

GRAFEN

DATASTRUCTUREN

Dr. D.P. Huijsmans

2 november 2011

Universiteit Leiden LIACS

ONDERWERPEN

- Meest algemene datastructuur om onderlinge relaties van n elementen aan te geven, inclusief relaties tot zichzelf (attributen)
- Gerichte en ongerichte verbindingen
- Datastructuren voor grafen
- Paden, kortste pad
- Cycles
- Spanning tree
- Topologische sortering



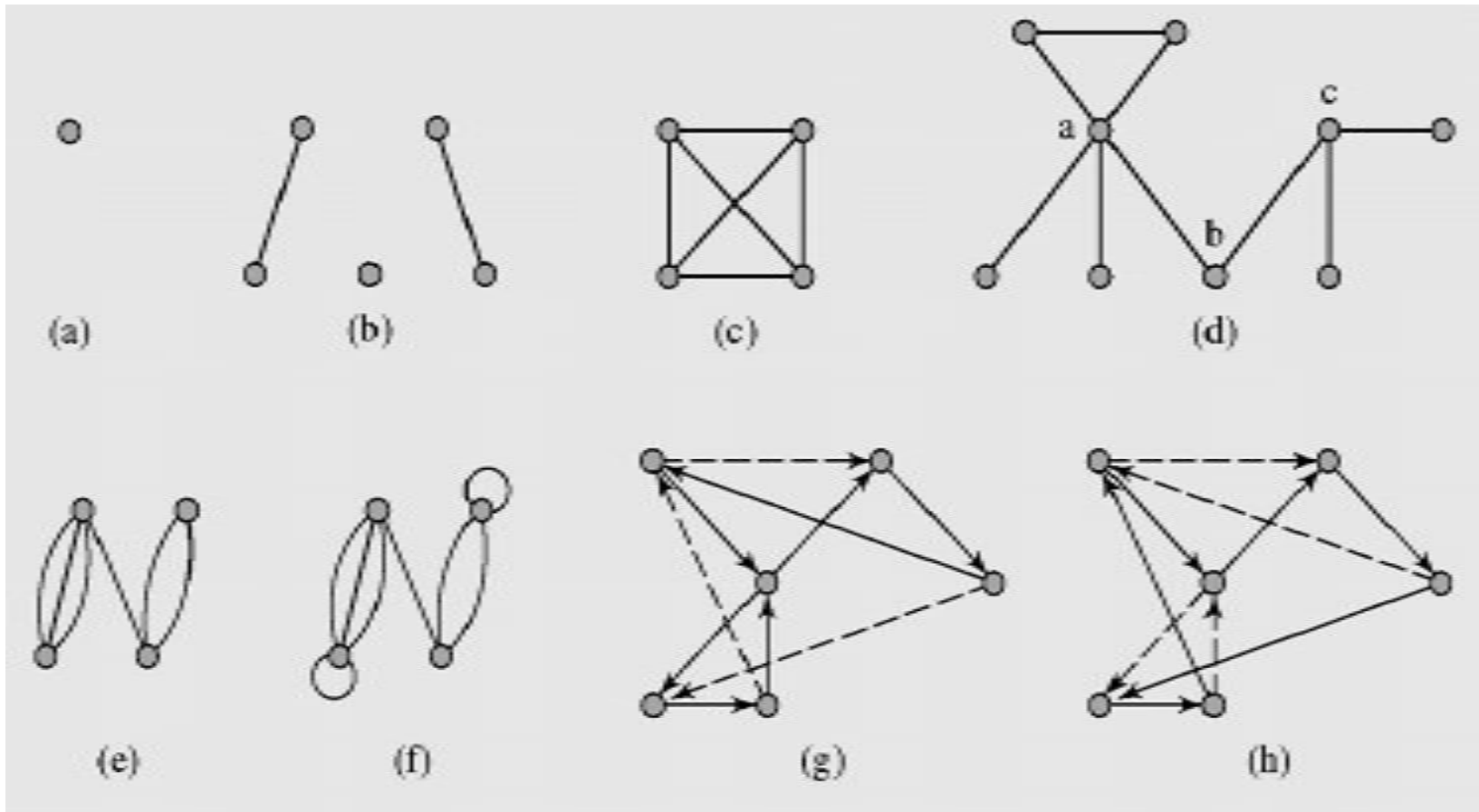
GRAFEN

EEN MEEST ALGEMENE DATASTRUCTUUR

- Om relaties tussen N objecten in zijn meest algemene vorm te representeren is een graaf de aangewezen datastructuur
- Een graaf bestaat uit N knopen (waarin waarden of records met informatie is opgeborgen) en plaats voor alle mogelijke relaties tussen die knopen
- Relaties heten ook/kunnen staan voor:
 - verbindingen, edges, overgangen, zijdes, takken, frequenties, gewichten, attributen etc.
- Verbindingen tussen knopen kunnen ongericht of gericht zijn:
 - digraaf in het geval van directed graph



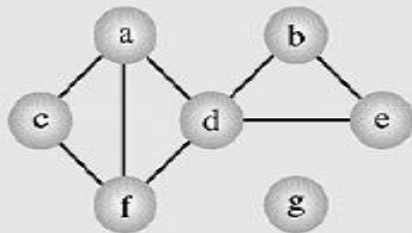
VOORBEELDEN VAN GRAFEN



Examples of graphs: (a–d) simple graphs; (c) a complete graph K_4 ; (e) a multigraph; (f) a pseudograph; (g) a circuit in a digraph; (h) a cycle in the digraph



