Orienterede grafer

- Introduktion
- Repræsentation
- Søgning
- Topologisk sortering og DAGs
- Stærke sammenhængskomponenter
- Implicitte grafer

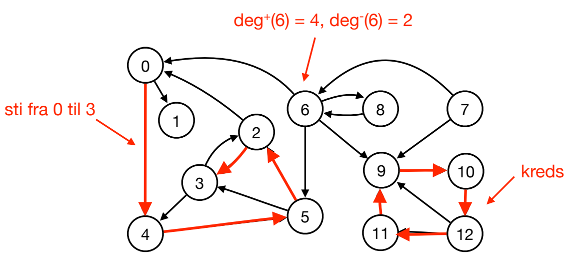
Philip Bille

Orienterede grafer

- Introduktion
- Repræsentation
- Søgning
- Topologisk sortering og DAGs
- Stærke sammenhængskomponenter
- Implicitte grafer

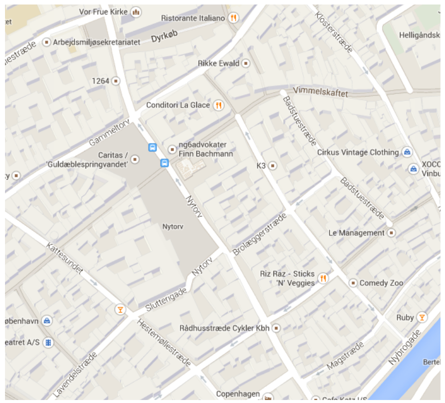
Orienterede grafer

- Orienteret graf *(directed graph)*. Mængde af knuder forbundet parvis med orienterede kanter.



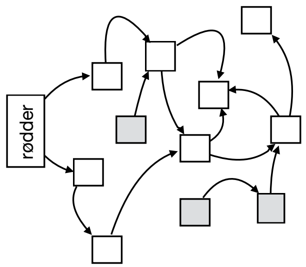
Vejnetværk

- Knude = vejkryds, kant = ensrettet vej.



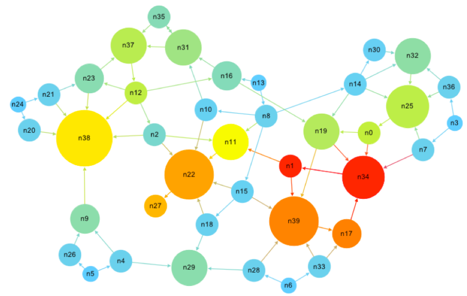
Spildopsamling (*garbage collection*)

- Knude = objekt, kant = peger/reference.
- Hvilke objekter kan vi nå fra en rod?



WWW

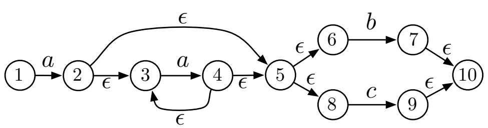
- Knude = hjemmeside, kant = hyperlink.
- Webcrawling
- PageRank



http://computationalculture.net/article/what_is_in_pagerank

Automater og regulære udtryk

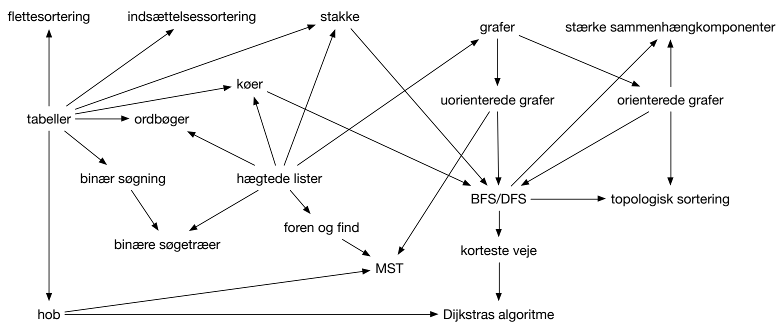
- Knude = tilstand, kant = tilstandsovergang.
- Accepterer automaten “aab” = findes sti fra 1 til 10 der matcher “aab”?
- Regulære udtryk kan repræsenteres som automat.



