University of Southern Denmark • Syddansk Universitet

DM503

Forelæsning 7

Indhold

Indhold

- Grafer

Indhold

- Grafer
  - Definitioner

Indhold

- Grafer
  - Definitioner
  - Gennemløb

Indhold

- Grafer
  - Definitioner
  - Gennemløb
    - Dybde-først-søgning

Indhold

- Grafer
  - Definitioner
  - Gennemløb
    - Dybde-først-søgning
    - Brede-først-søgning

Graf

Graf

- En graf (graph) er et par (V,E) hvor

Graf

- En graf (graph) er et par (V,E) hvor
  - V er en mængde af knuder

Graf

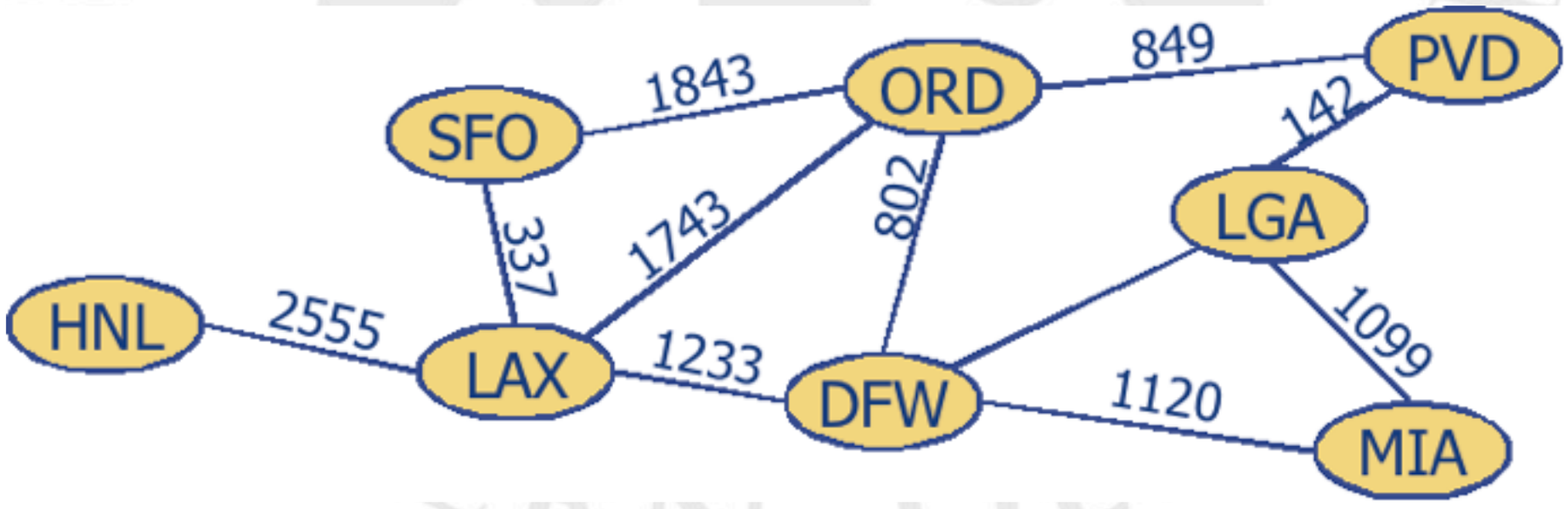
- En graf (graph) er et par (V,E) hvor
  - V er en mængde af knuder
  - E er en samling af par af knuder kaldet kanter

Graf

- En graf (graph) er et par (V,E) hvor
  - V er en mængde af knuder
  - E er en samling af par af knuder kaldet kanter
  - Både knuder og kanter kan indeholde objekter

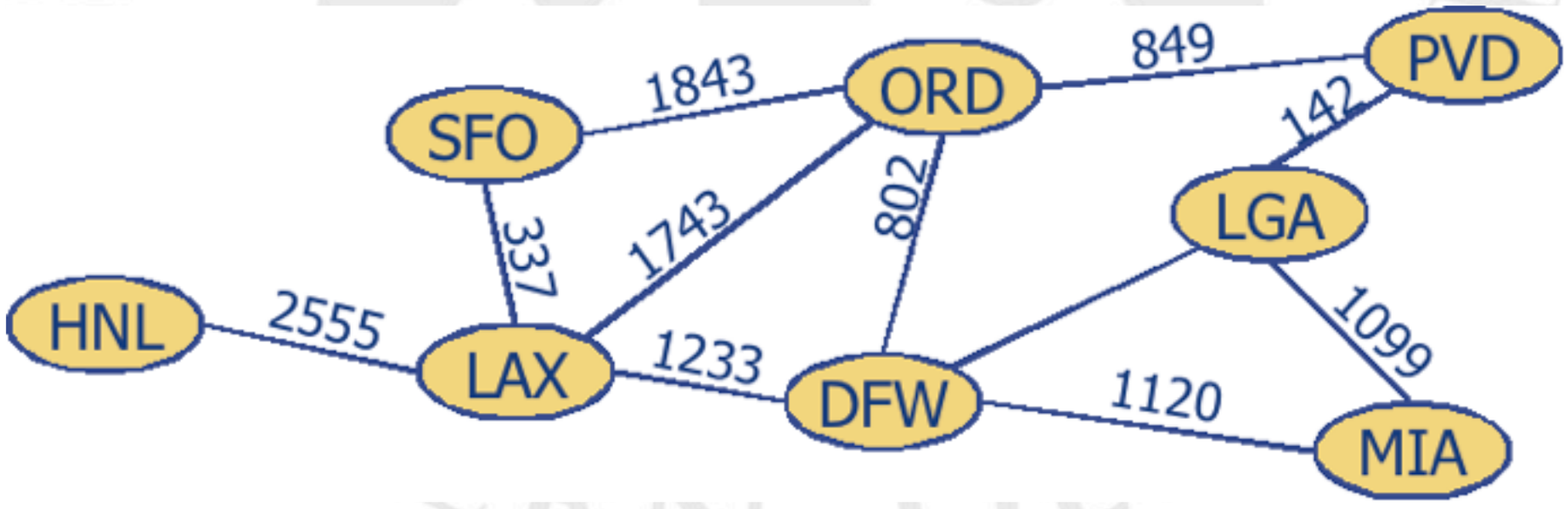
Graf

- En graf (graph) er et par (V,E) hvor
  - V er en mængde af knuder
  - E er en samling af par af knuder kaldet kanter
  - Både knuder og kanter kan indeholde objekter
- Eksempel



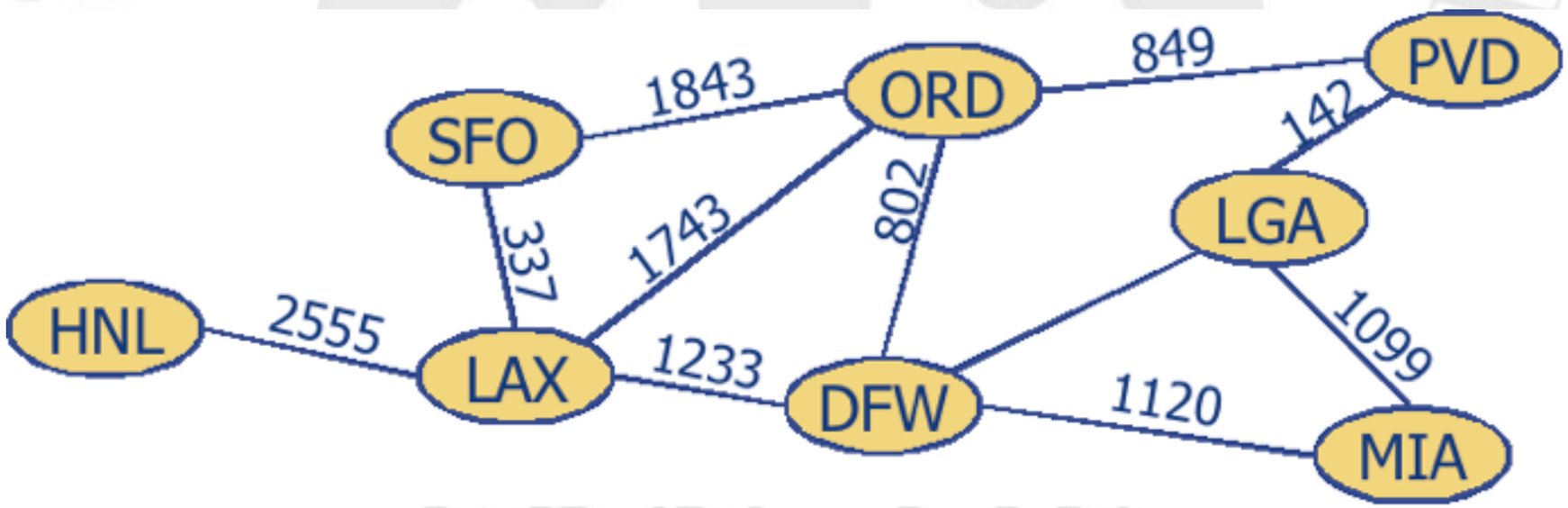
Graf

- En graf (graph) er et par (V,E) hvor
  - V er en mængde af knuder
  - E er en samling af par af knuder kaldet kanter
  - Både knuder og kanter kan indeholde objekter
- Eksempel
  - Knuderne repræsenterer amerikanske lufthavne



Graf

- En graf (graph) er et par (V,E) hvor
  - V er en mængde af knuder
  - E er en samling af par af knuder kaldet kanter
  - Både knuder og kanter kan indeholde objekter
- Eksempel
  - Knuderne repræsenterer amerikanske lufthavne
  - En kant repræsenterer en rute mellem to lufthavne med tilhørende afstand



Kant-typer

Kant-typer

- Orienteret kant

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden
- Ikke-orienteret kant

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden
- Ikke-orienteret kant
  - Et uordnet par (u,v) af knuder

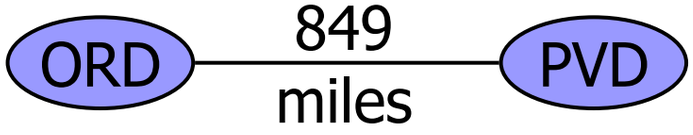
Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden

- Ikke-orienteret kant
  - Et uordnet par (u,v) af knuder
  - Fx en fly-rute

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden
- Ikke-orienteret kant
  - Et uordnet par (u,v) af knuder
  - Fx en fly-rute
- Orienteret graf



Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden
- Ikke-orienteret kant
  - Et uordnet par (u,v) af knuder
  - Fx en fly-rute
- Orienteret graf
  - Alle kanterne i grafen er orienteret

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden
- Ikke-orienteret kant
  - Et uordnet par (u,v) af knuder
  - Fx en fly-rute
- Orienteret graf
  - Alle kanterne i grafen er orienteret
- Ikke-orienteret graf

Kant-typer

- Orienteret kant
  - Et ordnet par (u,v) af knuder
  - Første knude u er start-knuden
  - Anden knude v er slut-knuden
- Ikke-orienteret kant
  - Et uordnet par (u,v) af knuder
  - Fx en fly-rute
- Orienteret graf
  - Alle kanterne i grafen er orienteret
- Ikke-orienteret graf
  - Alle kanterne er ikke-orienterede

Anvendelser

Anvendelser

- Elektronik
  - Printkort

Anvendelser

- Elektronik
  - Printkort
- Transport-netværk
  - Vej-netværk
  - Fly-netværk

Anvendelser

- Elektronik
  - Printkort
- Transport-netværk
  - Vej-netværk
  - Fly-netværk
- Computer-netværk
  - LAN (Local Area Network)
  - Internet

Anvendelser

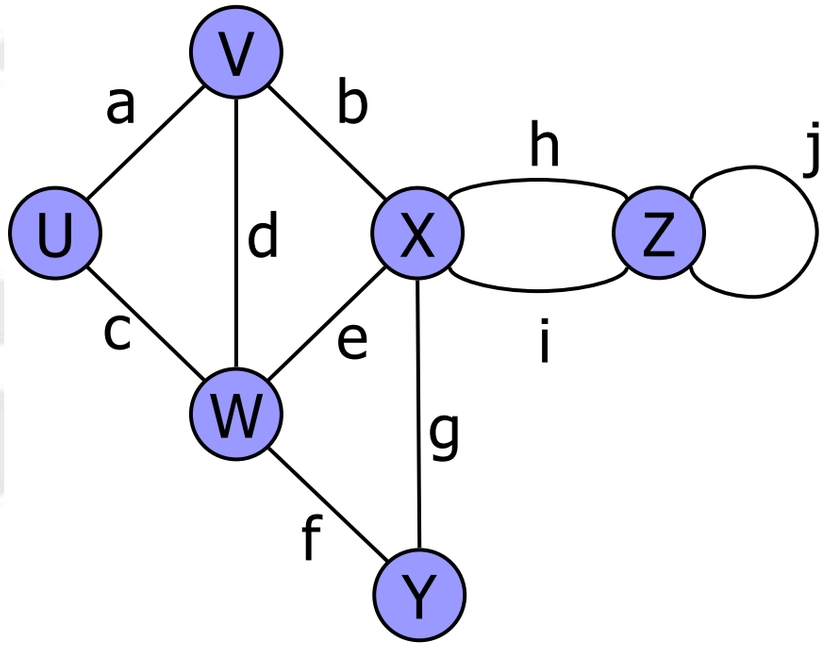
- Elektronik
  - Printkort
- Transport-netværk
  - Vej-netværk
  - Fly-netværk
- Computer-netværk
  - LAN (Local Area Network)
  - Internet
- Relationer
  - Socialt netværk (six degrees of separation)

Anvendelser

- Elektronik
  - Printkort
- Transport-netværk
  - Vej-netværk
  - Fly-netværk
- Computer-netværk
  - LAN (Local Area Network)
  - Internet
- Relationer
  - Socialt netværk (six degrees of separation)
- ...