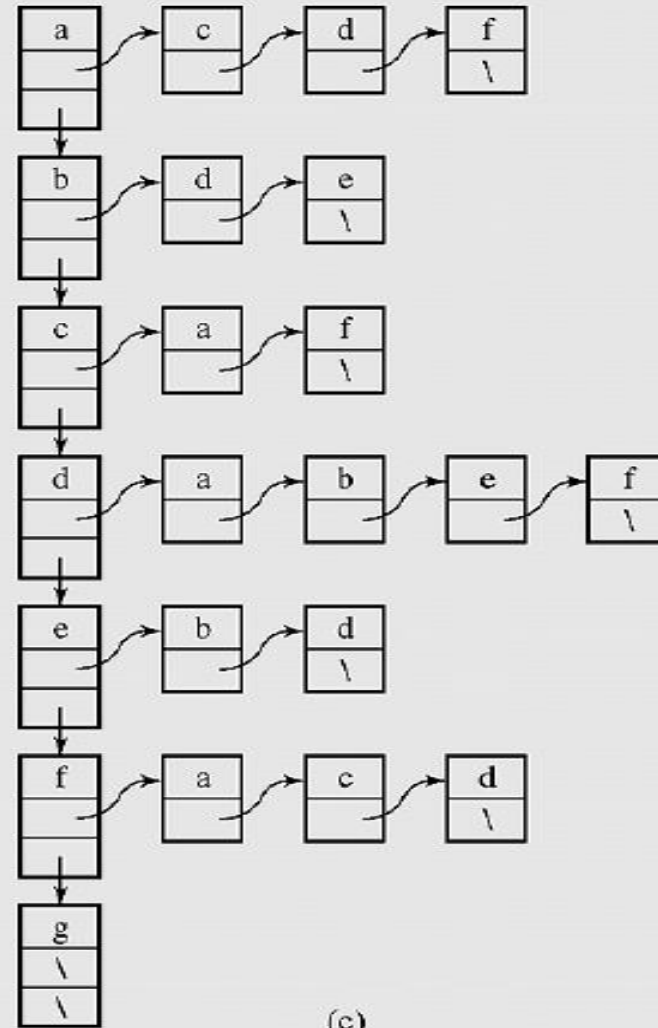
REPRESENTATIES VAN GRAFEN



(a)

a	c	d	f		
b	d	e			
c	a	f			
d	a	b	e	f	
e	b	d			
f	a	c	d		
g					

(b)



(c)

a) Visueel b) adjacency matrix c) linked lists representation



REPRESENTATIES VAN GRAFEN-2

	a	b	c	d	e	f	g
a	0	0	1	1	0	1	0
b	0	0	0	1	1	0	0
c	1	0	0	0	0	1	0
d	1	1	0	0	1	1	0
e	0	1	0	1	0	0	0
f	1	0	1	1	0	0	0
g	0	0	0	0	0	0	0

(d)

d) adjacency/connectivity matrix

	ac	ad	af	bd	be	cf	de	df
a	1	1	1	0	0	0	0	0
b	0	0	0	1	1	0	0	0
c	1	0	0	0	0	1	0	0
d	0	1	0	1	0	0	1	1
e	0	0	0	0	1	0	1	0
f	0	0	1	0	0	1	0	1
g	0	0	0	0	0	0	0	0

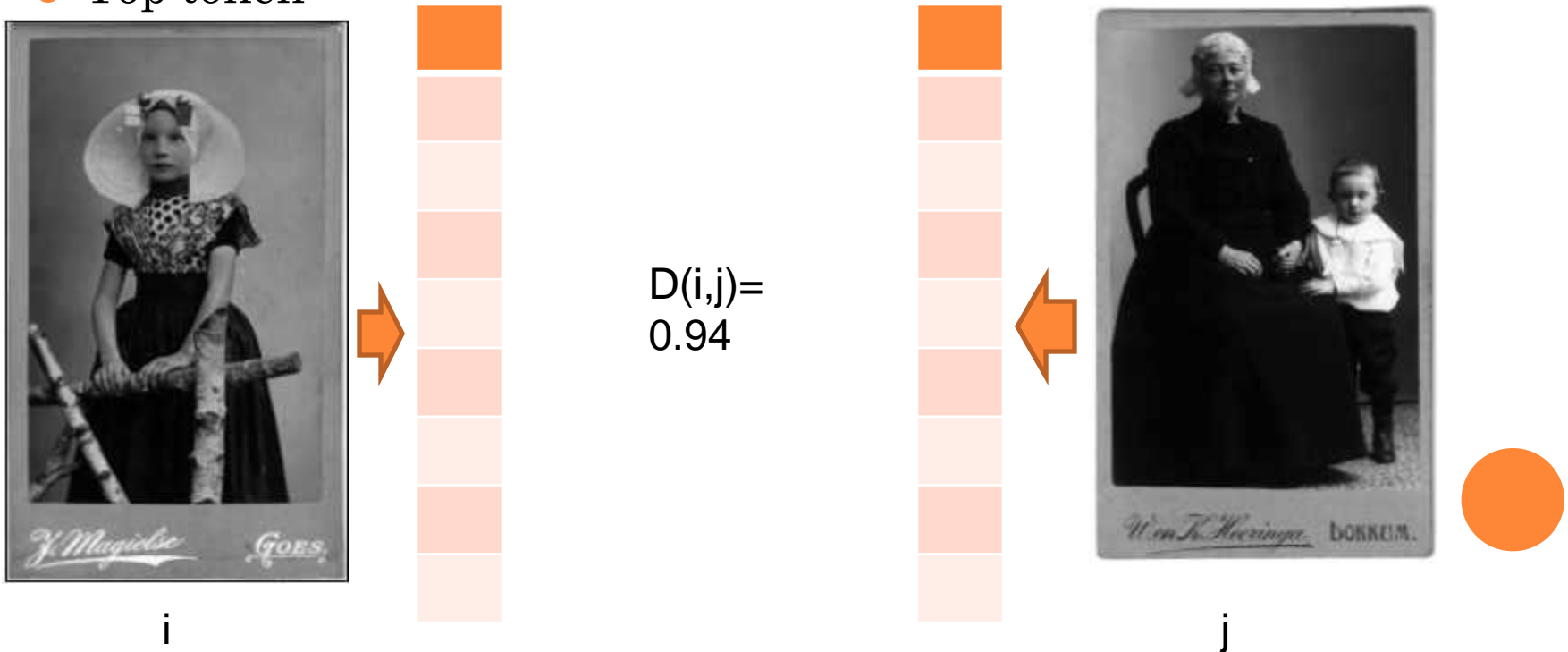
(e)