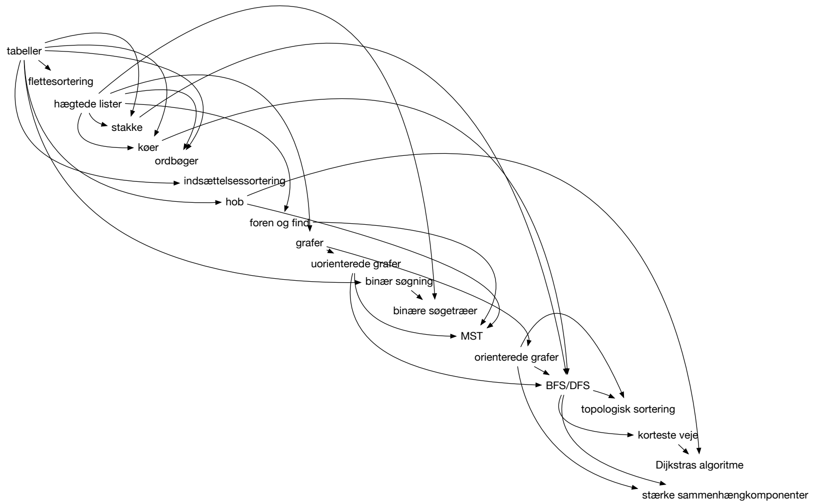
R = a·(a*)·(b|c)

Afhængigheder

- Knude = emner, kant = afhængighed.
- Er der nogle cykliske afhængigheder? Kan vi finde en rækkefølge af emnerne så vi undgår cyckliske afhængigheder?



Afhængigheder

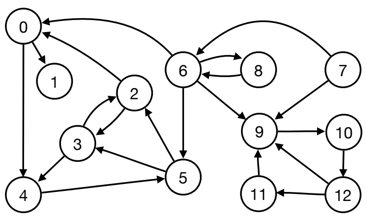


Grafanvendelser

graf	knuder	kanter
internet	hjemmeside	hyperlink
transport	vejkryds	ensrettet vej
skedulering	job	precedens relation
infektionssygdom	person	infektion
citation	artikel	citation
objekt graf	objekter	pegere/referencer
objekt hieraki	klasse	nedarver fra
kontrol-flow	kode	hop

Orienterede grafer

- Lemma. $\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = m$.
- Bevis. Hver kant har netop en startknude og slutknude.



Algoritmiske problemer på orienterede grafer

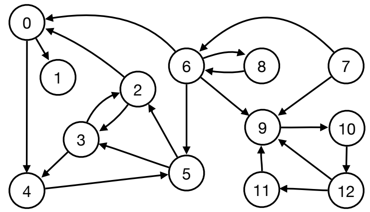
- Sti. Er der en sti fra s til t?
- Korteste sti. Hvad er den korteste sti fra s til t.
- Orienteret acyklisk graf. Findes der kredse i grafen?
- Topologisk sortering. Kan vi ordne knuder i en sekvens så alle kanter peger samme vej?
- Stærk sammenhængskomponent. Er der en vej mellem alle par af knuder?
- Transitiv afslutning. For hvilke knuder v og w er der en sti fra v til w?

Orienterede grafer

- Introduktion
- **Repræsentation**
- Søgning
- Topologisk sortering og DAGs
- Stærke sammenhængskomponenter
- Implicitte grafer

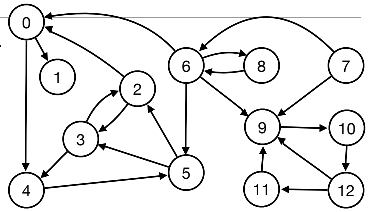
Repræsentation

- G orienteret graf med n knuder og m kanter.
- Repræsentation. Vi skal bruge følgende operationer på uorienterede grafer.
  - PointsTo(v, u): afgør om knude v peger på knude u.
  - Neighbors(v): returner alle knuder som v peger på.
  - Insert(v, u): tilføj kant (v, u) til G (medmindre den allerede findes).



Incidensmatrix

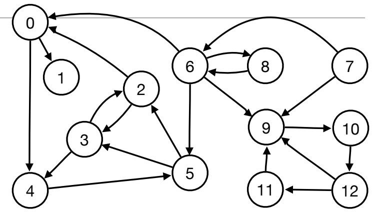
- Orienteret graf G med n knuder og m kanter.
- Incidensmatrix.
  - 2D n × n tabel A.
  - A[i,j] = 1 hvis i peger på j og 0 ellers.
- Plads. $O(n^2)$
- Tid.
  - PointsTo i O(1) tid
  - Neighbors(v) i O(n) tid.
  - Insert(v, u) i O(1) tid.