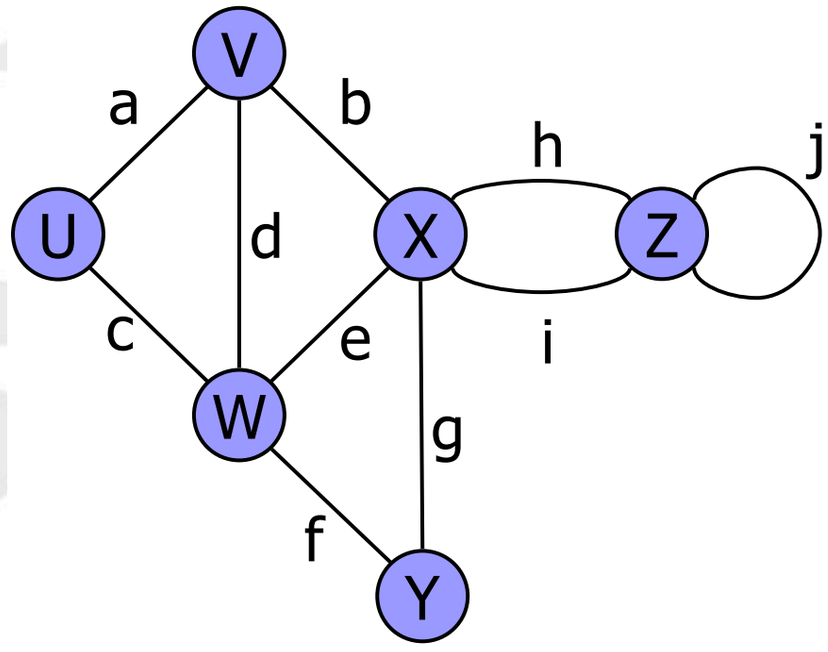
Terminologi

Terminologi



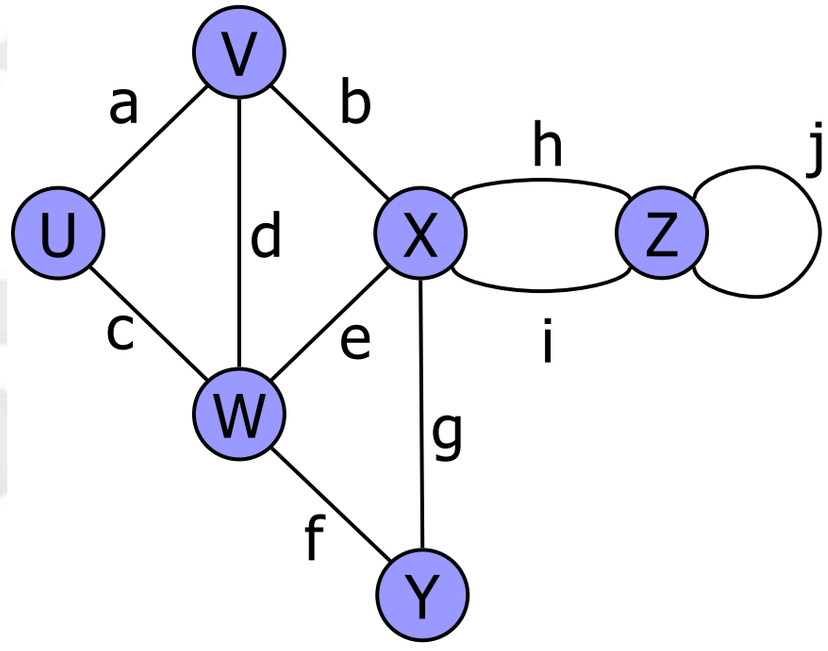
Terminologi

- Endepunkter af en kant



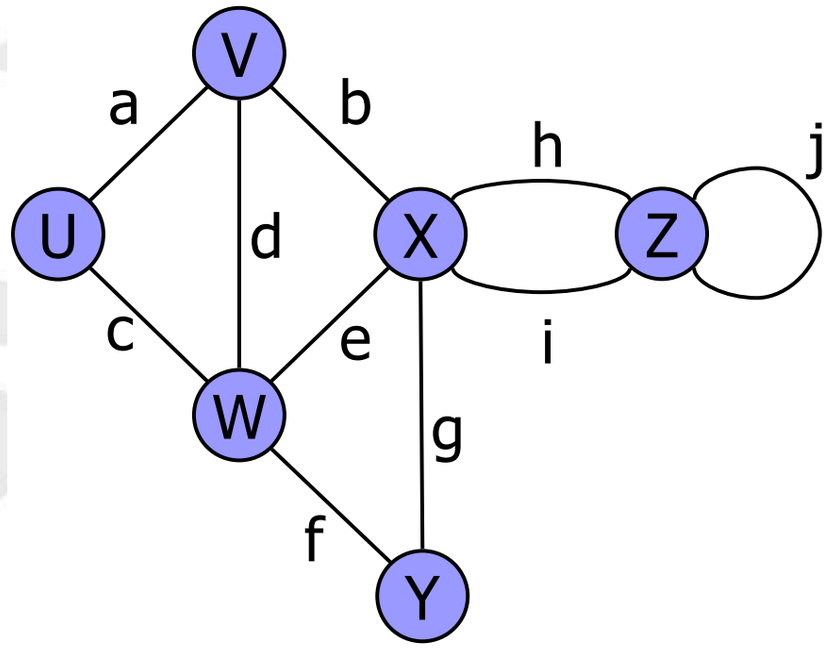
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a



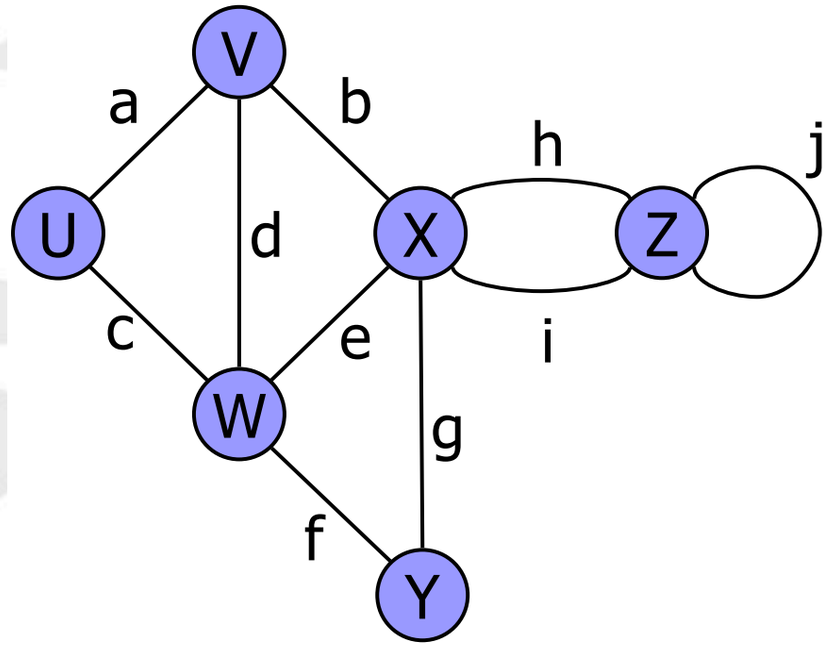
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude



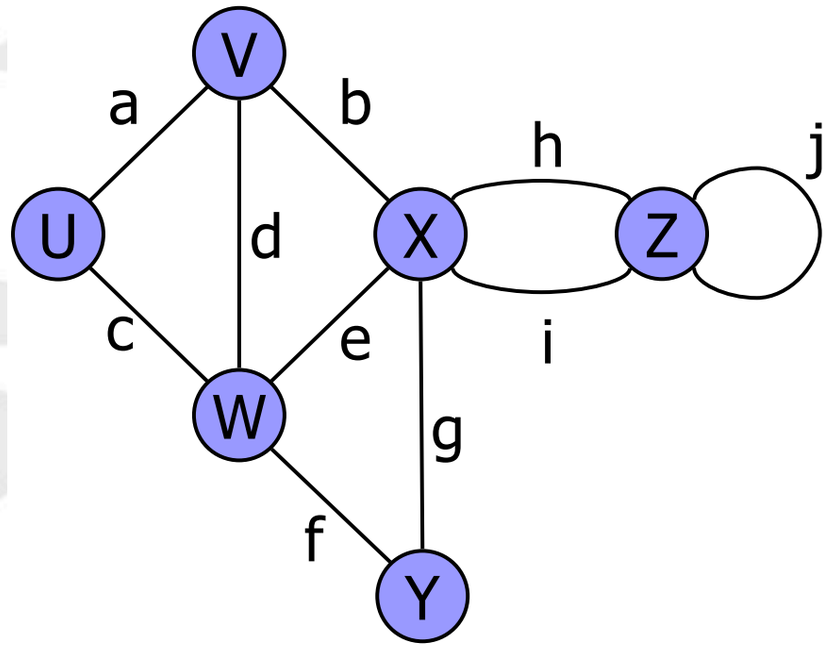
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V



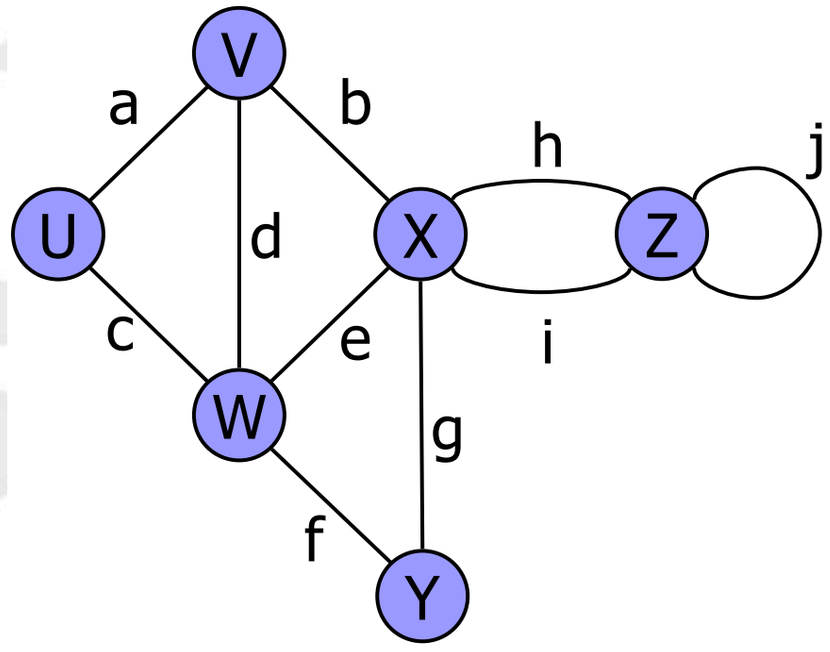
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder



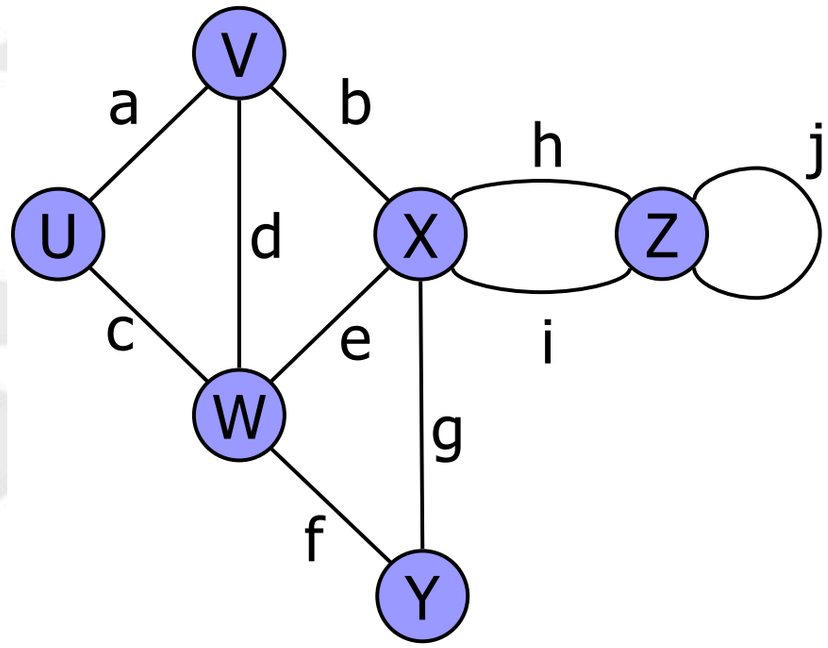
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder
  - U og V er naboer



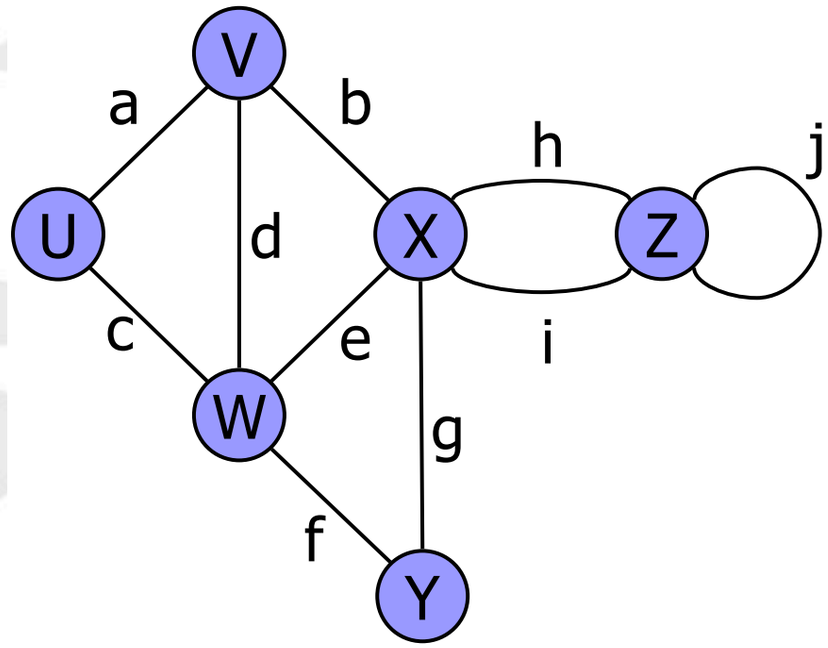
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder
  - U og V er naboer
- Valens af en knude