e) incidence matrix



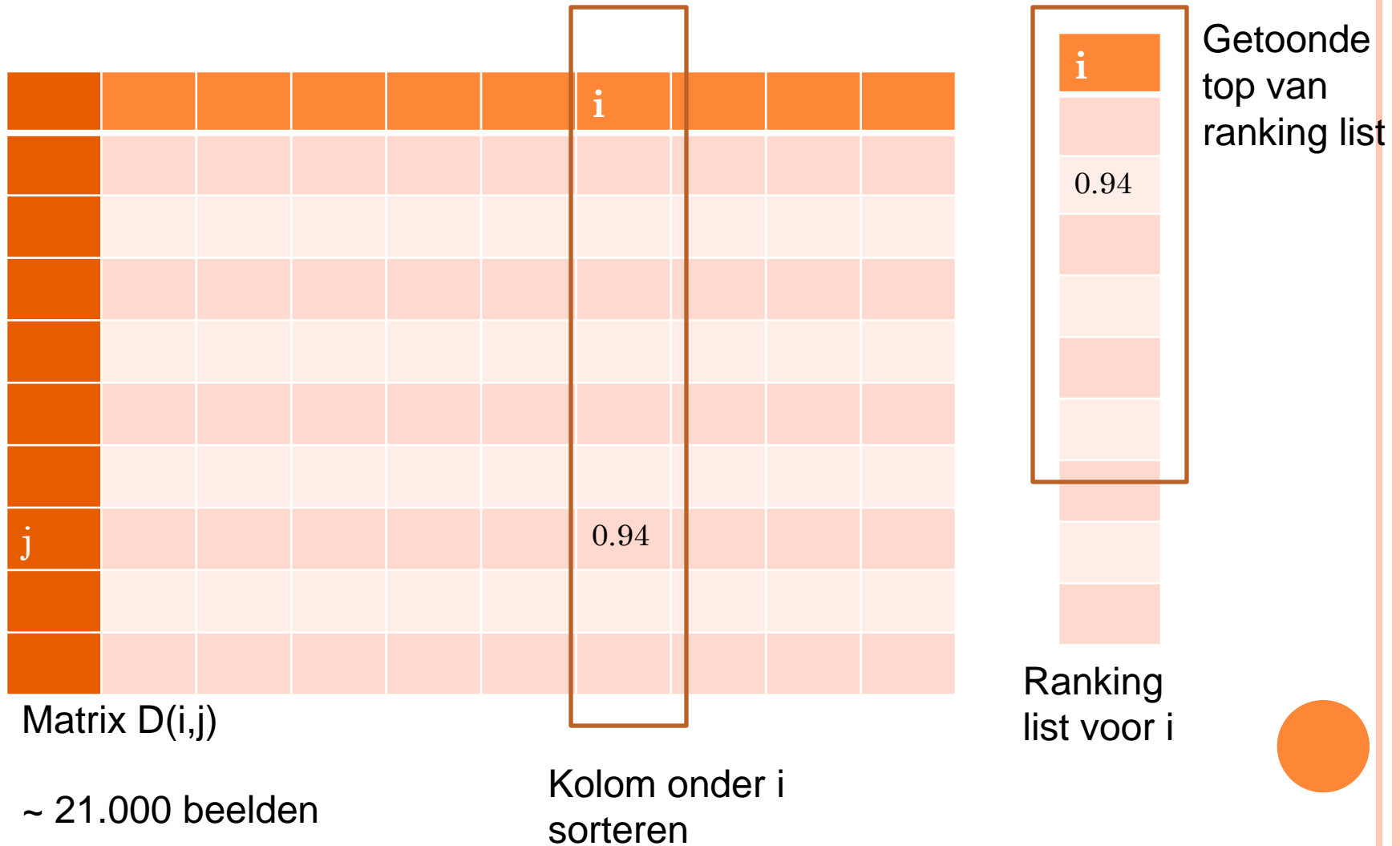
VOORBEELD GRAAF GEBRUIK BIJ ZOEKEN NAAR SOORTGELIJKE BEELDEN

- Opzet LCPD (Leiden 19th-Century Portrait Database)
- Op grond van kenmerkvector elk beeld in de database met elk ander beeld vergelijken -> afstandsmaat $D(i,j)$
- Resultaten per zoekbeeld i sorteren -> rank list $R(i,j)$
- Top tonen



VERVOLG VOORBEELD LCPD

VISUELE GELIJKHEID



DEMO LCPD



Content-based Image Retrieval in LCPD: the Leiden 19th-Century Portrait Database

[Leiden Imaging and Multimedia group](#), [Computer Science department](#) Leiden University
[Study and Documentation Centre for Photography \(Printroom\) of Leiden University](#),
Photohistorian Steven Wachlin (Biographical Information Photographers)
The Netherlands



Image search tools:

- **Studio:**

[Carte de Visite portraits made by the same photographic studio](#)

- **Album:**

[original family album context is virtually restored](#)

- **Browsing:**

[when the other entries fail](#)

- **Portrait Query:**

[starting at random using relevance feedback](#)

- **Logo Query:**



ADT_GRAAF

- Omdat de voorstellingen van grafen zo verschillen is het moeilijk de specificatie en functionaliteit van een Abstract Data Type voor een Graaf los te zien van de uiteindelijke implementatie
- Bekijk (op losse bladen):
- ADT_AdjGraaf specificatie
- ADT_LLGraaf specificatie



BEPERKINGEN OP GRAFEN VOOR MATRIXVOORSTELLING

- In het algemeen kunnen grafen:
 - Meerdere verbindingen hebben tussen 2 knopen (multigraaf)
 - Lege verbindingen hebben
 - Verbindingen naar zichzelf hebben (lussen, pseudograaf)
- Binnen datastructuren zullen we ons beperken tot grafen die:
 - Maximaal 1 gerichte verbinding hebben tussen 2 knopen (geen parallelle gerichte verbindingen)
 - Geen lege verbindingen hebben (alleen verbindingen tussen knopen)
 - Wel/niet lussen hebben (verbinding met zichzelf): met lussen noemen we het een pseudograaf



MATRIX REPRESENTATIE VAN EEN GRAAF

- De $N \times N$ connectivity matrix waarin voor elke combinatie van N objecten met elkaar zowel de
- $N_i \rightarrow N_j$ en de $N_j \rightarrow N_i$ met $i, j \in [1..N]$ verbinding gerepresenteerd kan worden geeft het beste totaal overzicht van de onderlinge relaties

$N_i \setminus N_j$	1	2	3	4
1	1	1	1	0
2	0	0	0	1
3	0	0	0	0
4	0	1	0	0

1= verbinding $N_i \rightarrow N_j$
0= geen verbinding

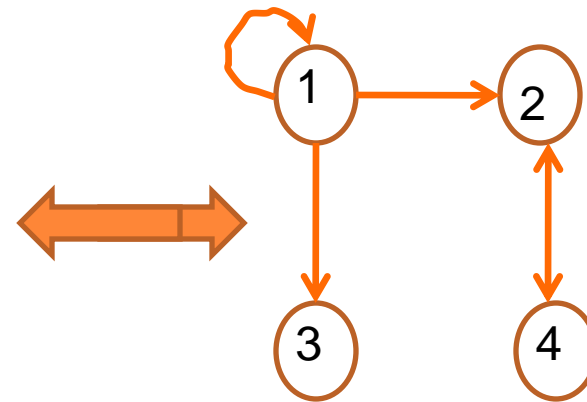
In principe is een bit per mogelijke verbinding voldoende



DIGRAAF REPRESENTATIE VAN EEN MATRIX

- Inzichtelijker voor een bepaalde configuratie is vaak de representatie in de vorm van knopen en bestaande gerichte verbindingen
- Maar de matrix geeft je een beter totaal overzicht

