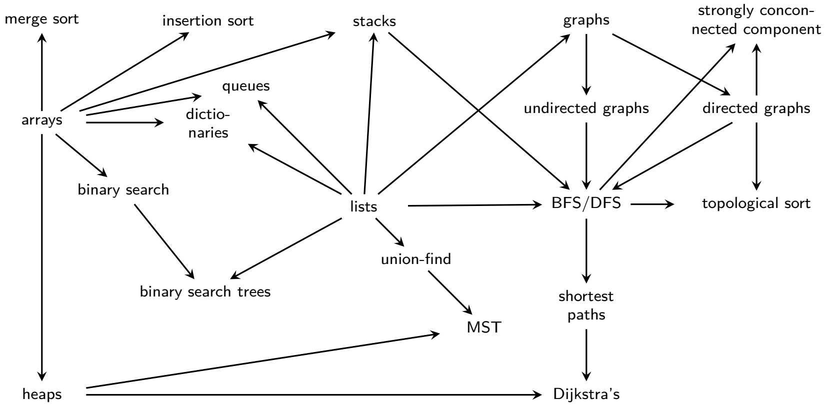
- Knude: Tilstand. Kant: Transition.
- Denne automat accepterer "aaab" $\Leftrightarrow$ Der er en vandring fra knude 1 til knude 10, der svarer til strengen "aaab"
- Regulære udtryk kan repræsenteres ved automater.



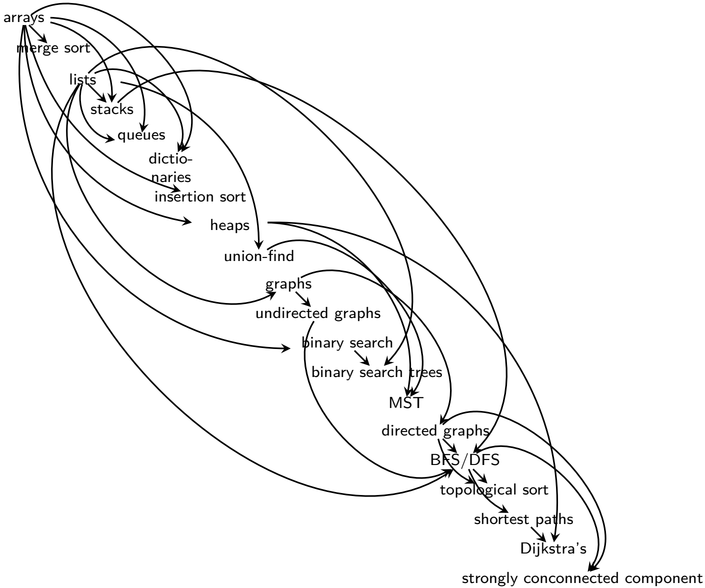
R = a·(a*)·(b|c)

Anvendelse: Afhængigheder

- Knude: emne. Kant: afhængighed.
- Er der nogen cykliske afhængigheder? Kan vi finde en rækkefølge af emnerne, så vi undgår cykliske afhægigheder?

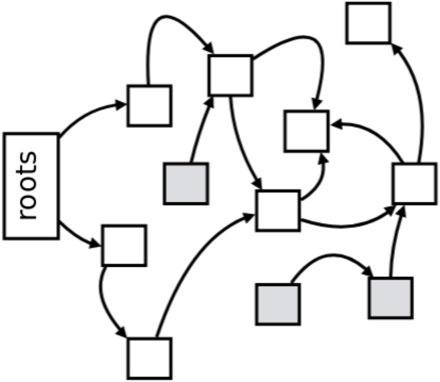


Anvendelse: Afhængigheder



Anvendelse: Garbage Collection

Knude: Objekt. Kant: Reference.