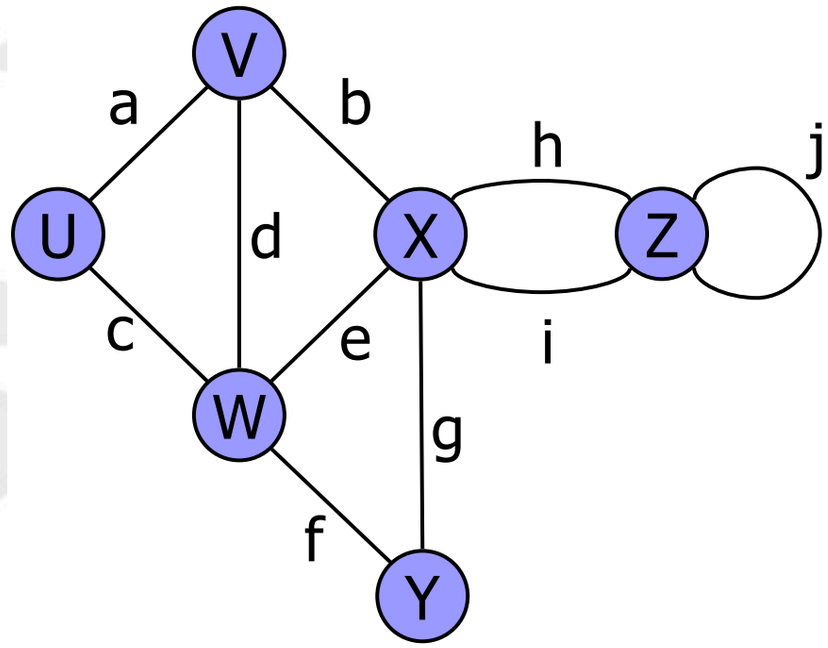
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder
  - U og V er naboer
- Valens af en knude
  - X har valens 5



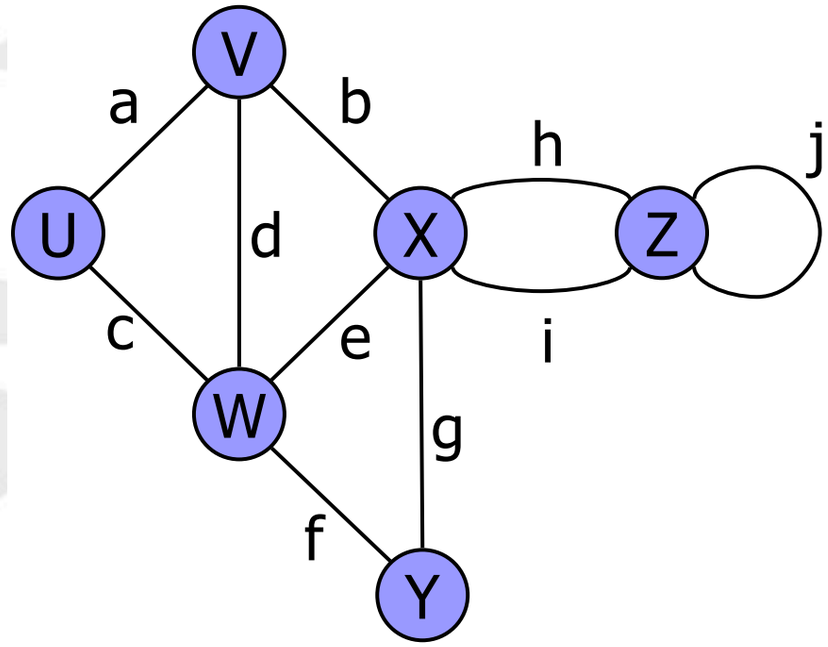
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder
  - U og V er naboer
- Valens af en knude
  - X har valens 5
- Parallelle kanter



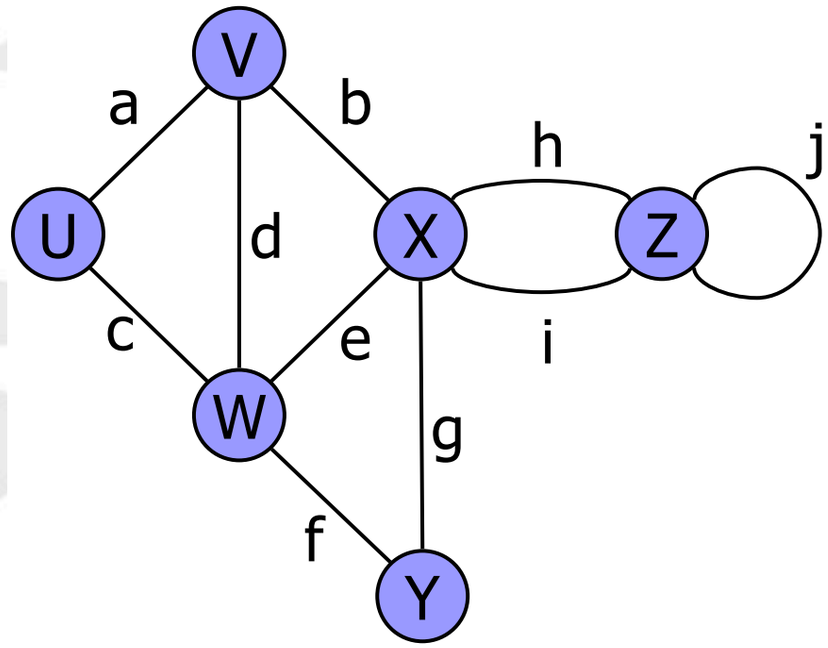
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder
  - U og V er naboer
- Valens af en knude
  - X har valens 5
- Parallelle kanter
  - h og i er parallelle kanter



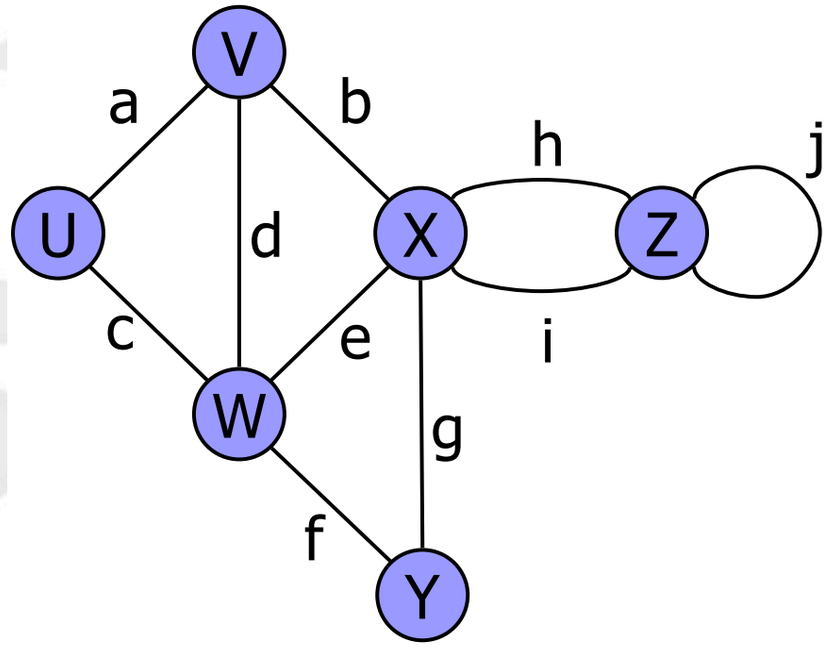
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder
  - U og V er naboer
- Valens af en knude
  - X har valens 5
- Parallelle kanter
  - h og i er parallelle kanter
- Sløjfe



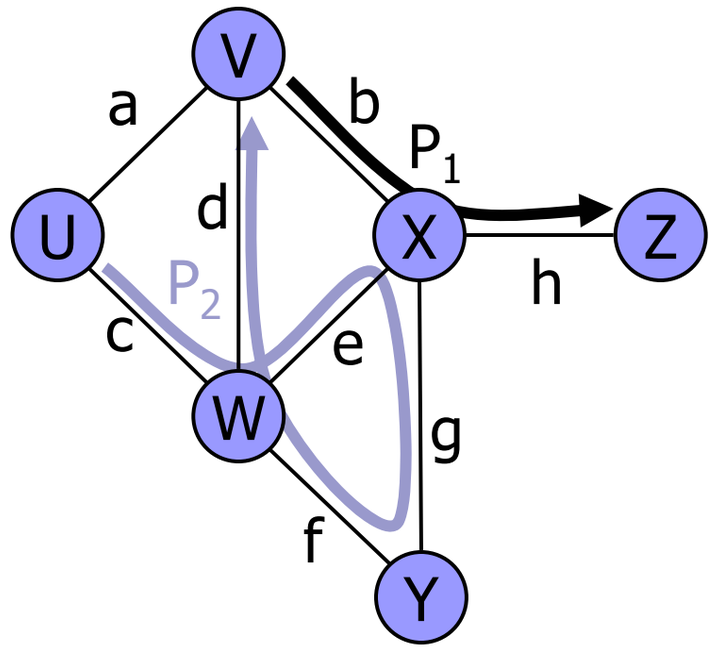
Terminologi

- Endepunkter af en kant
  - U og V er endepunkter af a
- Kanter er incidente med en knude
  - a, d og b er incidente med V
- Nabo-knuder
  - U og V er naboer
- Valens af en knude
  - X har valens 5
- Parallelle kanter
  - h og i er parallelle kanter
- Sløjfe
  - Kanten j er en sløjfe



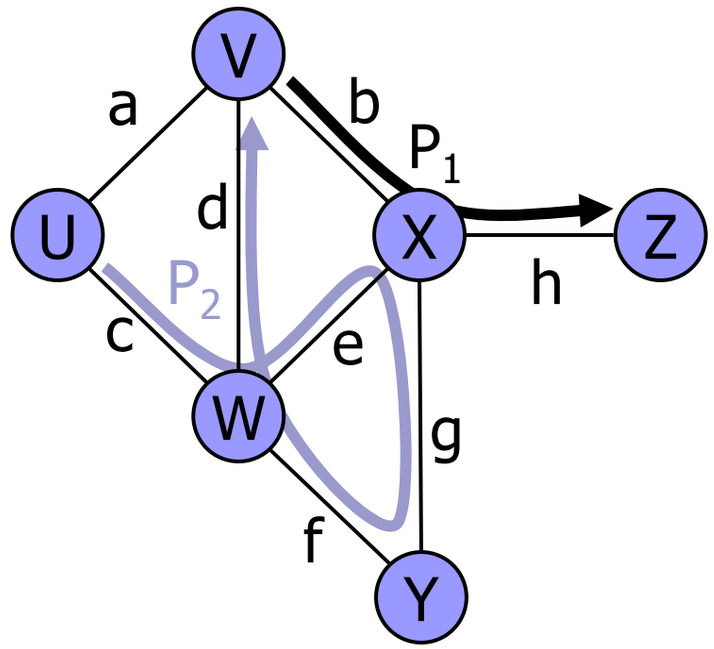
Terminologi

Terminologi



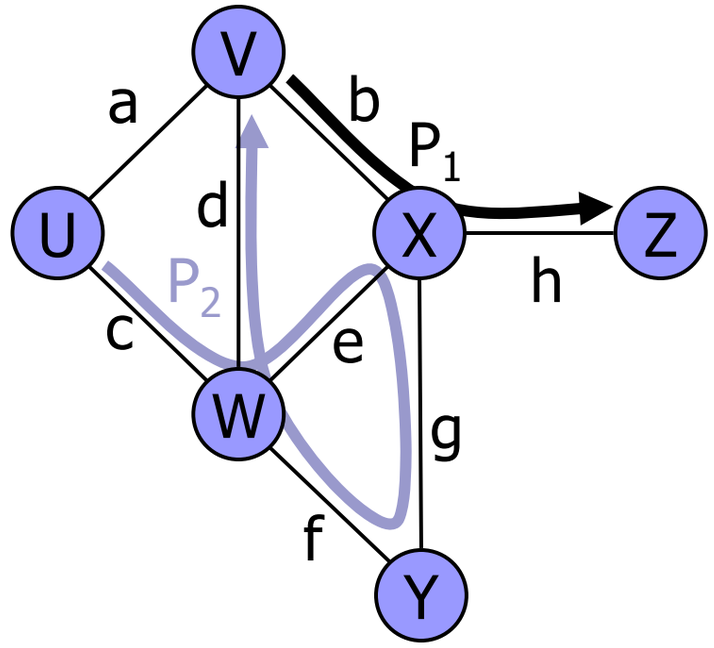
Terminologi

- Sti



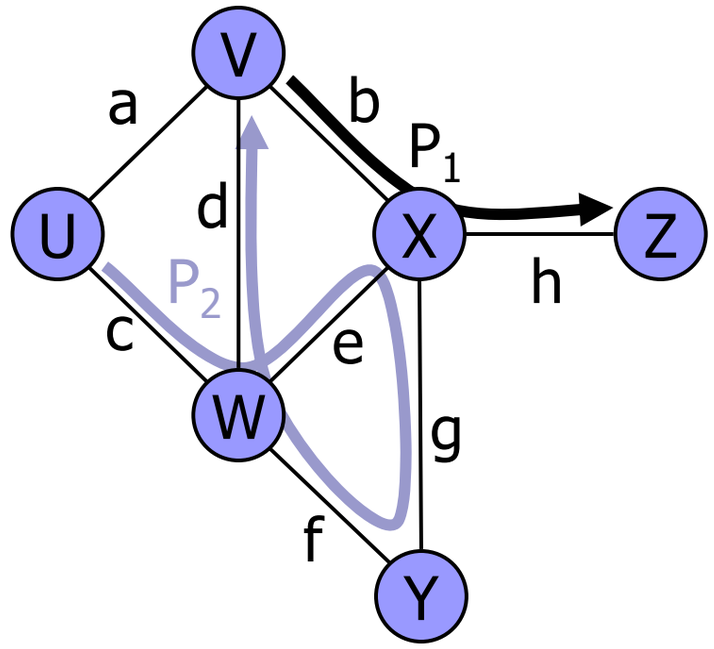
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter