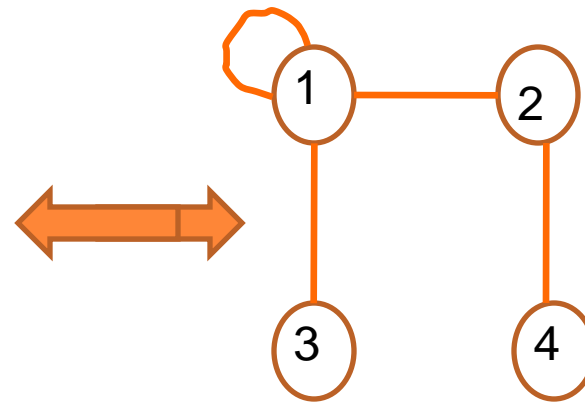
$N_i \setminus N_j$	1	2	3	4
1	1	1	1	0
2	0	0	0	1
3	0	0	0	0
4	0	1	0	0



(ONGERICHTE) GRAAF REPRESENTATIE VAN EEN MATRIX

- Bij ongerichte verbindingen geldt de bestaande relatie wederzijds $N_i \leftrightarrow N_j$
- Er kan volstaan worden met een
 - boven of onder driehoeksmatrix vorm inclusief diagonaal voor lussen

$N_i \setminus N_j$	1	2	3	4
1	1	1	1	0
2		0	0	1
3			0	0
4				0

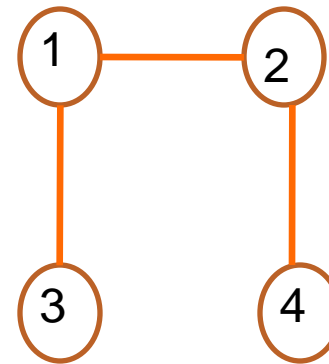


(ONGERICHTE) GRAAF

REPRESENTATIE VAN EEN MATRIX ZONDER LUSSEN

- Bij ongerichte verbindingen geldt de bestaande relatie wederzijds $N_i \leftrightarrow N_j$
- En kan volstaan worden met een
 - boven of onder driehoeksmatrix vorm zonder diagonaal

$N_i \setminus N_j$	1	2	3	4
1		1	1	0
2			0	1
3				0
4				

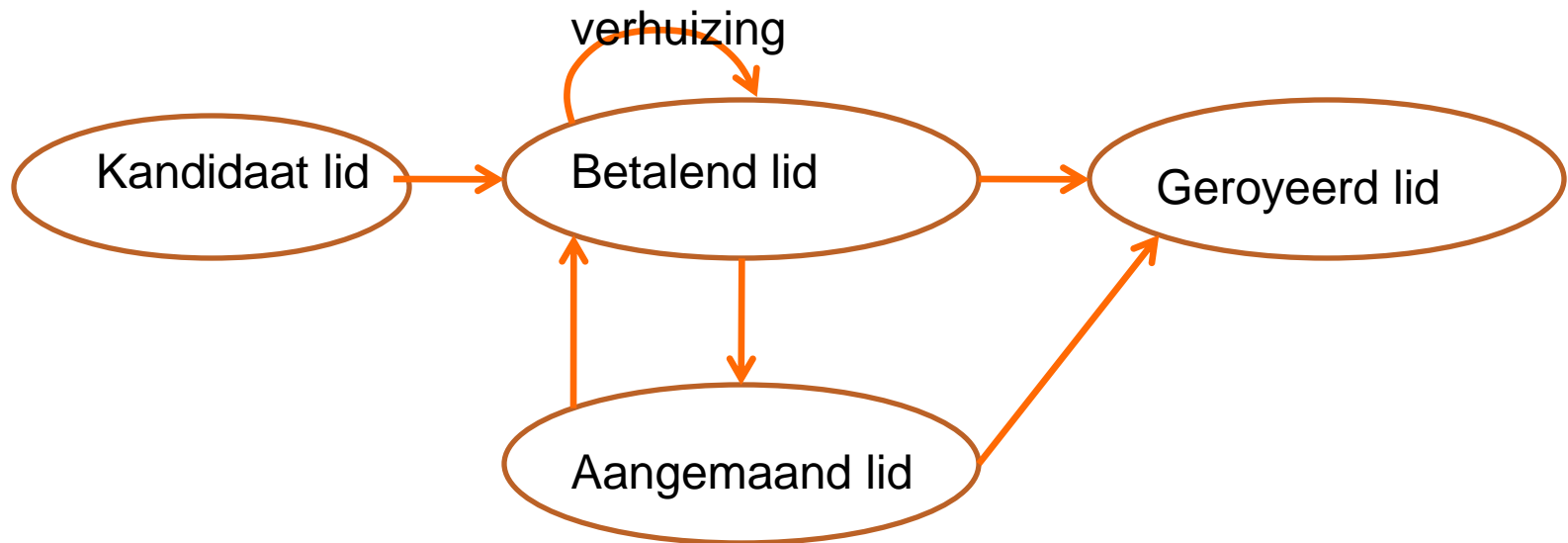


WEL/NIET LUSSEN WAAR NUTTIG VOOR?

- Lussen staan voor relaties die een knoop met zichzelf kan hebben:
- - attribuut (b.v. edge met leeftijd informatie)
- - onveranderde toestand na bewerking:
 - Grafen worden veel gebruikt om mogelijke toestanden van een systeem te beschrijven inclusief alle mogelijke toestandsovergangen
 - B.v. contributietoestand van persoon in ledenadministratie;
 - Blijft betalend lid na verhuizing

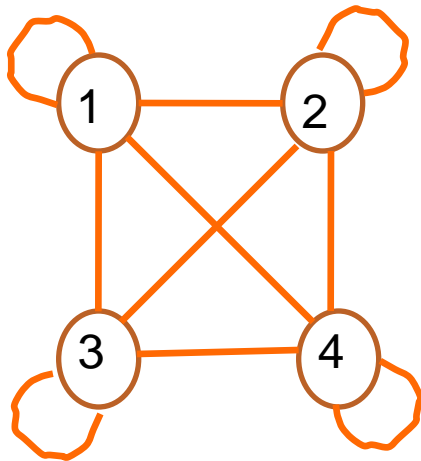
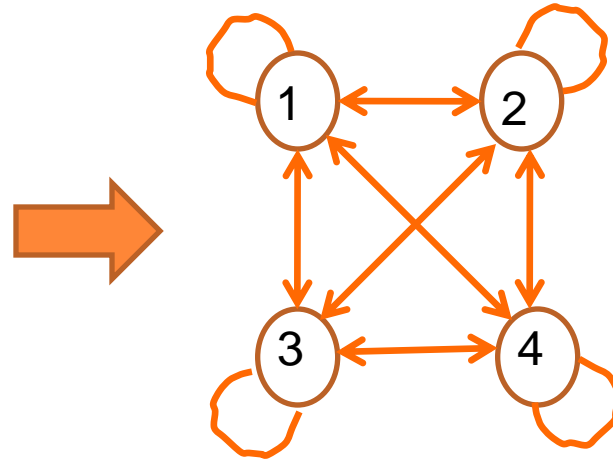