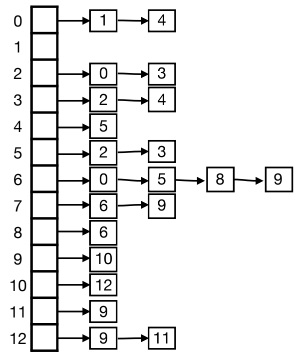


	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	0	0	0	0	0	0	0
5	0	0	1	1	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	1	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	0	0	1	0	0	0
8	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0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

- Graf G med n knuder og m kanter.
- Incidensliste.
  - Tabel A[0..n-1].
  - A[i] indeholder liste af knuder som i peger på.
- Plads. $O(n + \sum_{v \in V} \deg^+(v)) = O(n + m)$
- Tid.
  - PointsTo, Neighbors og Insert i O(deg(v)) tid.





Repræsentation

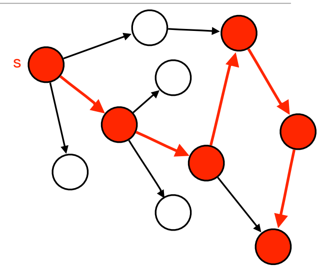
Datastruktur	PointsTo	Neighbors	Insert	Plads
incidensmatrix	O(1)	O(n)	O(1)	$O(n^2)$
incidensliste	$O(\deg^+(v))$	$O(\deg^+(v))$	$O(\deg^+(v))$	O(n+m)

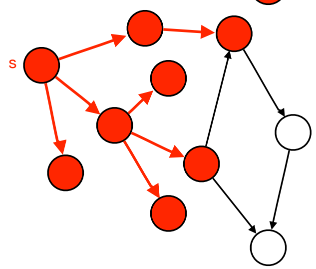
Orienterede grafer

- Introduktion
- Repræsentation
- **Søgning**
- Topologisk sortering og DAGs
- Stærke sammenhængskomponenter
- Implicitte grafer

Søgning

- Dybdeførst søgning.
  - Lad alle knuder være umarkede og besøg knude s.
  - Når vi besøger knude v:
    - Marker v.
    - Besøg alle umarkede knuder som v peger på rekursivt.
- Breddeførst søgning.
  - Lad alle knuder være umarkede.
  - Marker s og tilføj s til kø K.
  - Sålænge K ikke er tom:
    - Udtag knude v fra K.
    - For alle umarkede knuder u som v peger på
      - Marker u.
      - Tilføj u til K.
- Tid. O(n + m)



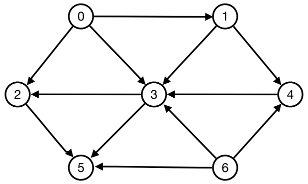


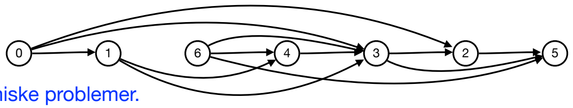
Orienterede grafer

- Introduktion
- Repræsentation
- Søgning
- **Topologisk sortering og DAGs**
- Stærke sammenhængskomponenter
- Implicitte grafer

Topologisk sortering og DAGs

- DAG. Orienteret acyklisk graf (*directed acyclic graph*). Graf uden kredse.
- Topologisk sortering (*topological sorting*). Ordning af knuder $v_0, v_1, \ldots, v_{n-1}$ fra venstre til højre således at alle kanter peger mod højre.

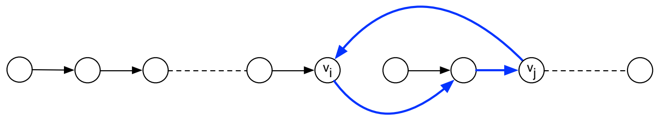




- Algoritmiske problemer.
  - Afgør om G er en DAG.
  - Beregn en topologisk sortering (hvis den findes).
- Mål. Vise G er en DAG $\Leftrightarrow$ G har en topologisk sortering + find algoritme til at løse begge problemer.