Anvendelser

graf	knuder	kanter
internet	hjemmeside	hyperlink
transport	vejkryds	(ensrettet) vej
skedulering	job	rækkefølge
smittespredning	person/dyr	smitte
citationsnetværk	artikel	citation
objektgraf	objekter	pointere
objekthierarch	klasse	nedarvning

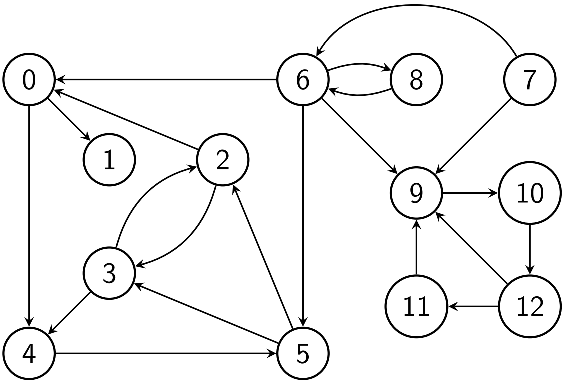
Rettede grafer

Lemma

$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = m$

Bevis.

Hver kant går fra præcis én knude til én knude. □



Algoritmiske problemer på rettede grafer

Tilgængelighed. Er der en sti *fra s til t*?

Korteste vej. Hvad er den korteste vej fra s til t?

Rettet kreds. Indeholder grafen en rettet kreds?

Topologiske sortering Kan vi ordne knuderne, så alle kanter peger fra en lavere-ordensknude til en højere-ordensknude?

Stærk sammenhæng. Kan man komme fra a til b for alle knuder a og b i grafen?

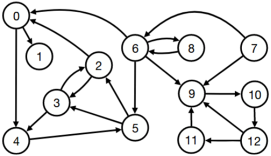
Transitiv afslutning. Enhver sti i G repræsenteres ved en kant i den transitive afslutning af G.

Rettede grafer

- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt sammenhængende komponenter
- Implicitte grafer

Repræsentation

- G er en rettet graf med n knuder og m kanter
- Repræsentation. Vi har brug for følgende metoder:
  - `PointsTo(u,v)`: Kant fra u til v?
  - `Neighbours(v)`: Returnerer alle knuder, der peger ud af v. (Dvs. alle vs *ud-naboer*.)
  - `Insert(v,u)`: Tilføj kanten (v,u) til G (medmindre den allerede findes.)

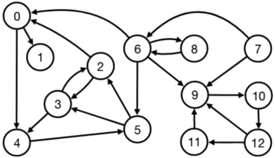


Incidensmatrix

Rettet graf G med n knuder og m kanter.

Incidensmatrix:

- $n \times n$ matrix A
- $A[i,j] = 1$ når $i \to j$, ellers 0.



	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0
5	0	0	1	1	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	1	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	0	0	1	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	1	0	1	0

`PointsTo(u,v)`

`PointsTo(u,v)` `PointsTo(6,0)?`

`PointsTo(u,v)` `PointsTo(0,6)?`

`Neighbours(v)`

`Neighbours(v)` `Neighbours(7)?`

`Insert(v,u)`

`Insert(v,u)` `Insert(8,4)?`

Incidensmatrix

Rettet graf G med n knuder og m kanter.
Incidensmatrix:

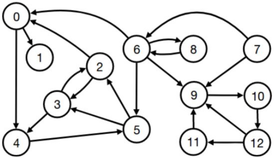
- $n \times n$ matrix A
- $A[i,j] = 1$ when $i \to j$, else 0.

Plads : ? $O(n^2)$.
Tid:

`PointsTo(u,v)` : ? $O(1)$ tid.

`Neighbours(v)` : ? $O(n)$ tid.

`Insert(v,u)` : ? $O(1)$ tid.

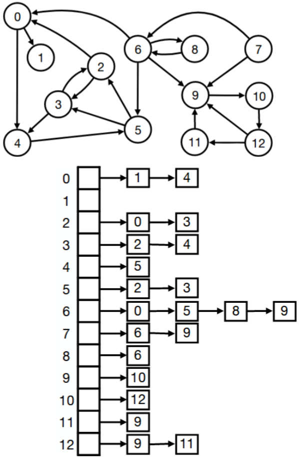


	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0
5	0	0	1	1	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	1	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	0	0	1	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	1	0	1	0

Incidensliste

Rettet graf G med n knuder og m kanter.
Incidensliste:

- Array $A[0 \dots n-1]$
- $A[i]$ indeholder en liste af alle knuder, som knude i peger på.