LIDMAATSCHAPSGRAAF



VOLLEDIGE GRAAF

$N_i \setminus N_j$	1	2	3	4
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1



$N_i \setminus N_j$	1	2	3	4
1	1	1	1	1
2		1	1	1
3			1	1
4				1



LINKED-LIST

SPECIALE VERSIE VAN EEN DIGRAAF

$N_i \backslash N_j$	1	2	3	4	5	6	7	8	9
1		1							
2			1						
3				1					
4					1				
5						1			
6							1		
7								1	
8									1
9									

Lege entries zijn 0

Er is 1 gericht pad van 1->2->3->4->5->6->7->8->9



LINKED-LIST 2

SPECIALE VERSIE VAN EEN DIGRAAF

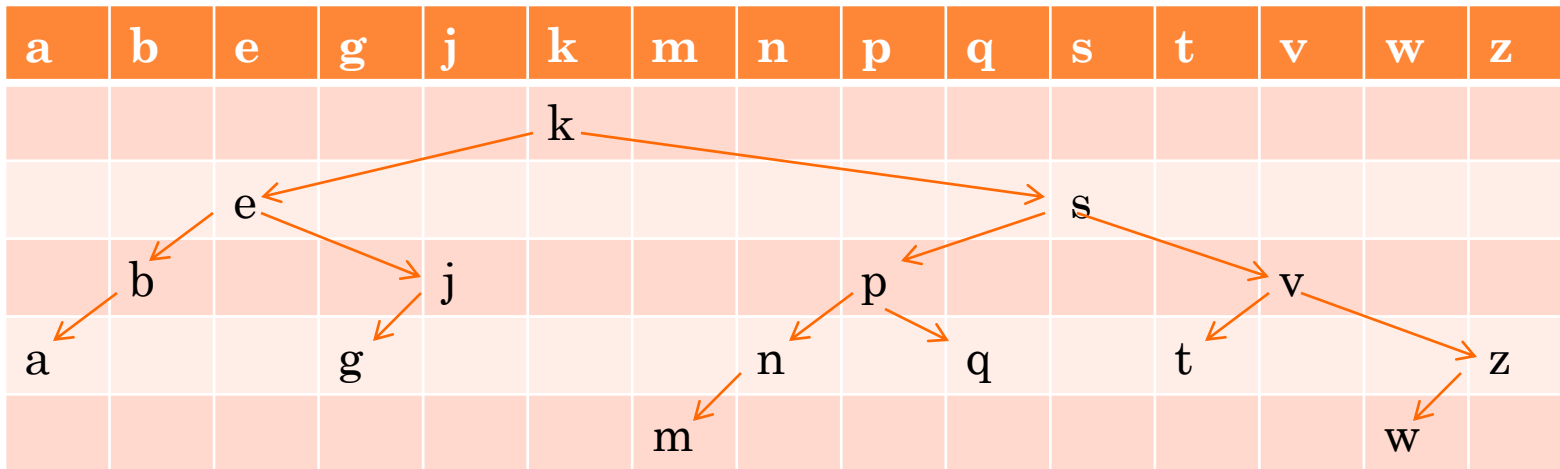
$N_i \backslash N_j$	1	2	3	4	5	6	7	8	9
1									1
2							1		
3									
4								1	
5		1							
6					1				
7				1					
8			1						
9						1			

Lege entries zijn 0

Er is 1 gericht pad van 1->9->6->5->2->7->4->8->3



VOORBEELD BINAIRE ZOEK BOOM



Van alle mogelijke kinderen (15x2) zijn er 14 niet nil
Efficientie opslag $14/30 \sim 47\%$



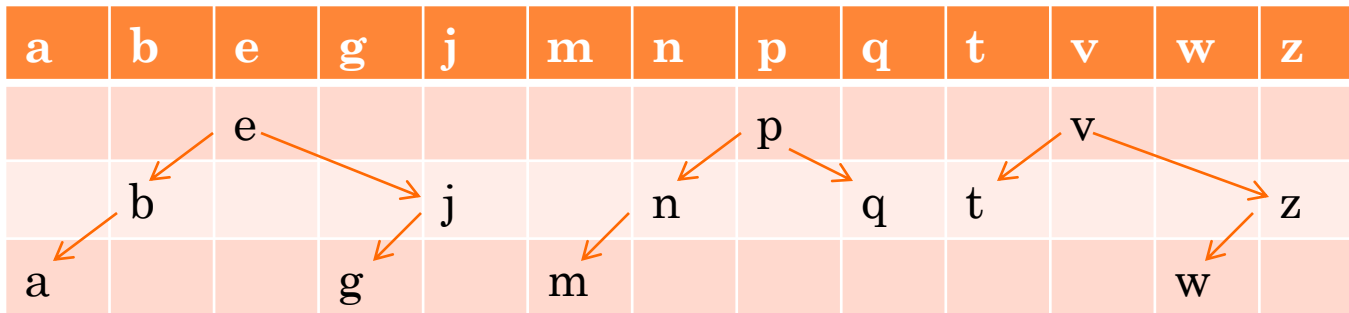
GRAAF VAN DEZE BINAIRE ZOEK BOOM

	a	b	e	g	j	k	m	n	p	q	s	t	v	w	z
a															
b	1														
e		1			1										
g															
j				1											
k			1								1				
m															
n							1								
p								1	1						
q															
s									1				1		
t															
v												1			1
w															
z														1	

Van de
15x15
(225)
verbinding
en zijn er
14
gebruikt
Efficiëntie
~ 6%
Wel maar
1 bit per
verbinding
nodig !!