DAGs og topologisk sortering

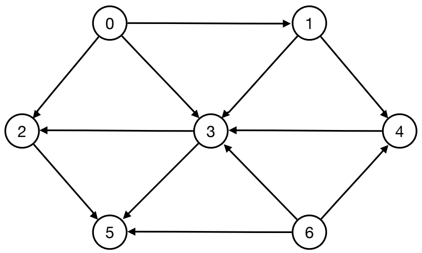
- Lemma. G har en topologisk sortering $\Rightarrow$ G er en DAG.
- Bevis. Modbevis.
  - Antag G har topologisk sortering $v_0, v_1, \ldots, v_{n-1}$ og ikke er en DAG.
  - $\Rightarrow$ G indeholder kreds K.



  - Kig på knude v_i i K som er længst til venstre i sortering og knude v_j lige før
  - (v_j, v_i) er kant i K og dermed i G.
  - Men v_i er før v_j i rækkefølge $\Rightarrow$ $v_0, v_1, \ldots, v_{n-1}$ er ikke en topologisk sortering.

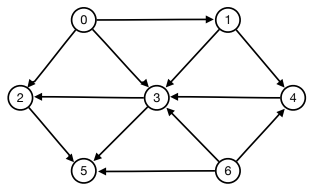
DAGs og topologisk sortering

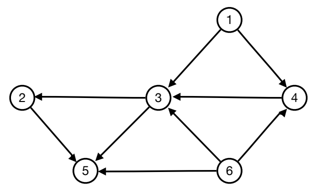
- Opgave. Givet en DAG, find på en strategi til at beregne en topologisk sortering.



DAGs og topologisk sortering

- Lemma. G er en DAG $\Rightarrow$ G har en knude med indgrad 0.

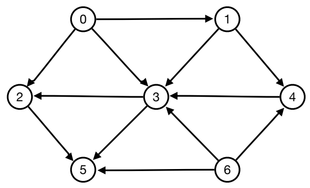


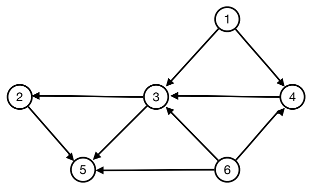


- Bevis. Modbevis.
  - Antag G er en DAG og alle knuder har indgrad ≥ 1.
  - Vælg knude v og følg kanter baglæns fra v.
  - Efter n+1 skridt må der være mindst 1 knude u vi har besøgt 2 gange.
  - u + de knuder mellem første og anden gang vi besøger u udgør kreds.

DAGs og topologisk sortering

- Lemma. G er en DAG $\Longrightarrow$ G har topologisk sortering.





- Bevis. Induktion over n (antallet af knuder).
  - Basis n = 1.
  - Induktionsskridt n > 1.
    - Find knude v med indgrad 0.
    - G - {v} er en DAG $\Longrightarrow$ G - {v} har en topologisk sortering.
    - Placer v længst til venstre efterfulgt af den topologiske sortering af G - {v}. Giver topologisk sortering af G da v ikke har nogen indgående kanter.

DAGs og topologisk sortering

- Algoritme. Induktionsbevis som algoritme.

```
TopSort(G)
   if G = ({v},∅) print v
   else
      find v så deg^-(v) = 0
      print v
      TopSort(G - {v})
```

- Korrekthed. Følger af induktionsbevis.

Implementation

- Mål. Implementer algoritmen effektivt på incidenslisterepræsentation

```
TopSort(G)
   if G = ({v},∅) print v
   else
      find v så deg^-(v) = 0
      print v
      TopSort(G - {v})
```

Implementation

- Løsning 1. Konstruer omvendt graf G^R. Linæer søgning i incidensliste for G^R efter knude med indgrad 0 i hvert skridt.

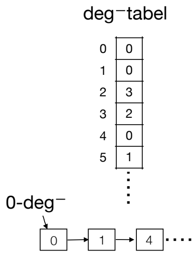
```
TopSort(G)
   if G = ({v},∅) print v
   else
      find v så deg^-(v) = 0
      print v
      TopSort(G - {v})
```

- Tid per knude.
  - Find knude v med indgrad 0: O(n).
  - Fjern kanter ud af v: $O(\deg^+(v))$
- Samlet tid for n knuder. $O(n^2 + \sum_{v\in V} \deg^+(v)) = O(n^2 + m) = O(n^2)$.