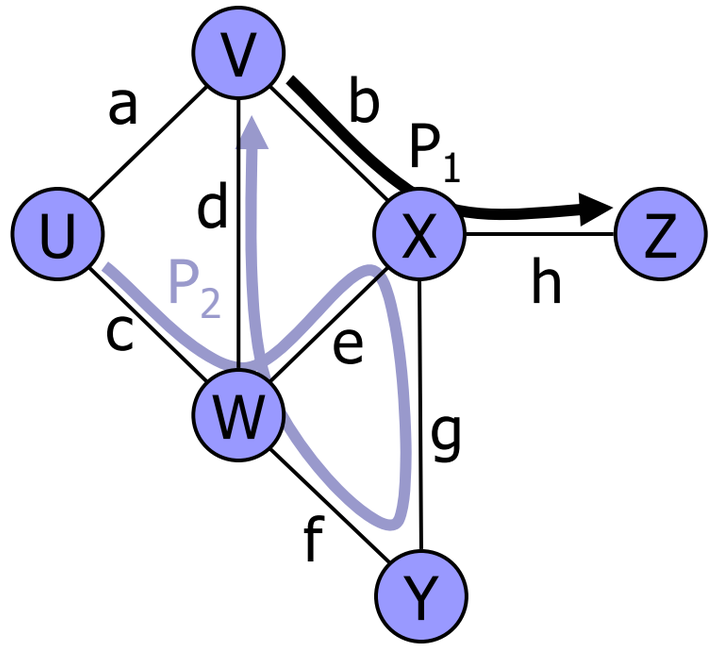
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude



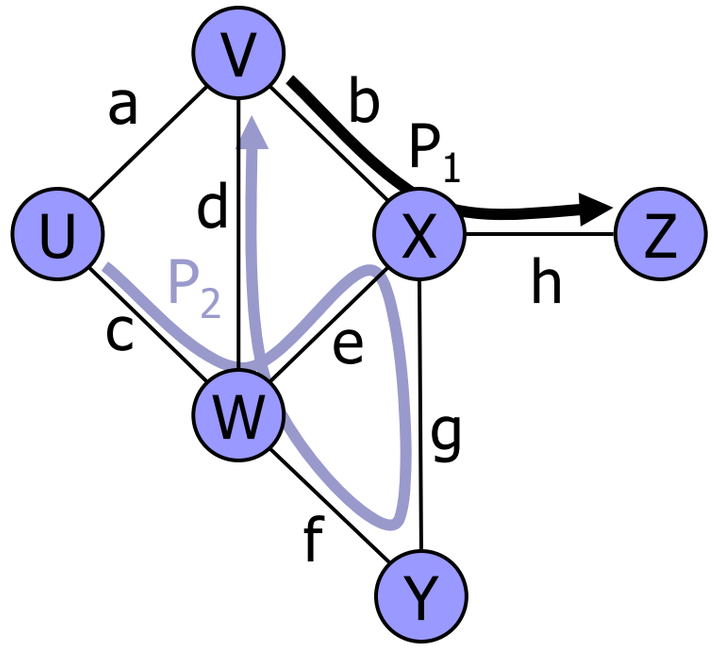
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude
  - Slutter i en knude



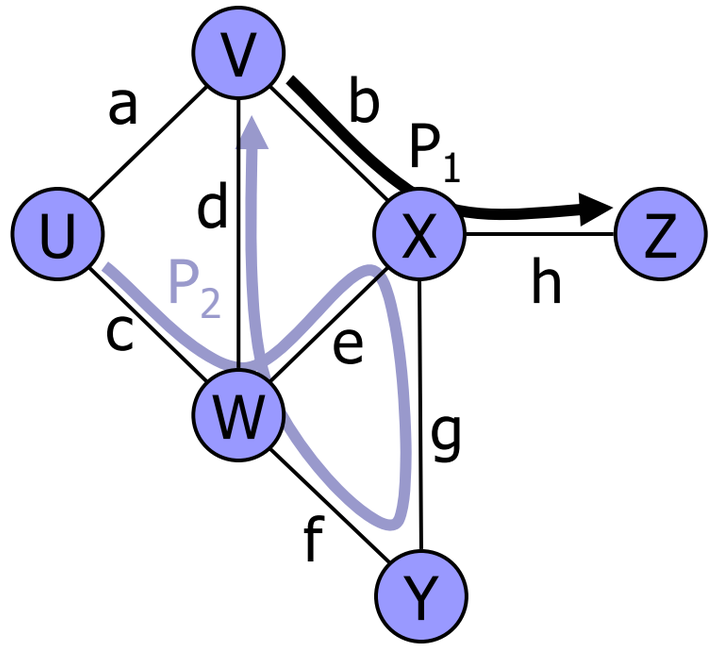
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude
  - Slutter i en knude
  - Hver kant starter i et endepunkt og slutter i det andet endepunkt



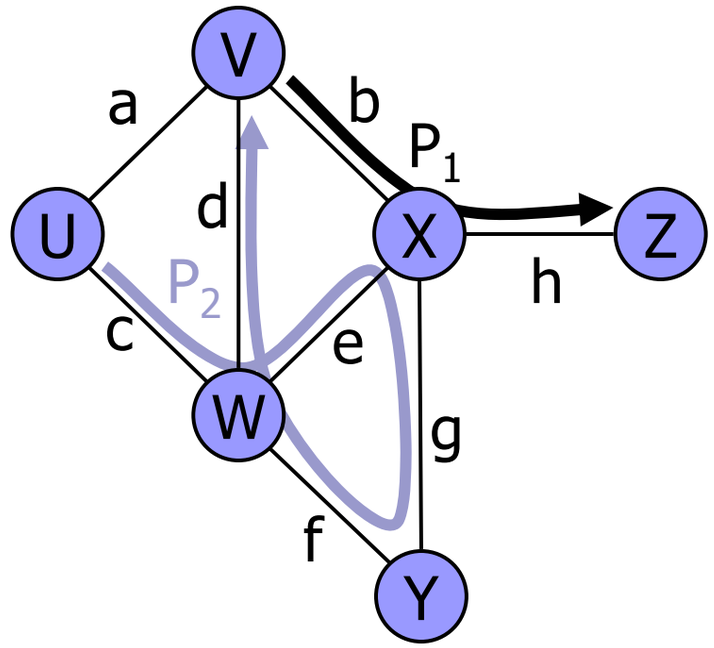
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude
  - Slutter i en knude
  - Hver kant starter i et endepunkt og slutter i det andet endepunkt
- Simpel sti



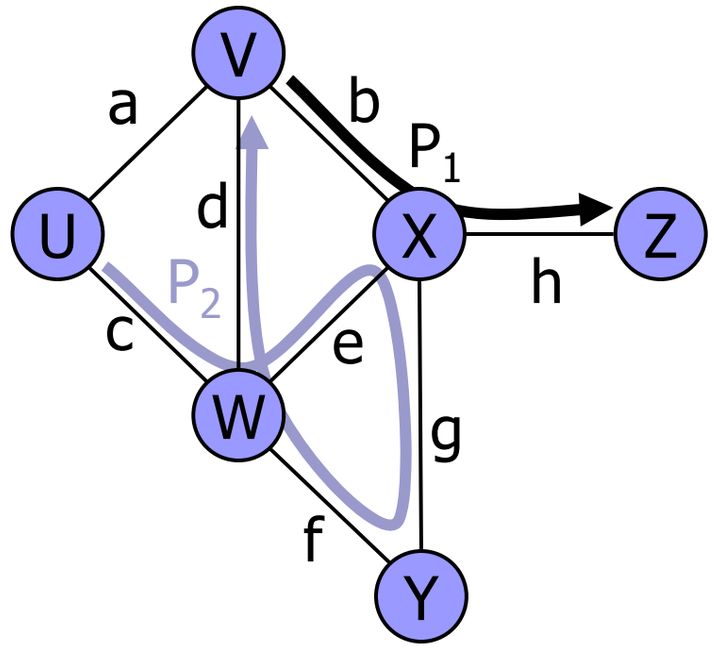
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude
  - Slutter i en knude
  - Hver kant starter i et endepunkt og slutter i det andet endepunkt
- Simpel sti
  - En sti uden gentagelser af knuder og kanter



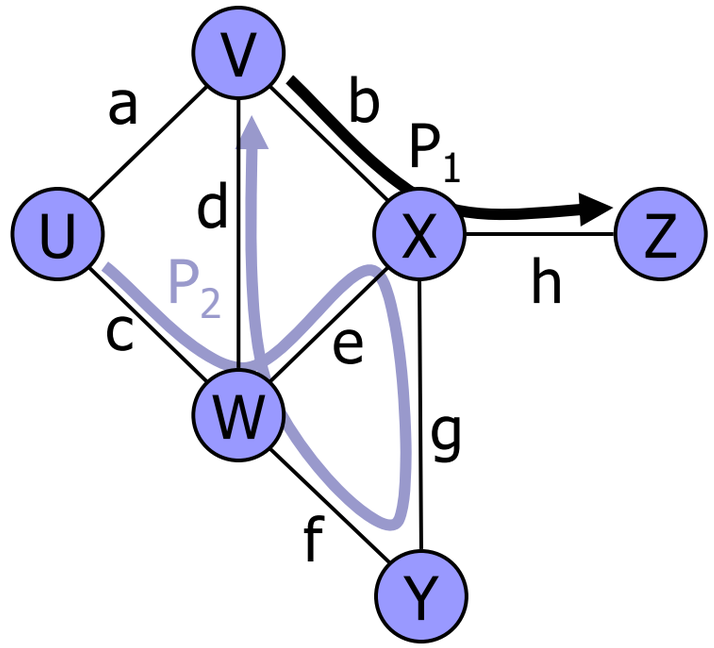
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude
  - Slutter i en knude
  - Hver kant starter i et endepunkt og slutter i det andet endepunkt
- Simpel sti
  - En sti uden gentagelser af knuder og kanter
- Eksempel



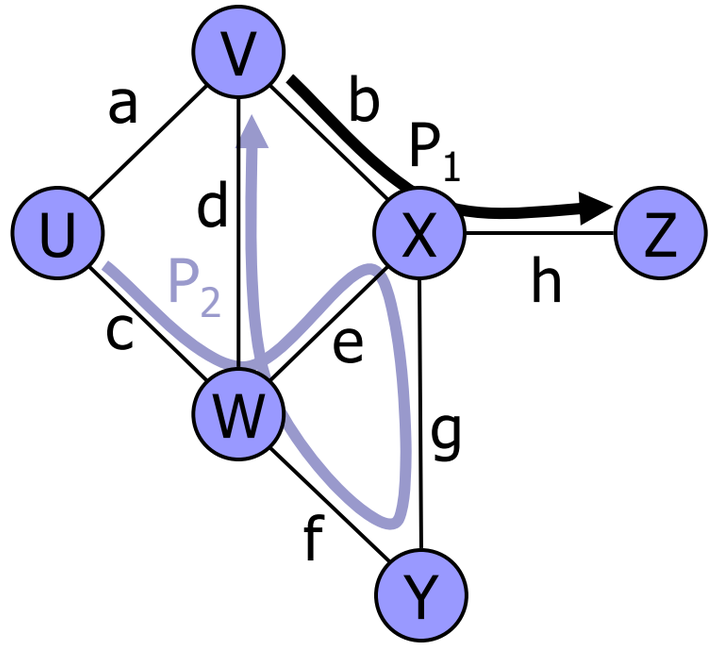
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude
  - Slutter i en knude
  - Hver kant starter i et endepunkt og slutter i det andet endepunkt
- Simpel sti
  - En sti uden gentagelser af knuder og kanter
- Eksempel
  - (V,b,X,h,Z) er en simpel sti