VOORBEELD FOREST



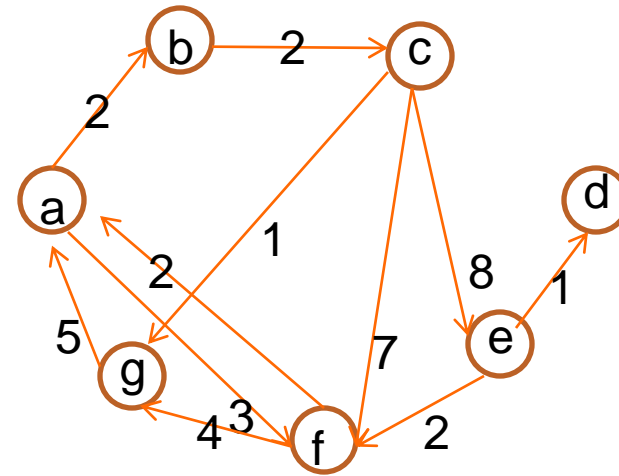
Als uit het voorbeeld van de binaire zoekboom k en s worden weggelaten blijven er 3 onderling niet verbonden graaf structuren over die ook m.b.v. een adjacency matrix Weergegeven kan worden

Elke graaf of stel grafen kan m.b.v. een adjacency matrix weergegeven worden



GRAAF MET GEWOGEN VERBINDINGEN

	a	b	c	d	e	f	g
a	∞	2	∞	∞	∞	3	∞
b	∞	∞	2	∞	∞	∞	∞
c	∞	∞	∞	∞	8	7	1
d	∞	∞	∞	∞	∞	∞	∞
e	∞	∞	∞	1	∞	2	∞
f	2	∞	∞	∞	∞	∞	4
g	5	∞	∞	∞	∞	∞	∞



i.p.v. ∞ kan maxint genomen worden

Verbonden graaf?

Boom?

Kortste pad a->d?

Langste pad a -> d?

Cycles?



GRAAF PROBLEMEN DIE BEKEKEN WORDEN

- Bekende graaf problemen zijn:
- 1) bepaal of twee knopen in de graaf verbonden zijn
- 2) als de verbindingen gewichten hebben, bepaal een kortste pad van een bepaalde knoop naar al de andere knopen
- 3) in een verbonden graaf bepaal een minimum verbonden subgraaf (spanning tree); als de graaf verbindingen gewichten hebben, zodanig dat de spanning tree minimaal is



GRAAF DEFINITIES

- Graaf (V,E) = knopen (Vertices), verbindingen tussen knopen (Edges)
- Subgraaf:= deelverzameling van (V,E)
- pad:= reeks verbonden knopen
- Simpel pad:= pad waarin een knoop maar 1 maal voorkomt
- Verbonden graaf:= er is een pad tussen elk paar knopen
- Buurknoop (adjacent):= 2 knopen die met elkaar verbonden zijn
- Buren van een knoop:= verzameling buurknopen van een knoop
- Boom:= verbonden graaf zonder cycli (verbonden graaf met n knopen en $n-1$ edges \rightarrow boom)
- Spanning Tree:= verbonden graaf min cycli
- Oriented Tree:= boom met een root knoop
- Digraaf:= directed graaf; elke verbinding (edge) heeft een richting (direction)
- Weighted graph:= verbindingen hebben een weegfactor



AD 1) VERBINDINGSSTRUCTUUR

- Er zijn meerdere manieren om een verbindingsstructuur te representeren
- Veelgebruikt zijn:
- Adjacency matrix
- Knoop en takken lijsten (node & edge lists)
- Vaak worden ook speciale datastructuren gebruikt b.v. in computer grafiek om reden van efficiency



3D OPPERVLAKTE MODEL

- Voor het weergeven van een ruimtelijk oppervlak maakt men vaak gebruik van een benadering met een driehoeksbelegging.
- Voordeel van een driehoeksbelegging is dat:
 - 3 punten een vlak definiëren
- Bij wijziging van gezichtspunt:
 - Alleen de 3 hoekpunten (vertices) getransformeerd hoeven te worden (vertex list)
 - De zijdes van een driehoek worden gevormd door verbindingen tussen 2 hoekpunten (edge list)
 - Een oppervlaktenormaal en een textuur gegeven kan zijn voor de positieve halfzijde
 - Hiervoor algoritmes bestaan die een lijnbelegging genereren met belichtingsverloop



VERTEX-, EDGE- EN TRIANGLE-LIST

Van de datastructuren hoeft alleen de vertex list getransformeerd te worden, de triangle en edge list waarvan de indexen verwijzen naar de indexen van de (getransformeerde) hoekpunten blijven gelijk

vertex	x	y	z
1	1.14	2.3	-16.8
2	...		
3			
4			
5			
6			
7			
8	...		
9	34.7	18.2	-19.8

edge	vbegin	vend
1	1	2
2	2	3
3	3	1
4	3	4
5	4	1
6	4	5
7	5	1
8	5	6
9	6	1

Δ	v1	v2	v3
1	1	2	3
2	1	3	4
3	1	4	5



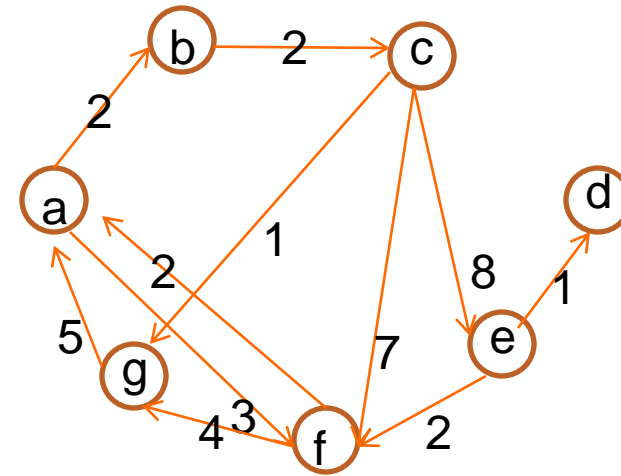
GRAAF WANDELINGEN (GRAPH TRAVERSAL)

- Om een wandeling af te werken hebben we per knoop een status veld nodig
- Status [waiting,ready,processed]
- Initieel status=waiting voor alle knopen
- Soorten doorloop:
 - Breadth first
 - Depth first
 - Priority first



BREADTH- & DEPTH-FIRST WANDELING GRAAF MET GEWOGEN VERBINDINGEN

	a	b	c	d	e	f	g
a	∞	2	∞	∞	∞	3	∞
b	∞	∞	2	∞	∞	∞	∞
c	∞	∞	∞	∞	8	7	1
d	∞	∞	∞	∞	∞	∞	∞
e	∞	∞	∞	1	∞	2	∞
f	2	∞	∞	∞	∞	∞	4
g	5	∞	∞	∞	∞	∞	∞