Implementation

- Løsning 2. Vedligehold indgrad af hver knude i incidensliste + hægtet liste af alle knuder med indgrad 0.

```
TopSort(G)
   if G = ({v},∅) print v
   else
      find v så deg⁻(v) = 0
      print v
      TopSort(G - {v})
```



- Initialisering. O(n+m)
- Tid per knude.
  - Find knude v med indgrad 0: O(1).
  - Fjern kanter ud af v: $O(\deg^+(v))$
- Samlet tid for n knuder. $O(n + \sum_{v \in V} \deg^+(v)) = O(n + m)$.

Topologisk sortering og DAGs

- Lemma. G er en DAG ⟺ G har en topologisk sortering.
- Theorem. Vi kan afgøre om G er DAG og hvis den er så finde en topologisk sortering i O(n + m) tid.

Topologisk sortering med DFS

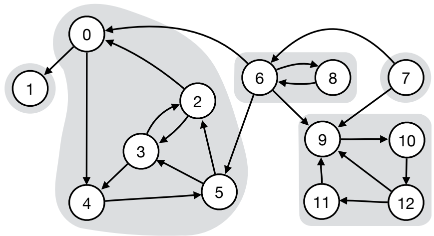
- Topologisk sortering med DFS.
- Ide.
  - Kør DFS på G.
  - Ved slutning af rekursivt kald til knude v læg v på stak.
  - Udskriv stak.
- Tid. O(n + m)
- Intuition. Finder rekursivt knuder med *udgrad* 0.

Orienterede grafer

- Introduktion
- Repræsentation
- Søgning
- Topologisk sortering og DAGs
- **Stærke sammenhængskomponenter**
- Implicitte grafer

Stærke sammenhængskomponenter

- Def. v og u er stærk sammenhængende (*strongly connected*) hvis der er sti fra v til u og fra u til v.
- Def. Et stærk sammenhængskomponent (*strongly connected component*) er en maksimal delmængde af stærk sammenhængende knuder.



Stærke sammenhængskomponenter med DFS

- Stærke sammenhængskomponenter med DFS.
- Ide.
  - Kør DFS på omvendt graf G^R.
  - Kør DFS på G ifht. rækkefølge fra første DFS. Hver rekursiv søgning finder et stærkt sammenhængskomponent.
- Intuition?
- Korrekthed. Se kap 22.5 i CLRS.
- Tid. O(n + m)

Stærke sammenhængskomponenter med DFS

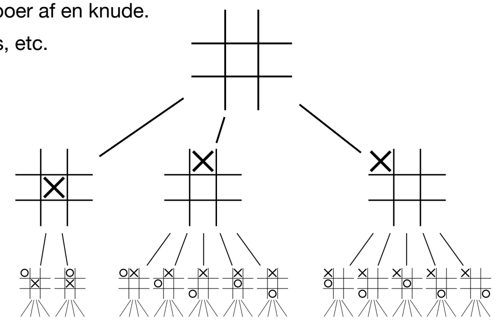
- Theorem. Vi kan finde alle stærke sammenhængskomponenter i O(n + m) tid.

Orienterede grafer

- Introduktion
- Repræsentation
- Søgning
- Topologisk sortering og DAGs
- Stærke sammenhængskomponenter
- Implicitte grafer

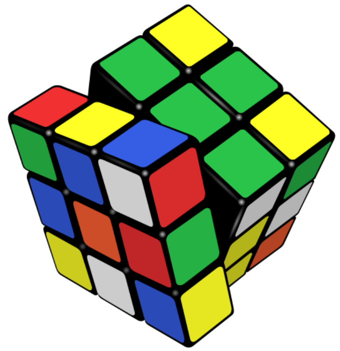
Implicit graf

- Implicit graf (*implicit graph*). Uorienteret/orienteret graf G repræsenteret implicit.
- Implicit repræsentation.
  - Startknude s.
  - Algoritme til at generere naboer af en knude.
- Bruges i spil, kunstig intelligens, etc.



Rubiks terning

- Rubiks terning
  - n+m = 43.252.003.274.489.856.000 ~ 43 trillioner.
- Hvad er det optimale antal træk man skal bruge på at "løse" en terning ligegyldigt hvor man starter?



Rubiks terning

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

www.cube20.org

Orienterede grafer

- Introduktion
- Repræsentation
- Søgning
- Topologisk sortering og DAGs
- Stærke sammenhængskomponenter
- Implicitte grafer