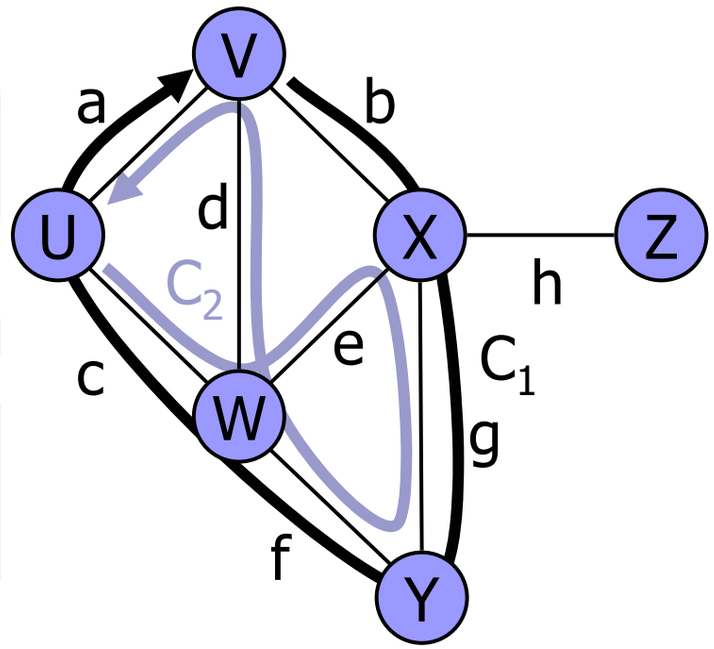
Terminologi

- Sti
  - Sekvens af alternerende knuder og kanter
  - Begynder i en knude
  - Slutter i en knude
  - Hver kant starter i et endepunkt og slutter i det andet endepunkt
- Simpel sti
  - En sti uden gentagelser af knuder og kanter
- Eksempel
  - (V,b,X,h,Z) er en simpel sti
  - (U,c,W,e,X,g,Y,f,W,d,V) er en (ikke-simpel) sti



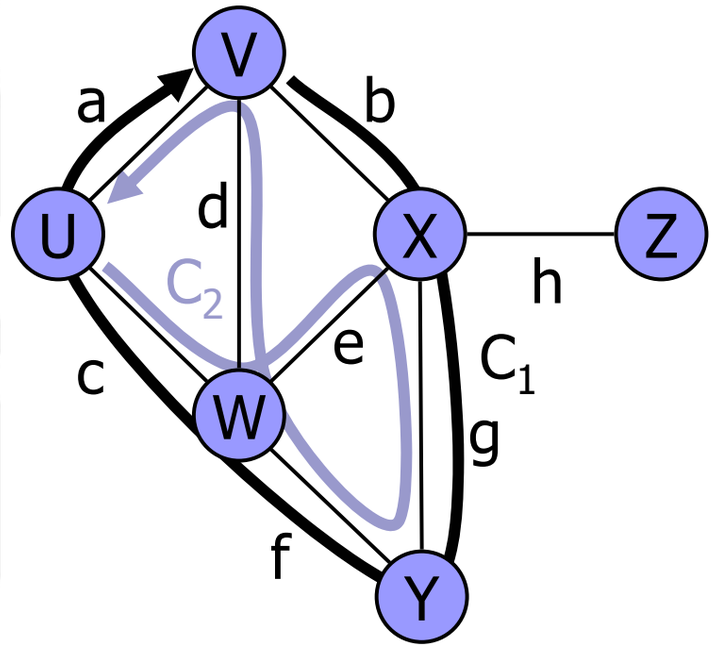
Terminologi

Terminologi



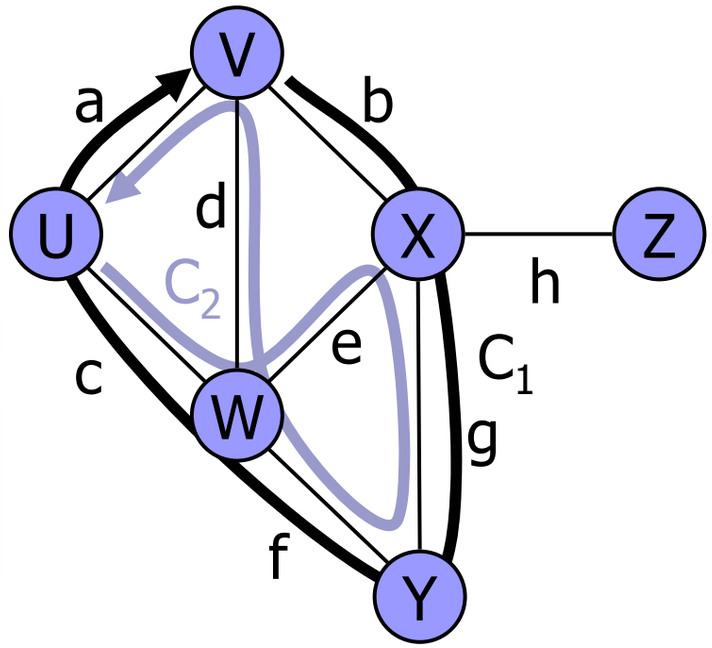
Terminologi

- Kreds



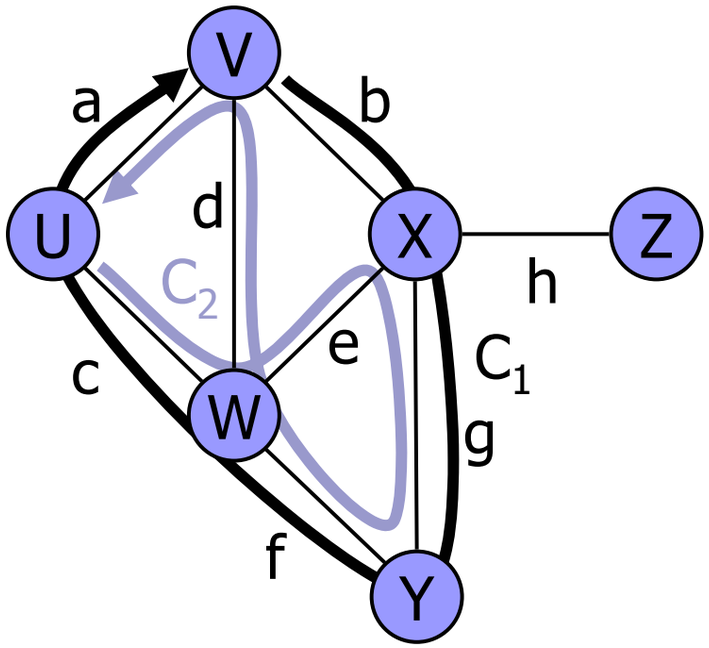
Terminologi

- Kreds
  - En sti der starter og slutter i samme knude



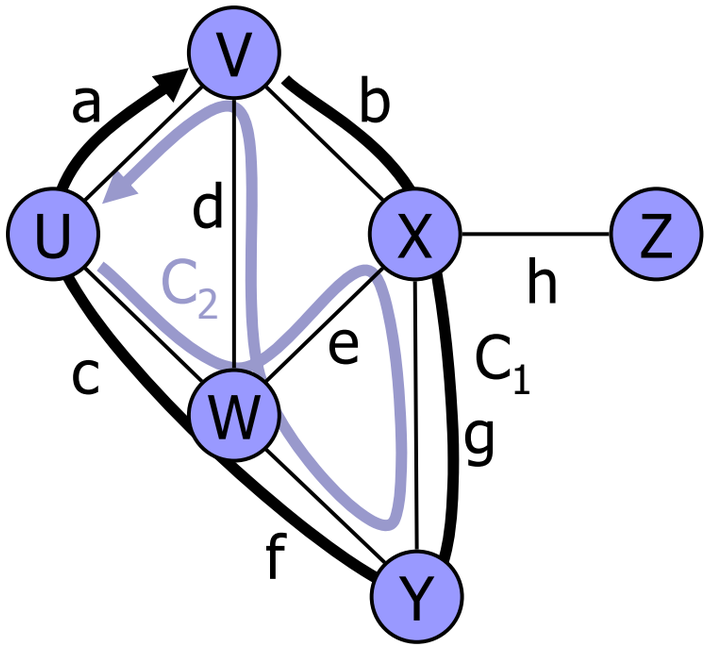
Terminologi

- Kreds
  - En sti der starter og slutter i samme knude
- Simpel kreds



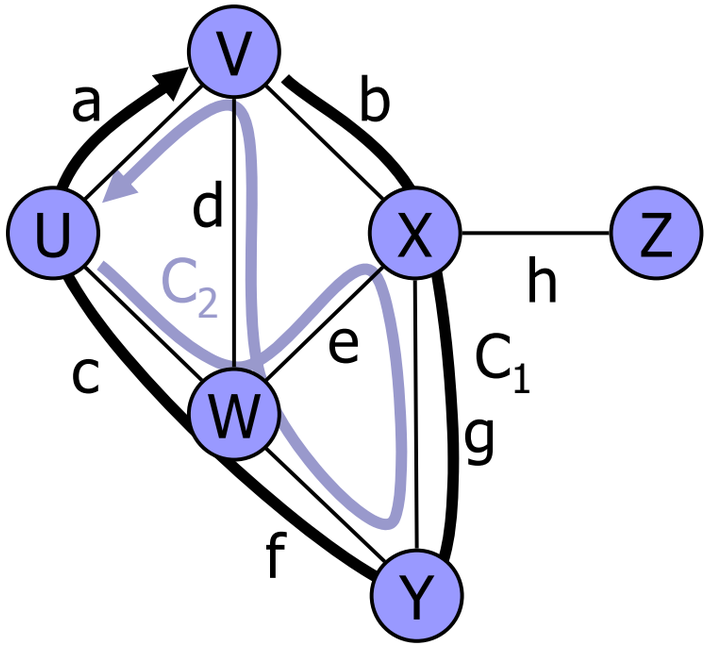
Terminologi

- Kreds
  - En sti der starter og slutter i samme knude
- Simpel kreds
  - En kreds så alle knuder og kanter er forskellige



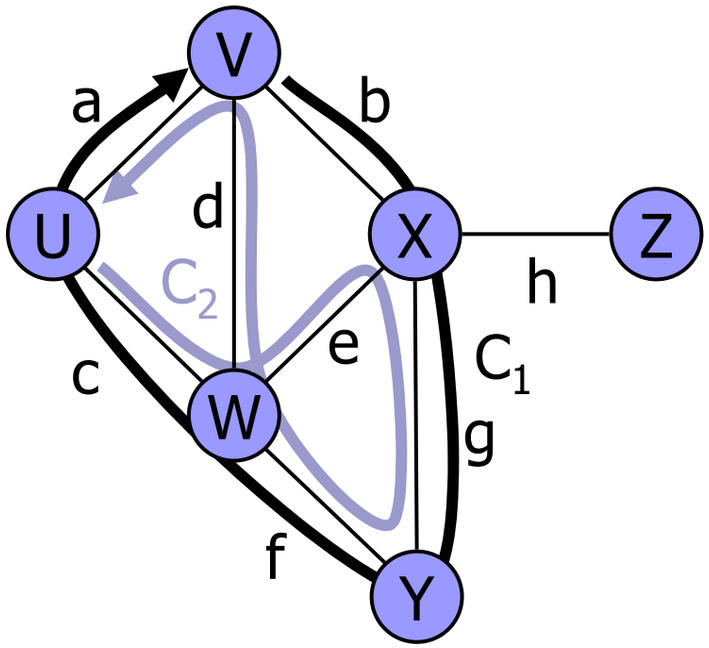
Terminologi

- Kreds
  - En sti der starter og slutter i samme knude
- Simpel kreds
  - En kreds så alle knuder og kanter er forskellige
- Simpel graf



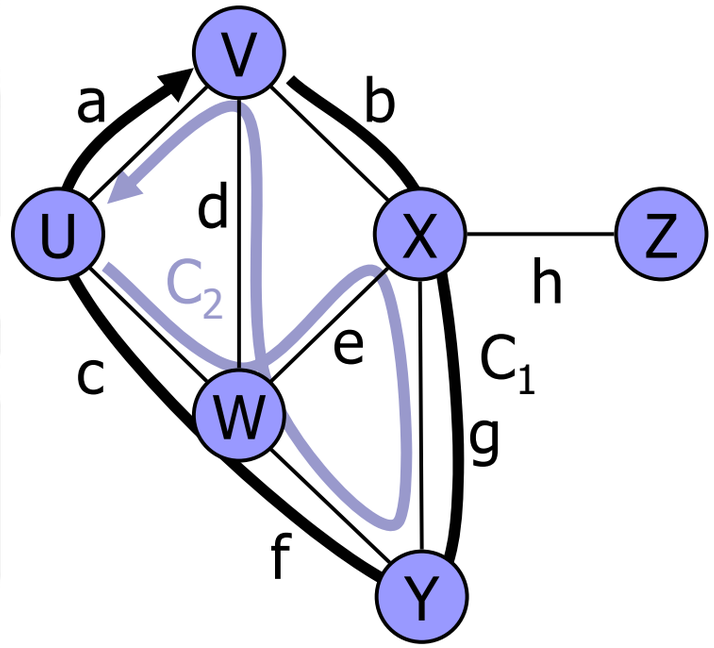
Terminologi

- Kreds
  - En sti der starter og slutter i samme knude
- Simpel kreds
  - En kreds så alle knuder og kanter er forskellige
- Simpel graf
  - En ikke-orienteret graf uden sløjfer og parallelle kanter