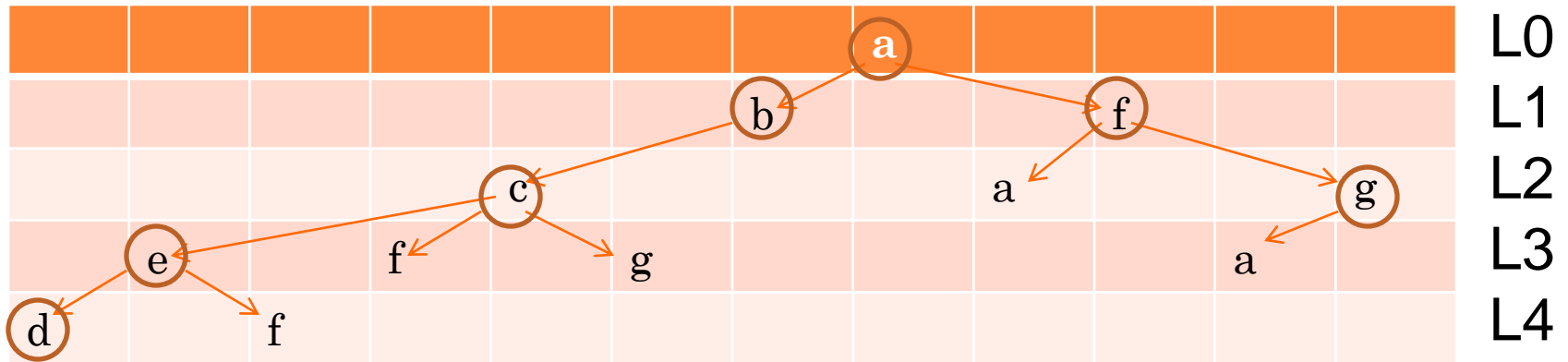
Niet bezocht=0;Bezocht ->1

Breadth-first: a,b,f dan c dan g dan e dan d:a,b,f,c,g,e,d (+priority queue)

Depth-first : a->b dan b->c dan c->e dan e->d dan c->f dan f->g:
a,b,c,e,d,f,g (+stack)



BREADTH-FIRST WANDELING IN GEWOGEN GRAAF

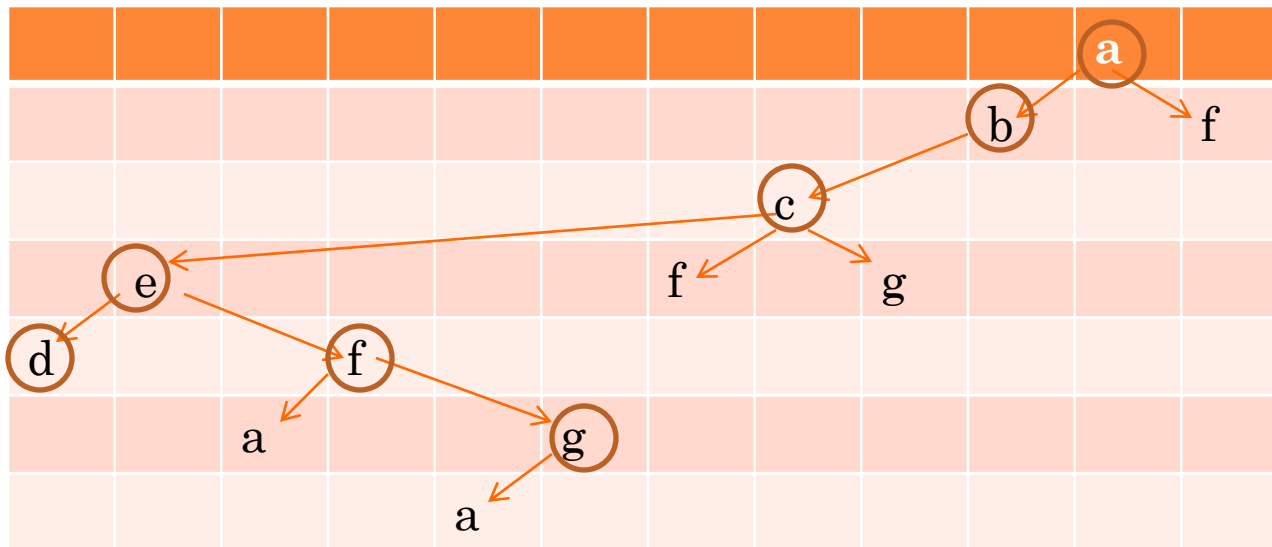


Li geeft buurafstand tot knoop a weer



DEPTH FIRST PREORDER WANDELING IN GEWOGEN GRAAF

- Volgorde knopen a t/m g

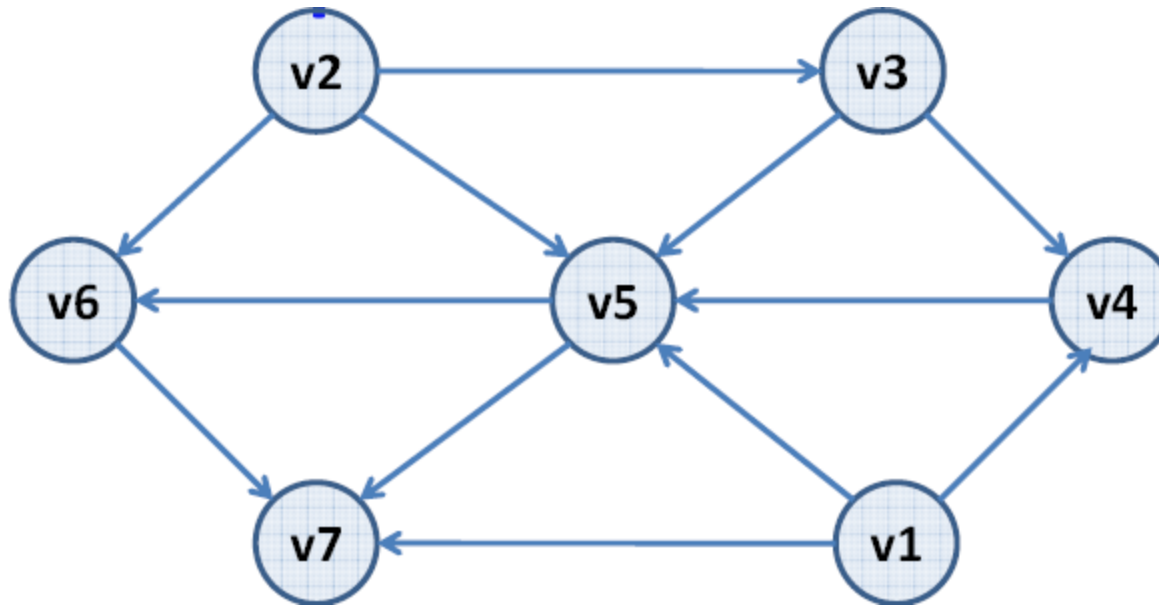


PreOrder wandeling door depthfirst boom omdat wandeling via L1 verbonden burens moet