`PointsTo(u,v)`

`Neighbours(v)`

`Insert(v,u)`

Incidensliste

Rettet graf G med n knuder og m kanter.
Incidensliste:

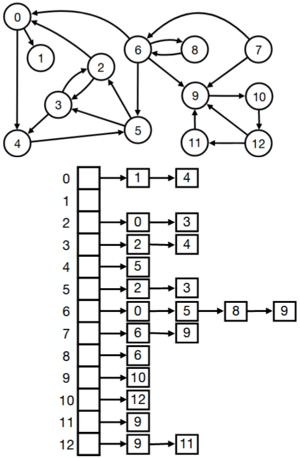
- Array $A[0 \ldots n-1]$
- $A[i]$ indeholder en liste over alle knuder som i peger på.

Plads : $O(n + \sum_{v \in V} \deg^+(v)) = O(n+m)$.
Tid

`PointsTo(u,v)`: $O(\deg^+(u))$ tid.

`Neighbours(v)`: $O(\deg^+(v))$ tid.

`Insert(v,u)`: $O(\deg^+(v))$ tid.



Repræsentation

Datastruktur	`PointsTo`	`Neighbours`	`Insert`	Plads
Incidensmatrix	$O(1)$	$O(n)$	$O(1)$	$O(n^2)$
Incidensliste	$O(\deg^+(v))$	$O(\deg^+(v))$	$O(\deg^+(v))$	$O(n+m)$

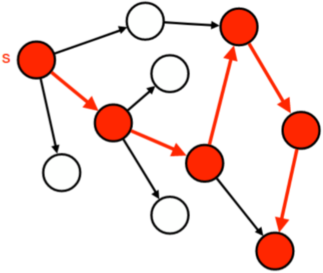
Rettede grafer

- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt sammenhængende komponenter
- Implicitte grafer

Dybdeførstsøgning / breddeførstsøgning

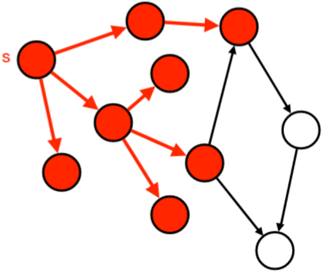
Dybdeførstsøgning

- Alle knuder starter som umarkerede. Besøg s.
- At besøge knuden v:
  - Marker v,
  - Besøg rekursivt alle vs umarkerede udnaboer.



Breddeførstsøgning

- Alle knuder starter umarkerede.
- Marker s og tilføj s til køen Q.
- Så længe Q ikke er tom:
  - Udtag køens ældste element, knuden v.
  - For alle udnaboer u til v $(v \to u)$:
    - Marker u, sæt u i kø.



Tid $O(n+m)$

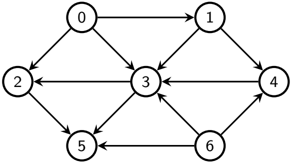
Rettede grafer

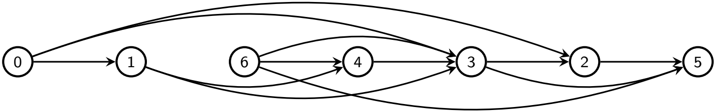
- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt sammenhængende komponenter
- Implicitte grafer

Topologisk sortering og rettede acykliske grafer

Acyklisk rettet graf. Indeholder ikke en kreds.

Topologisk sortering. En ordning af knuderne, så alle kanter går fra højre til venstre.