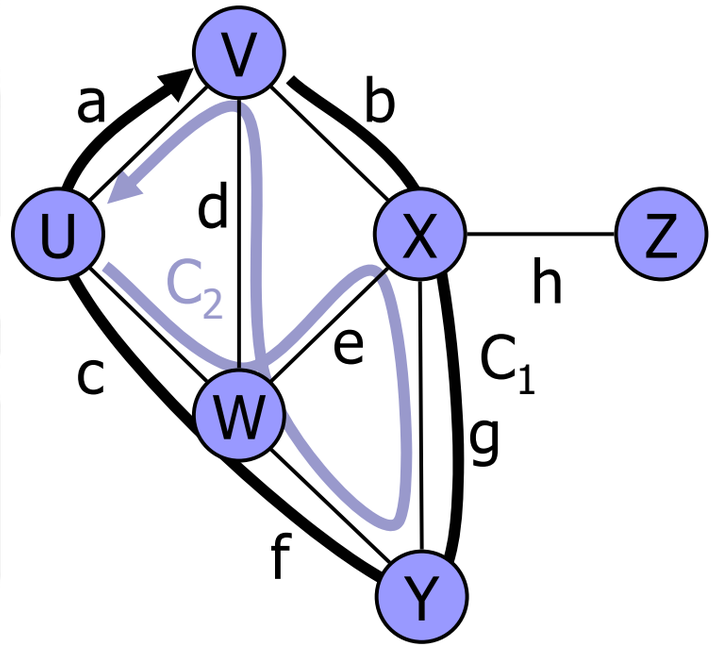
Terminologi

- Kreds
  - En sti der starter og slutter i samme knude
- Simpel kreds
  - En kreds så alle knuder og kanter er forskellige
- Simpel graf
  - En ikke-orienteret graf uden sløjfer og parallelle kanter
- Eksempel



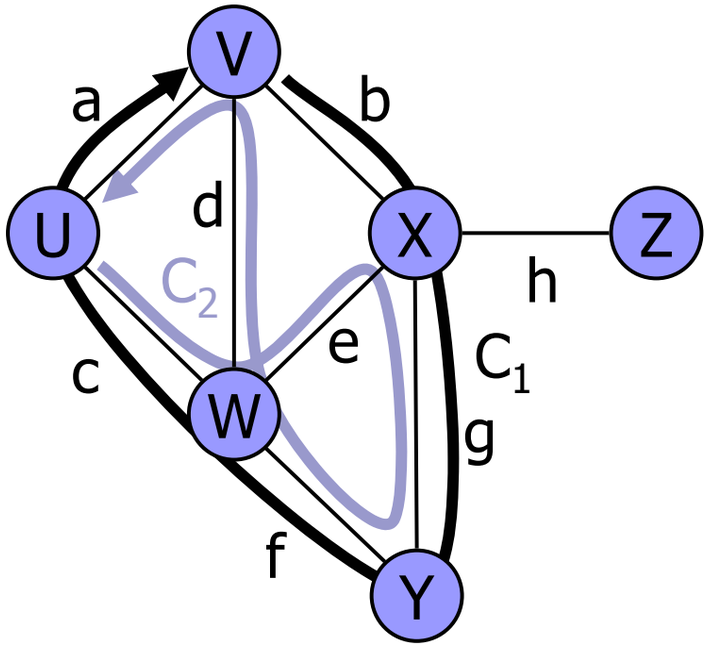
Terminologi

- Kreds
  - En sti der starter og slutter i samme knude
- Simpel kreds
  - En kreds så alle knuder og kanter er forskellige
- Simpel graf
  - En ikke-orienteret graf uden sløjfer og parallelle kanter
- Eksempel
  - (V,b,X,g,Y,f,W,c,U,a,-) er en simpel sti



Terminologi

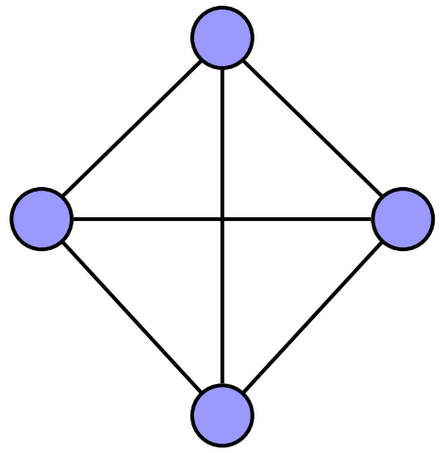
- Kreds
  - En sti der starter og slutter i samme knude
- Simpel kreds
  - En kreds så alle knuder og kanter er forskellige
- Simpel graf
  - En ikke-orienteret graf uden sløjfer og parallelle kanter
- Eksempel
  - (V,b,X,g,Y,f,W,c,U,a,-) er en simpel sti
  - (U,c,W,e,X,g,Y,f,W,d,V,a,-) er en (ikke-simpel) kreds



Egenskaber

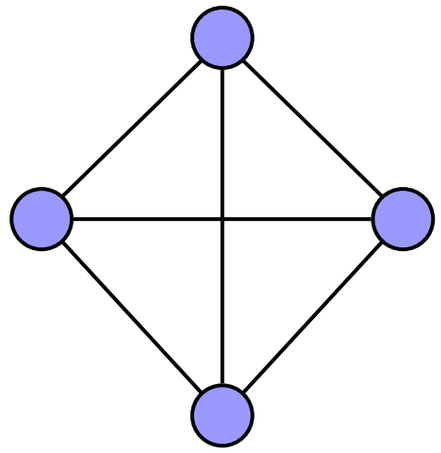
Egenskaber

- Notation
  - n : antallet af knuder
  - m : antallet af kanter
  - deg(v) : valensen af knude v



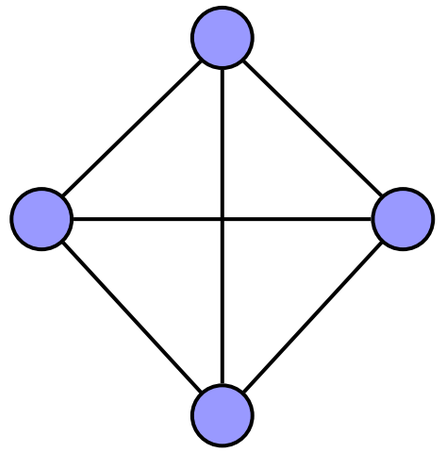
Egenskaber

- Notation
  - n : antallet af knuder
  - m : antallet af kanter
  - deg(v) : valensen af knude v
- Egenskab 1



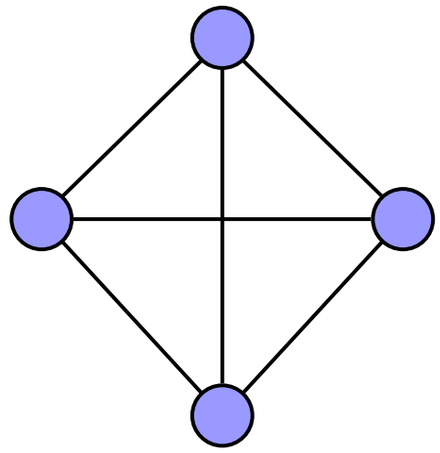
Egenskaber

- Notation
  - n : antallet af knuder
  - m : antallet af kanter
  - deg(v) : valensen af knude v
- Egenskab 1
  - $\sum_v deg(v) = 2m$



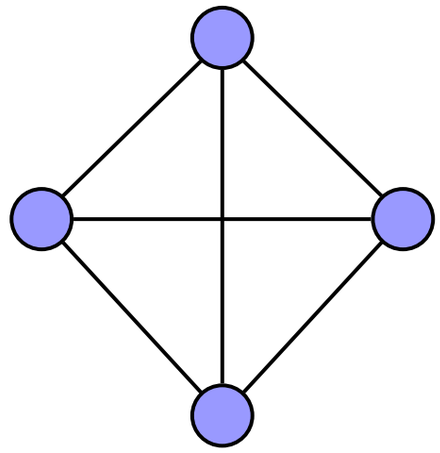
Egenskaber

- Notation
  - n : antallet af knuder
  - m : antallet af kanter
  - deg(v) : valensen af knude v
- Egenskab 1
  - $\sum_v deg(v) = 2m$
- Egenskab 2



Egenskaber

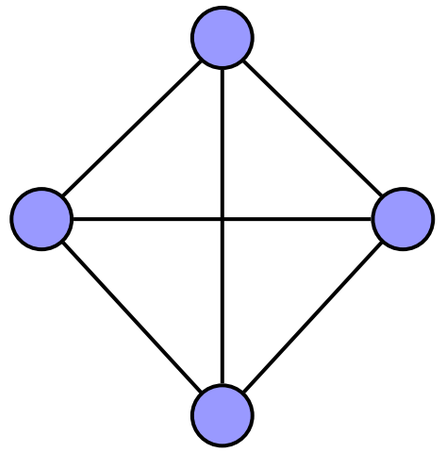
- Notation
  - n : antallet af knuder
  - m : antallet af kanter
  - deg(v) : valensen af knude v
- Egenskab 1
  - $\sum_v deg(v) = 2m$



- Egenskab 2
  - I en simpel graf er $m \leq n(n-1)/2$

Egenskaber

- Notation
  - n : antallet af knuder
  - m : antallet af kanter
  - deg(v) : valensen af knude v
- Egenskab 1
  - $\sum_v deg(v) = 2m$



- Egenskab 2
  - I en simpel graf er $m \leq n(n-1)/2$
  - Hvad er grænsen i en simpel orienteret graf?

Vigtige metoder på en graf

Vigtige metoder på en graf

- Aflæsninger

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)
  - removeVertex(v)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)
  - removeVertex(v)
  - removeEdge(e)

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)
  - removeVertex(v)
  - removeEdge(e)
- Andre

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)
  - removeVertex(v)
  - removeEdge(e)
- Andre
  - numVertices()

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)
  - removeVertex(v)
  - removeEdge(e)
- Andre
  - numVertices()
  - numEdges()

Vigtige metoder på en graf

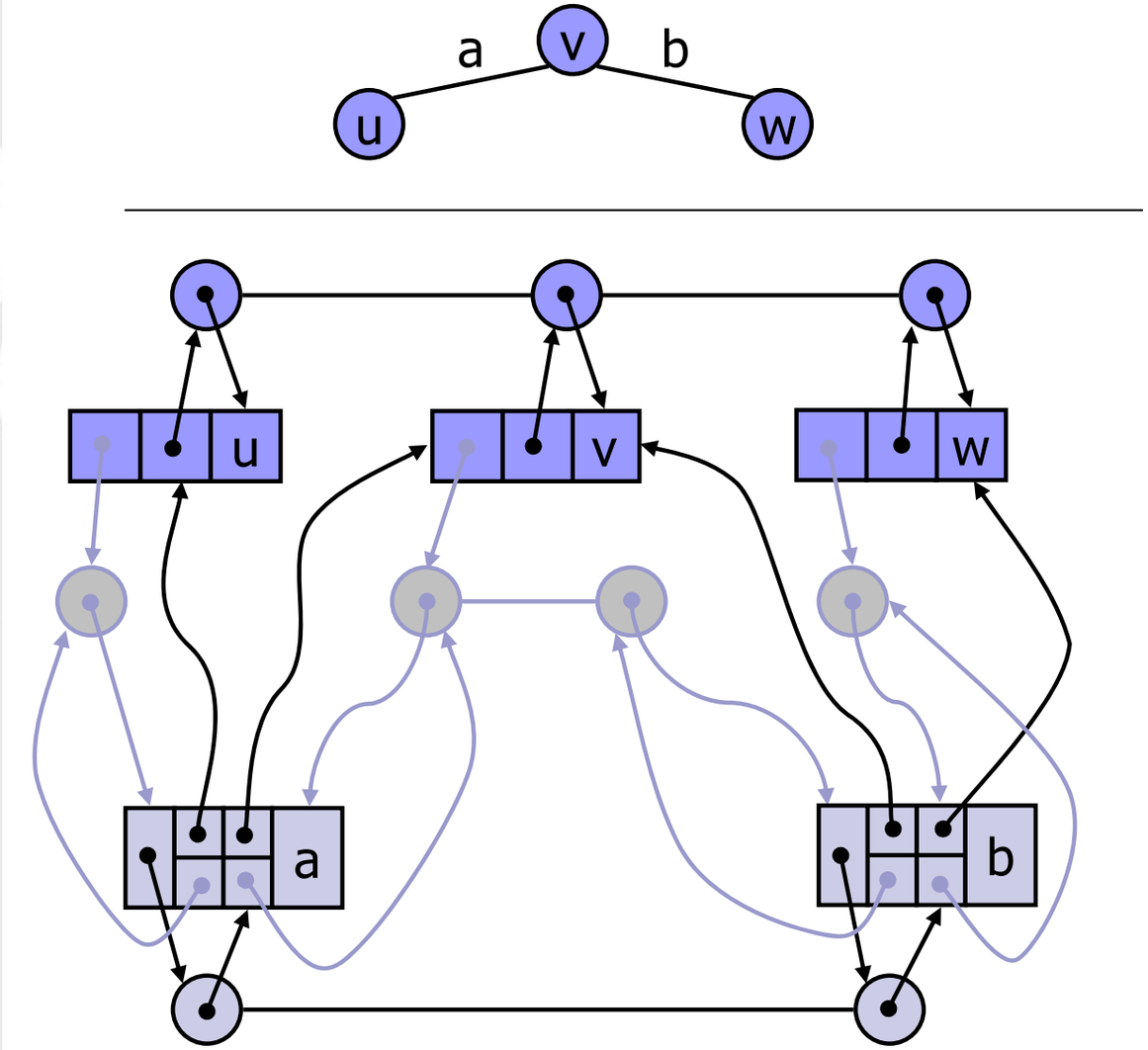
- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)
  - removeVertex(v)
  - removeEdge(e)
- Andre
  - numVertices()
  - numEdges()
  - vertices()

Vigtige metoder på en graf

- Aflæsninger
  - incidentEdges(v)
  - endVertices(e)
  - isDirected(e)
  - origin(e)
  - destination(e)
  - opposite(v, e)
  - areAdjacent(v, w)
- Opdateringer
  - insertVertex(o)
  - insertEdge(v, w, o)
  - insertDirectedEdge(v, w, o)
  - removeVertex(v)
  - removeEdge(e)
- Andre
  - numVertices()
  - numEdges()
  - vertices()
  - edges()