TOPOLOGISCHE SORTERING VAN EEN DIGRAAF

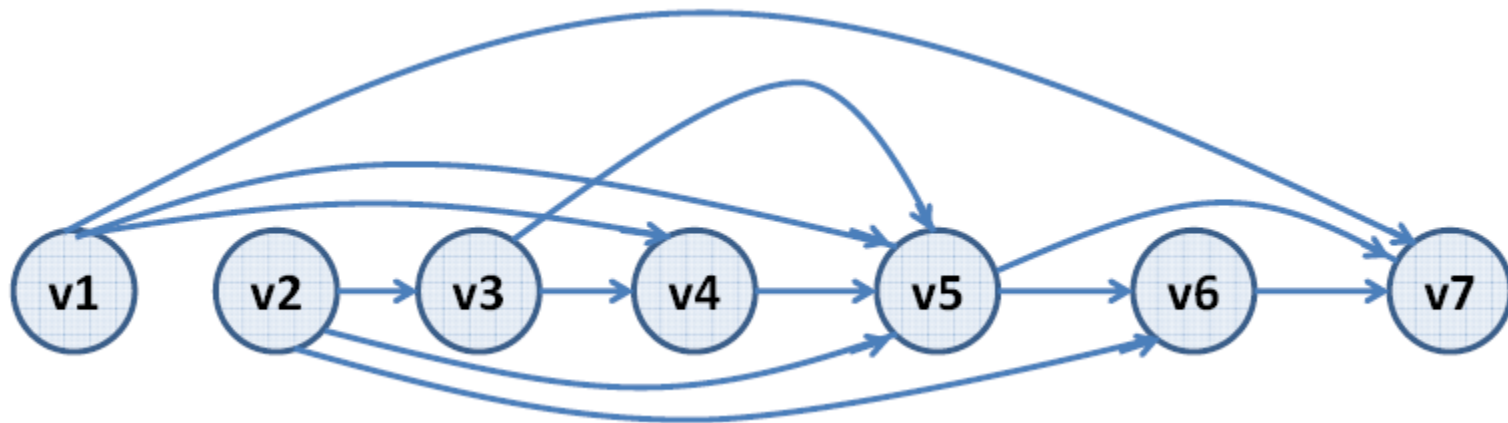
KRITIEKE PAD ALGORITME



Heel informeel Algoritme voor lineariseren: zolang er knopen zijn,
verwijder een knoop met alleen inkomende takken; verwijder z'n takken
en zet deze knoop op stack
Zet op eind rest knopen op stack en voer van stack gehaalde knopen uit



GELINEARISEERDE ACYCLISCHE DIGRAAF



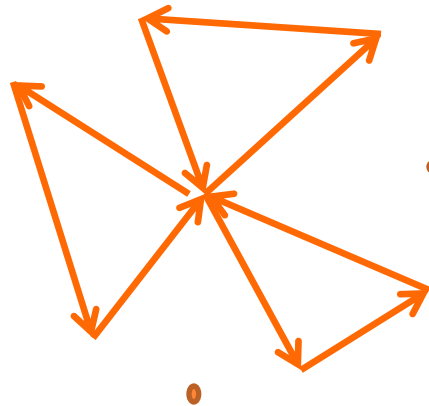
BIPARTITE GRAFEN

- Een speciaal geval van grafen wordt gevormd door (mogelijke) relaties tussen 2 verzamelingen van objecten N en M , waarvoor een $N \times M$ matrix gebruikt kan worden
- Denk aan:
- Matchen van (hetero)partners
- Toewijzen werk aan werknemers gegeven vaardigheden
- Plannen busroutes, bussen en chauffeurs



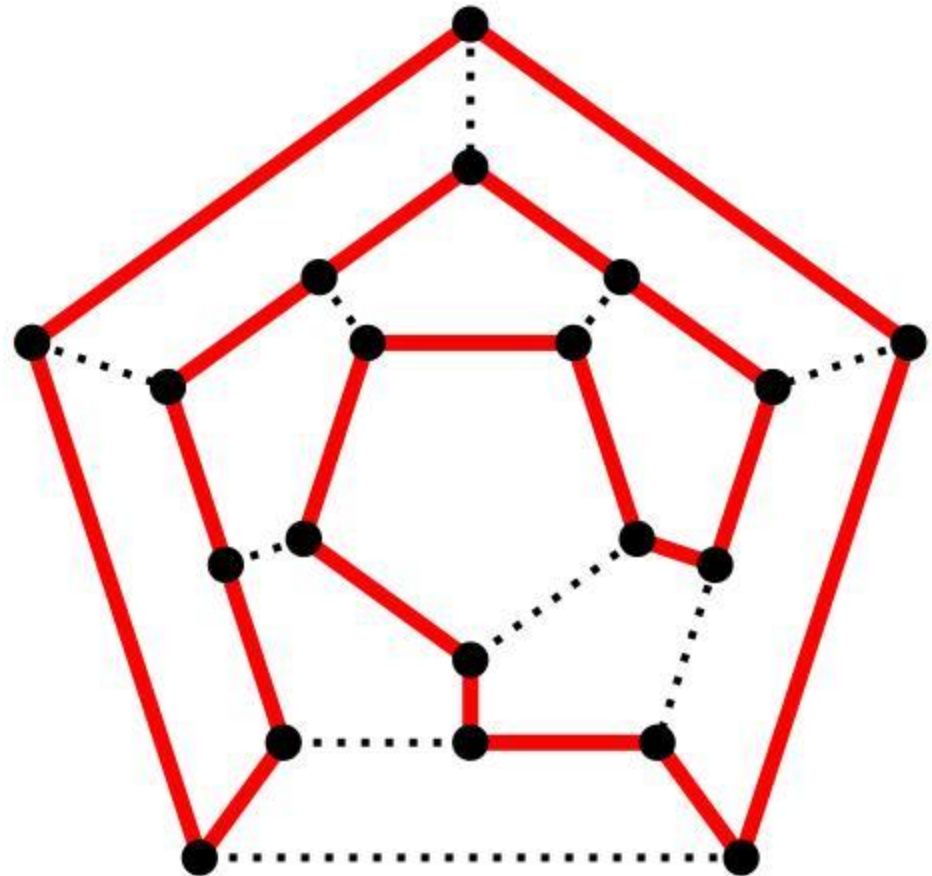
EULER GRAFEN

- Een graaf met een (Euleriaans) spoor dat alle takken precies 1 maal gebruikt
- Een graaf met een Euleriaanse cirkel heeft een gesloten spoor en heet een Euler-graaf
- Knopen in de graaf mogen meermalen bezocht worden en er kunnen losse knopen in voorkomen!!



HAMILTONIAANSE CYCLUS IN EEN GRAAF

- Een hamiltoniaans gesloten pad in een graaf bezoekt alle knopen precies 1 maal
- Niet alle takken hoeven hierbij doorlopen te worden!!
- Figuur van Christoph Sommer (Creative commons toestemming)



Welke wijziging maakt deze graafwandeling tevens tot een Euleriaans pad?