Algoritmiske problemer

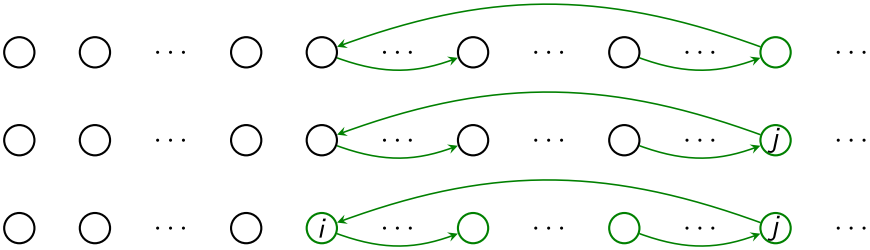
- Afgør om grafen G er acyklisk.
- Returner en topologisk sortering (hvis ja).

Mål: Vise, at G acyklisk $\Leftrightarrow$ G har en topologisk sortering.

Give en algoritme, der løser begge problemer.

Topologisk sortering og acykliske grafer

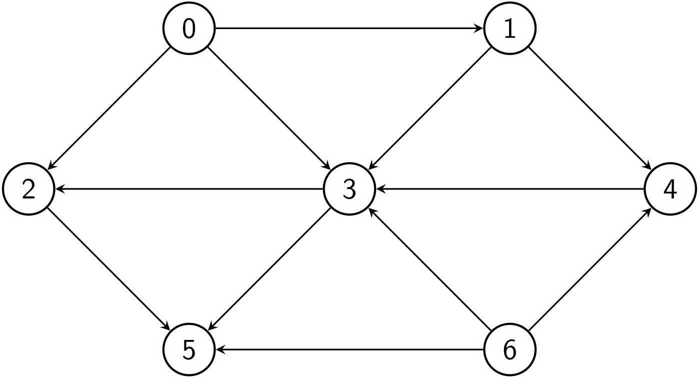
Lemma G har en topologisk sortering $\Rightarrow$ G is er acyklisk.
Bevis. Antag, G har en topologisk sortering.



Hvis G ikke er acyklisk, så har den en kreds, $K = v_{k_0}$.
Lad j være den knude i K som er *længst* til højre.
Der er en kant $j \rightarrow i$ i K, og dermed i G.
Men i er før j $\Rightarrow$ ikke en topologisk sortering.

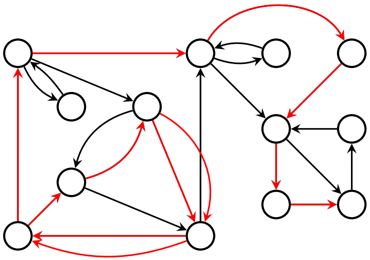
Topologisk sortering og rettede acykliske grafer - Øvelse

Kom frem til en strategi for at finde en topologisk sortering af en given acyklisk rettet graf.



Topologisk sortering og acykliske rettede grafer

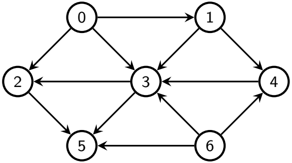
Lemma. G er en acyklisk rettet graf $\Rightarrow$ G har en knude v med $\deg^{-}(v) = 0$,
dvs, *indgrad* 0. Ingen andre knuder peger på v. (v er en *kilde*)

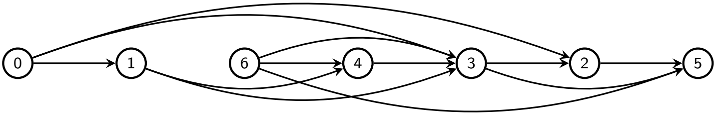


Bevis. Antag enhver knude v har indgrad ≥ 1.
Start i en vilkårlig knude s og gå "baglæns"i $n + 1$ skridt.
Siden alle knuder har en indkant, findes en sådan vandring.
Der er kun n knuder i G, så mindst én knude er besøgt to gange.
Vi har fundet en kreds. G er ikke acyklisk.

Topologisk sortering og acykliske rettede grafer

Lemma G er acyklisk $\Rightarrow$ G har en topologisk sortering





Bevis ved induktion over antal knuder i G.