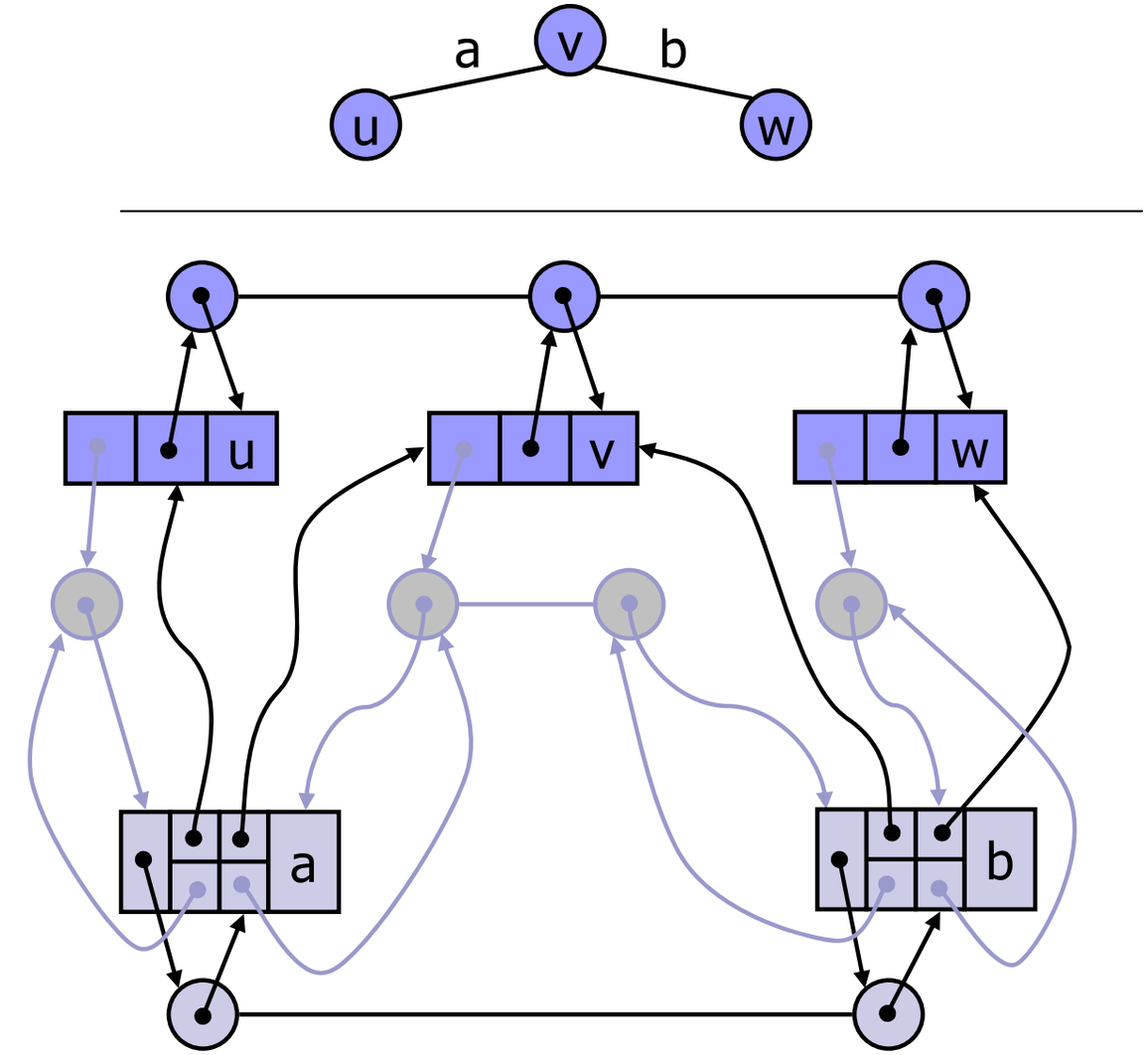
Nabolister

Nabolister



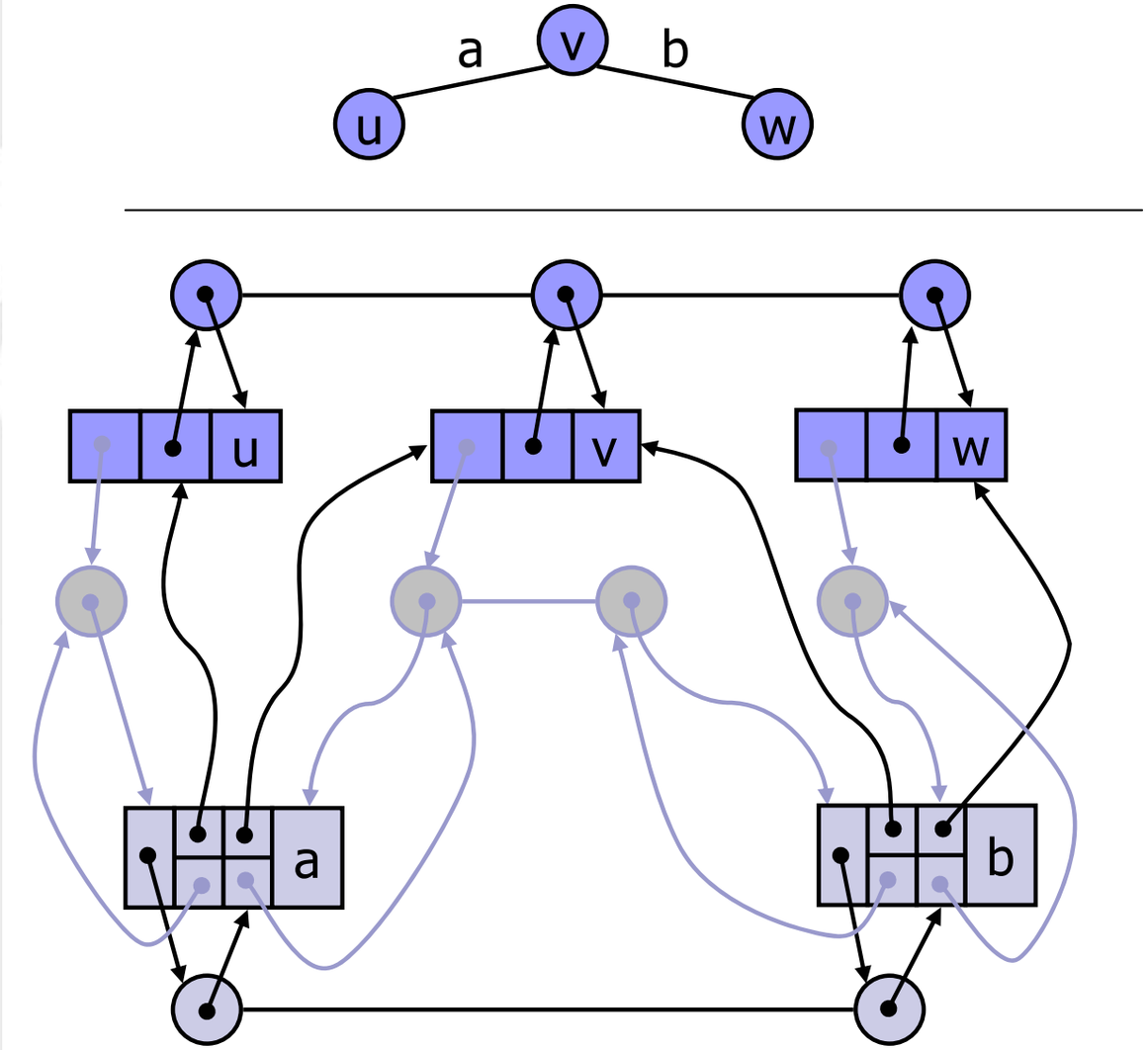
Nabolister

- En liste af kanter



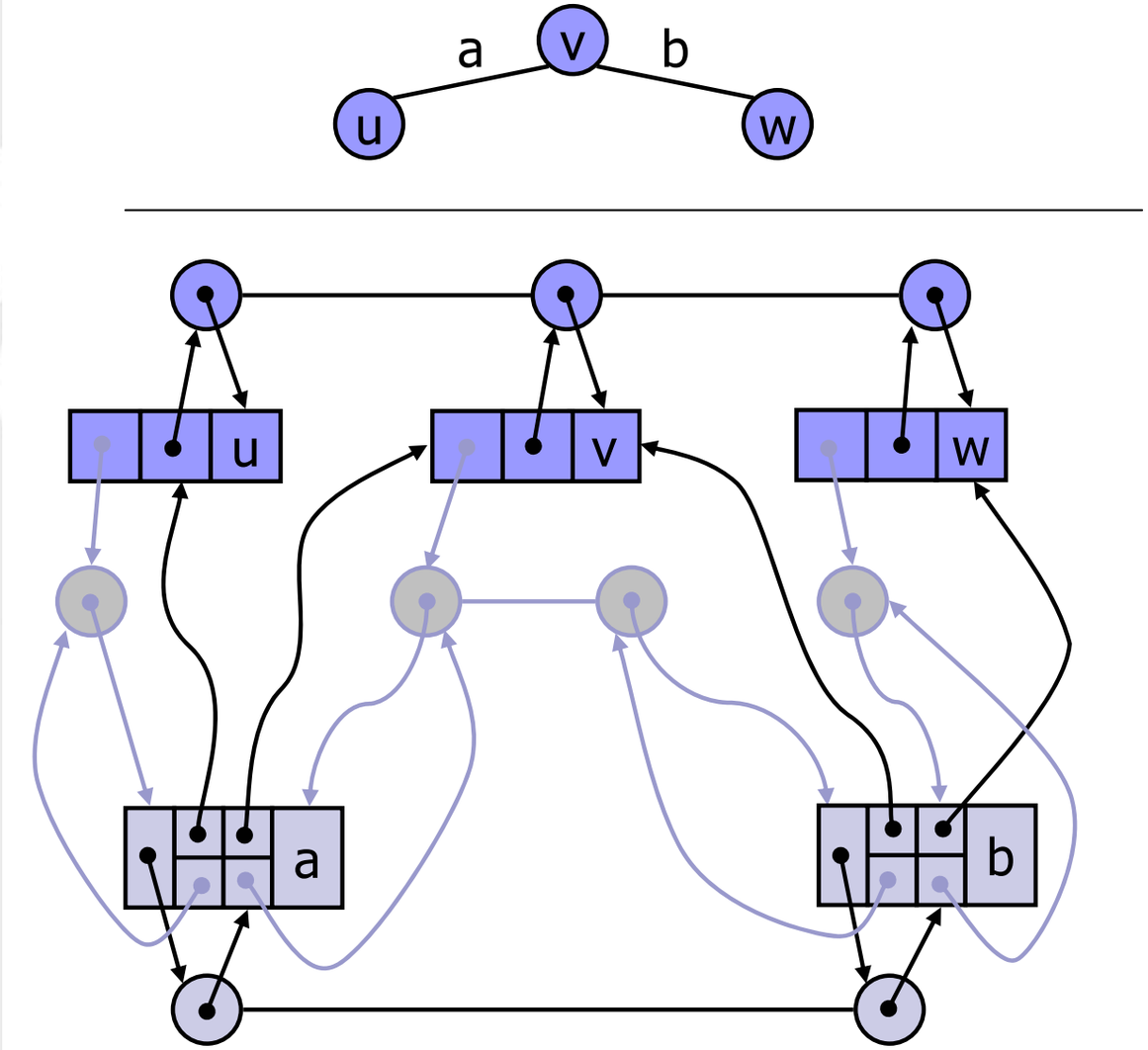
Nabolister

- En liste af kanter
- En liste af knuder



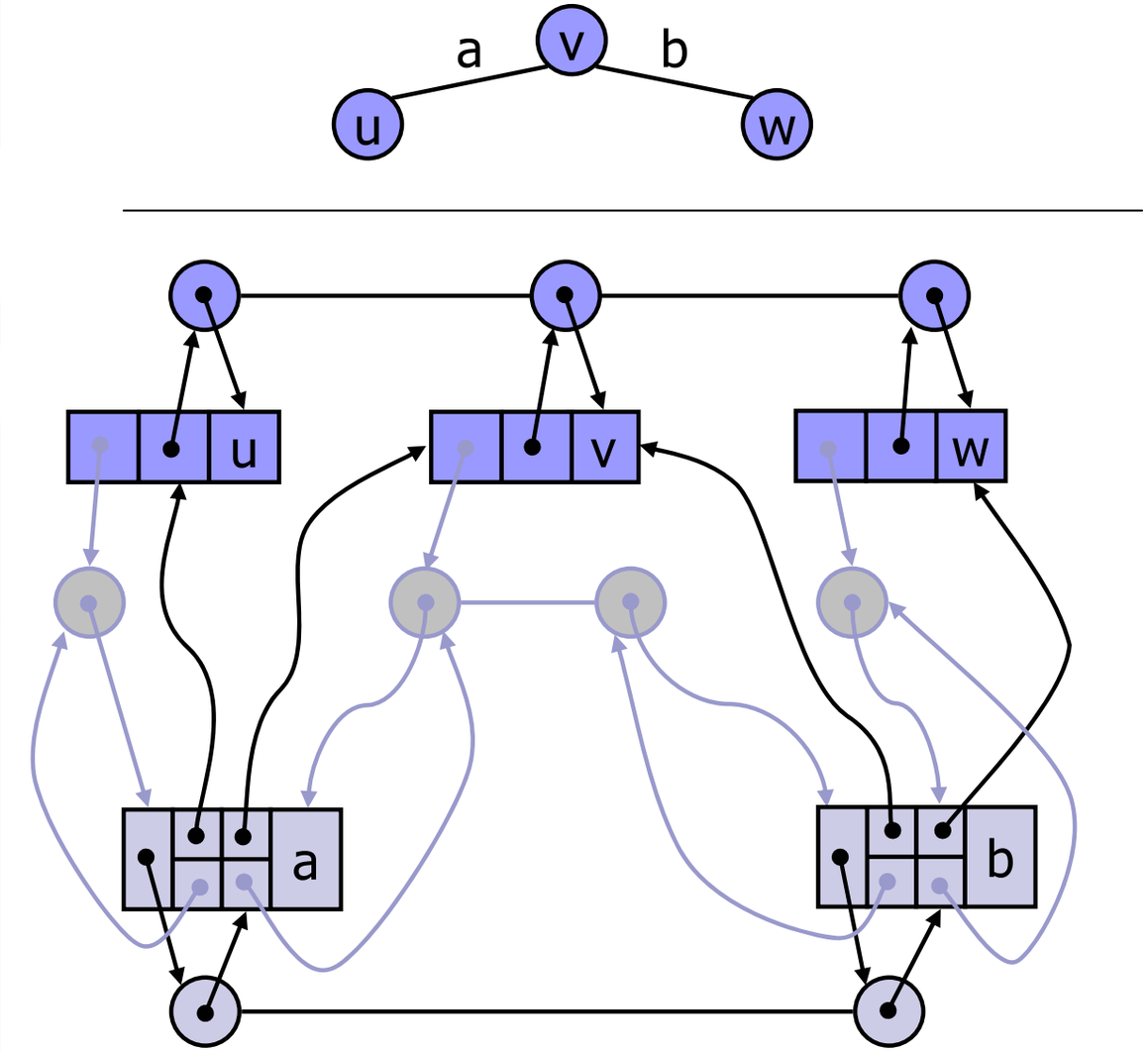
Nabolister

- En liste af kanter
- En liste af knuder
- Knuder



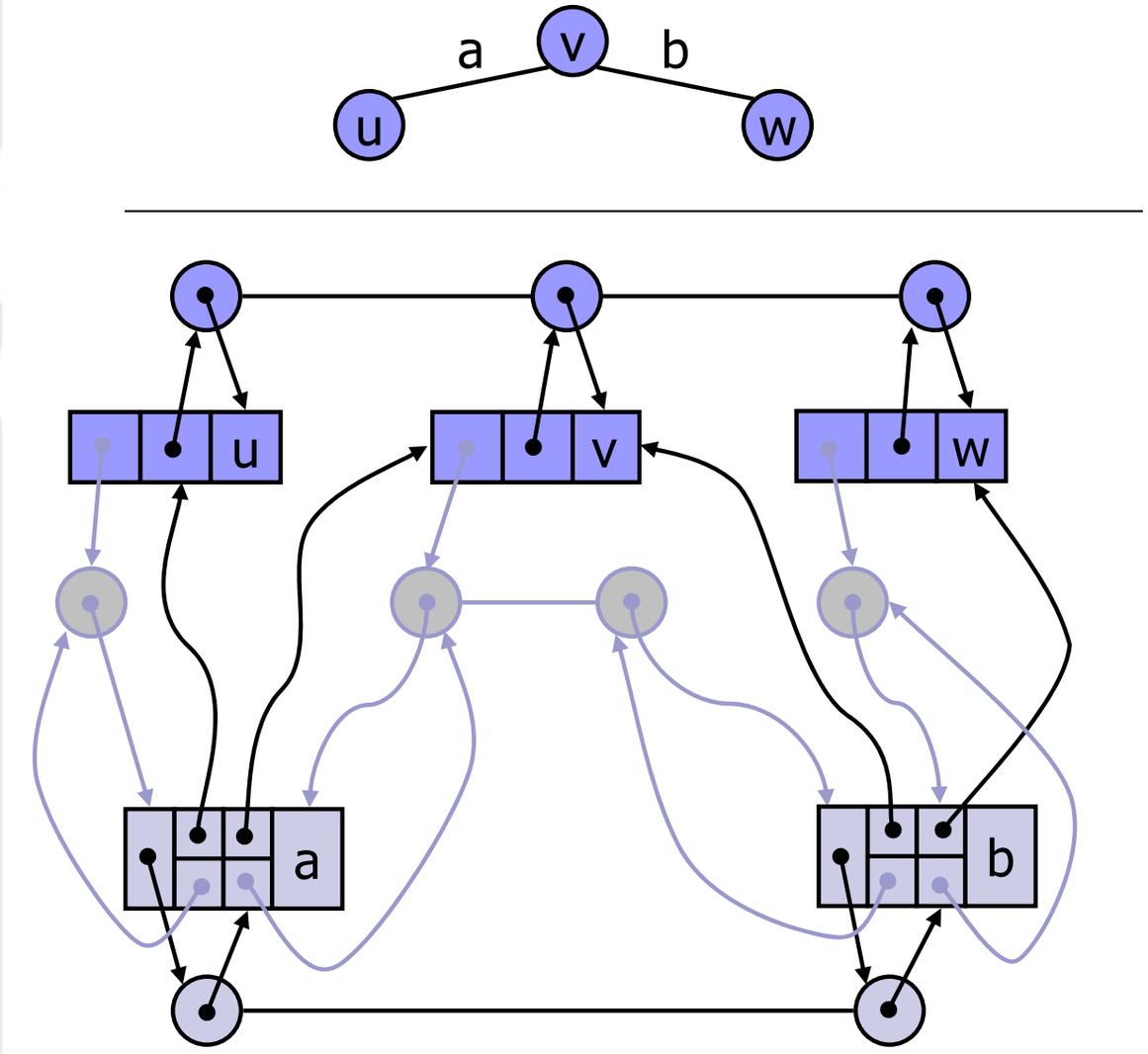
Nabolister

- En liste af kanter