- (Induktionsstart) Hvis der kun er en knude, – allerede sorteret.
- (Induktionsskridt)
  - Find en knude v med $\deg^{-}(v) = 0$.
  - $G - v$ er stadigvæk acyklisk. $G - v$ har en topologisk sortering.
  - Placer v længst til venstre, efterfulgt af sorteringen af $G - v$. Dette er en topologisk sortering, da ingen kanter går ind i v.

Topologisk sortering – Implementation

Mål Effektiv algoritme givet incidenslisterepræsentation.

Algorithm Baseret på beviset:

if $G = (\{v\}, \emptyset)$ **then**
  print v.
else
  find v med $\deg^-(v) = 0$
  print v
  TopSort$(G - v)$
end if

Korrekthed Følger af beviset.

Tid Gentag ind til alle pånær en knude er fjernet: n gange.

- Find en knude med indgrad 0 Hvor lang tid tager dette?
- Fjern den fra grafen. Hver kant er fjernet netop en gang $\Rightarrow$ Samlet tid $O(m)$ for dette skridt.

Topologisk sortering – Implementation 0 (dum)

Løsning 0 Vedligehold incidensmatrix.

if $G = (\{v\}, \emptyset)$ **then**
  print v.
else
  find v with $\deg^{-}(v) = 0$
  print v
  TopSort$(G - v)$
end if

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0
5	0	0	1	1	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	1	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	0	0	1	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0	1	0	1	0

En knude med indgrad 0 har en søjle med kun 0'er.
Fjern knuden ved at erstatte dens række med lutter 0'er.
Tid: Gentag ind til alle pånær en knude er fjernet:

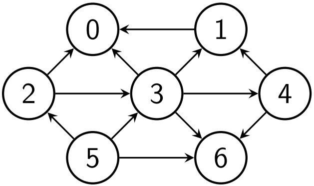
- Find en knude af indgrad 0 $O(n^2)$ tid
- Fjern den fra fra grafen $O(1)$ tid per knude.

I alt $O(n^3)$.

Topologisk sortering – Implementation 1 (ikke smart)

Løsning 1 Byg den *omvendte* graf G^R:

if $G = (\{v\}, \emptyset)$ **then**
  print v.
else
  find v with $\deg^-(v) = 0$
  print v
  TopSort($G - v$)
end if



Gennemsøg G^R (lineært) efter en knude med udgrad 0.

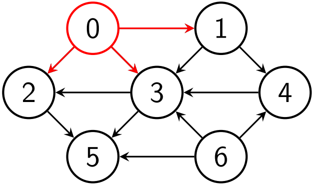
Tid Gentag indtil alle pånær en knude er fjernet:

- Find en knude med indgrad 0 $O(n)$ time
- Fjern den fra grafen Hver kant bliver fjernet
  Netop en gang $\Rightarrow$ I alt $O(m)$ tid på dette trin.

I alt $O(n^2 + m) = O(n^2)$.

Topologisk sortering – Implementation 2 (smart)

Løsning 2 Vedligehold information om indgrad af alle knuder. Hav en liste over knuder med indgrad 0. (Hægtet liste.)



$(v, \deg^-(v))$ table

0	0
1	1
2	2
3	4
4	2
5	3
6	0

0-deg$^-$ list

$\boxed{0} \rightarrow \boxed{6}$

Initialising $O(n + m)$ time.

Gentag indtil alle pånær en knude er fjernet: n gange.

- Find en knude af indgrad 0 $O(1)$ time
- Fjern den fra grafen Hver kant fjernes netop een gang $\Rightarrow$ Samlet tid $O(m)$ på dette trin.

I alt: $O(n + m)$.

Topologisk sortering og acykliske rettede grafer

Lemma

En rettet graf G er acyklisk $\Leftrightarrow$ G har en topologisk sortering.

Theorem

Derer en algoritme, der i $O(n+m)$ tid afgør om G er acyklisk, og, hvis ja, returnerer en topologisk sortering.

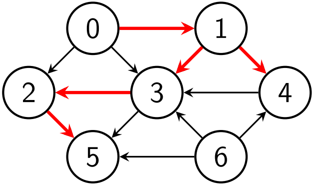
Topologisk sortering ved hjælp af dybdesøgning

Ide:

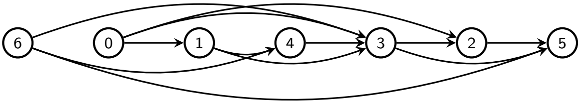
- Kør dybdeførstsøgning på G.
- Når vi returnerer fra vs rekursive kald, læg v på en stak
- Print stakken.

Tid $O(m + n)$

Intuition Finder rekursivt knuder af udgrad 0.



6 0 1 4 3 2 5



Rettede grafer

- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt sammenhængende komponenter
- Implicitte grafer

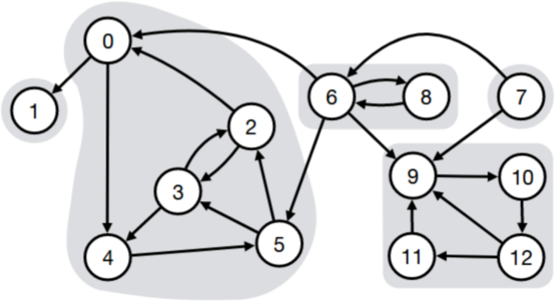
Stærkt sammenhængende komponenter

Definition (Stærkt sammenhængende)

u og v er *stærkt sammenhængende* hvis der både findes en sti $u \to v$ og en sti $v \to u$.

Definition (Stærkt sammenhængende komponent)

Maksimal delmængde af stærkt sammenhængende knuder.



Stærkt sammenhængende komponenter via dybdesøgning

Ide

- Kør dybdesøgning på den omvendte graf G^R.
  Noter alle knuders sluttider.
- Kør dybdesøgning på G, men start hver "runde" med den umarkerede knude med den seneste sluttid.
- Hver runde finder og markerer en stærkt sammenhængende komponent.

Korrekthed Se kapitel 22.5 i CLRS.
Tid $O(n + m)$

Rettede grafer

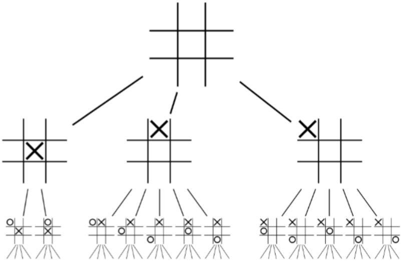
- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt-sammenhængende komponenter
- Implicitte grafer

Implicit grafrepræsentation

Implicit graf. Rettet eller urettet graf givet ved en *implicit repræsentation*:

- startknude s
- algoritme, der *genererer* naboerne til en knude.

Anvendelser Spil, kunstig intelligens, logisk modellering, ...

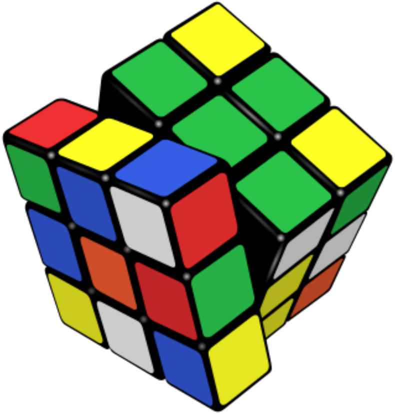


Eksempel på implicit graf: Rubik's Kube

Rubics Cube.

- $n + m = 43.252.003.274.489.856.000 \simeq 43 \cdot 10^1 8$

Hvad er det færreste antal træk til den "ordnede" kube, uanset hvor rodet, kuben starter som at være?



Eksempel på implicit graf: Rubik's Kube

År	nedre grænse	øvre grænse
1981	18	52
1990	18	42
1992	18	39
1992	18	37
1995	18	29
1995	20	29
2005	20	28
2006	20	27
2007	20	26
2008	20	25
2008	20	23
2008	20	22
2010	20	20

Rettede grafer

- Rettede grafer
- Repræsentation
- Dybdeførstsøgning / breddeførstsøgning
- Topologisk sortering
- Stærkt sammenhængende komponenter
- Implicitte grafer