- En liste af knuder
- Knuder
  - Objekt (indhold)



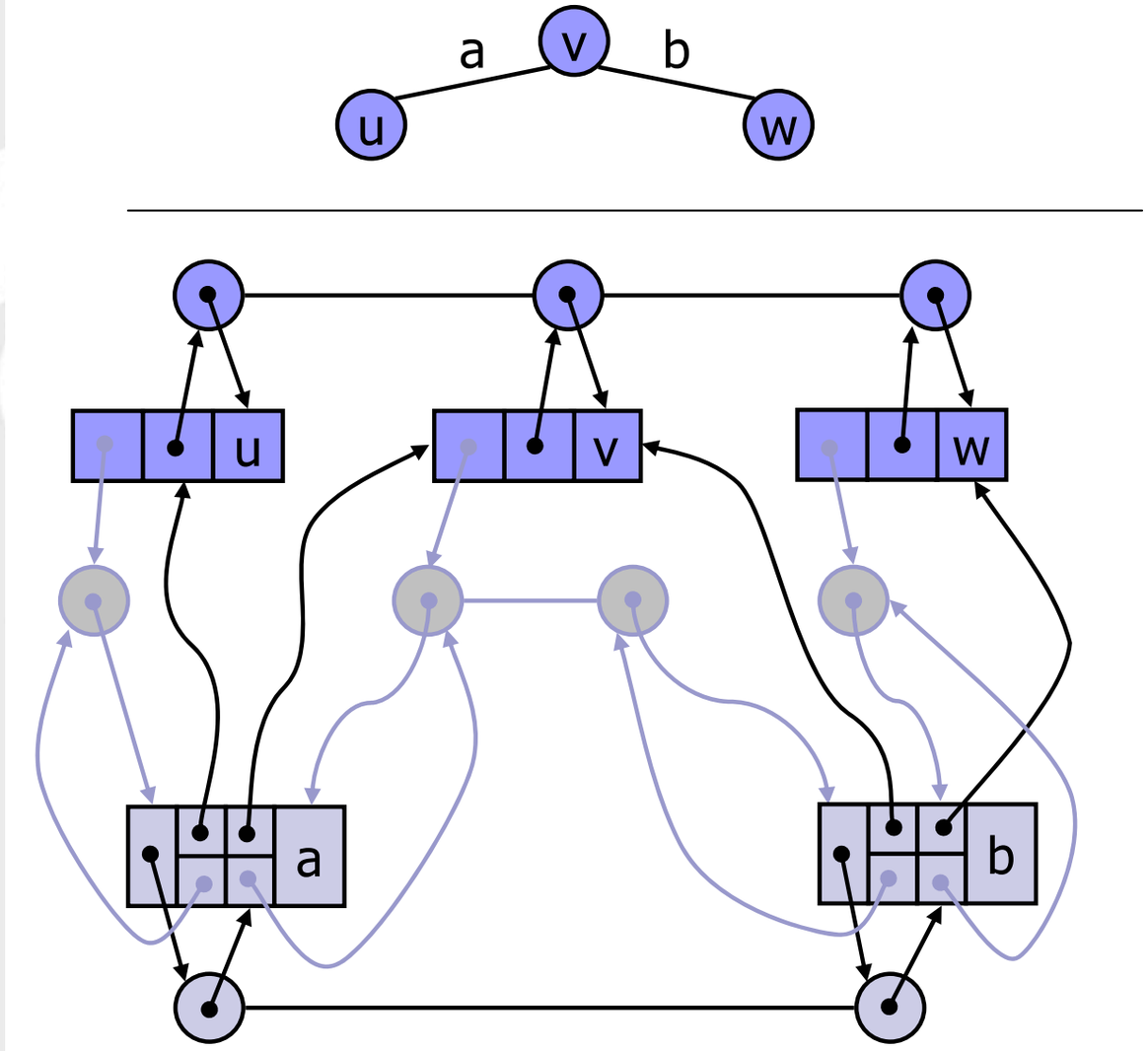
Nabolister

- En liste af kanter
- En liste af knuder
- Knuder
  - Objekt (indhold)
  - Liste af referencer til incidente kanter



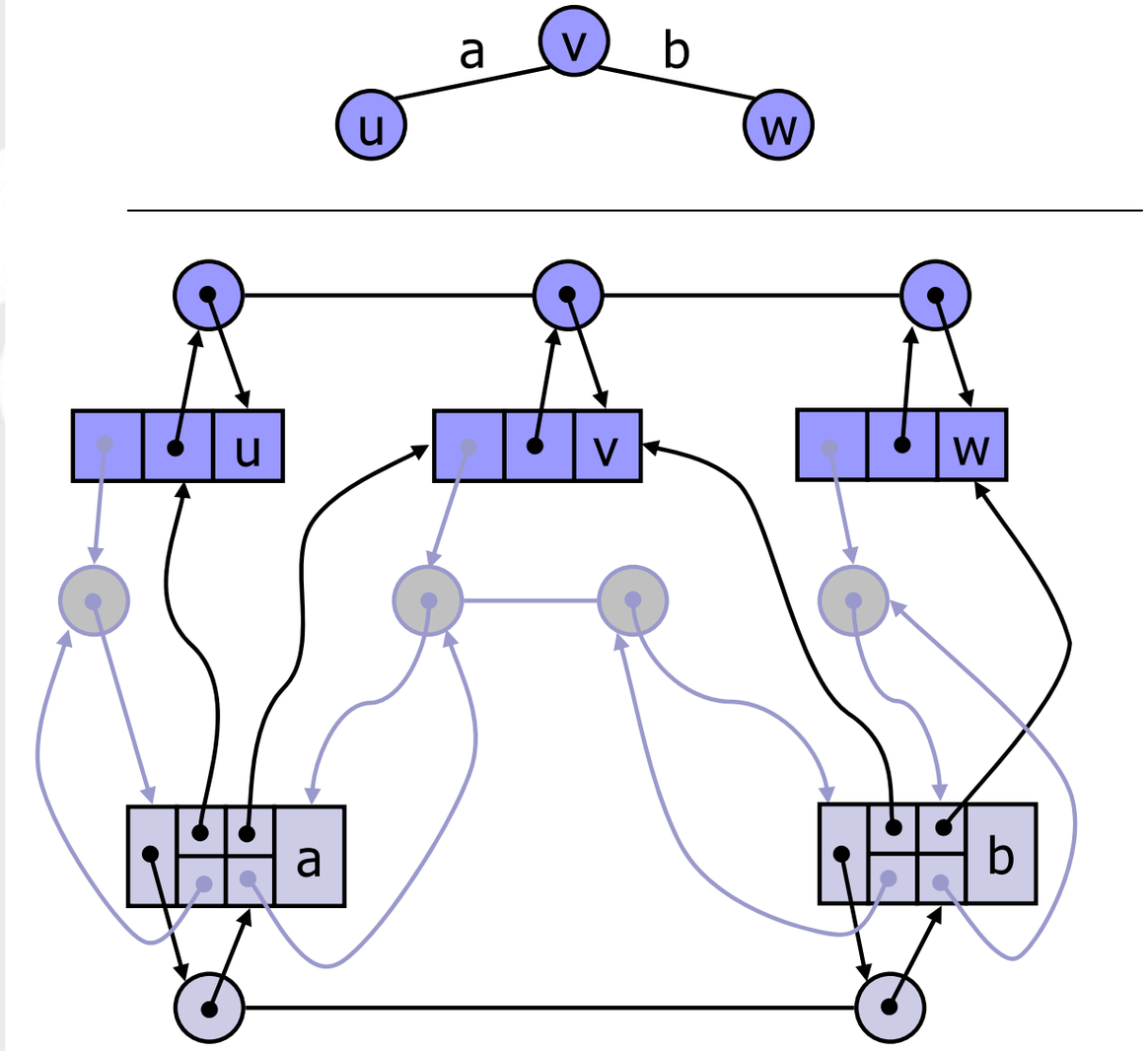
Nabolister

- En liste af kanter
- En liste af knuder
- Knuder
  - Objekt (indhold)
  - Liste af referencer til incidente kanter
- Kanter



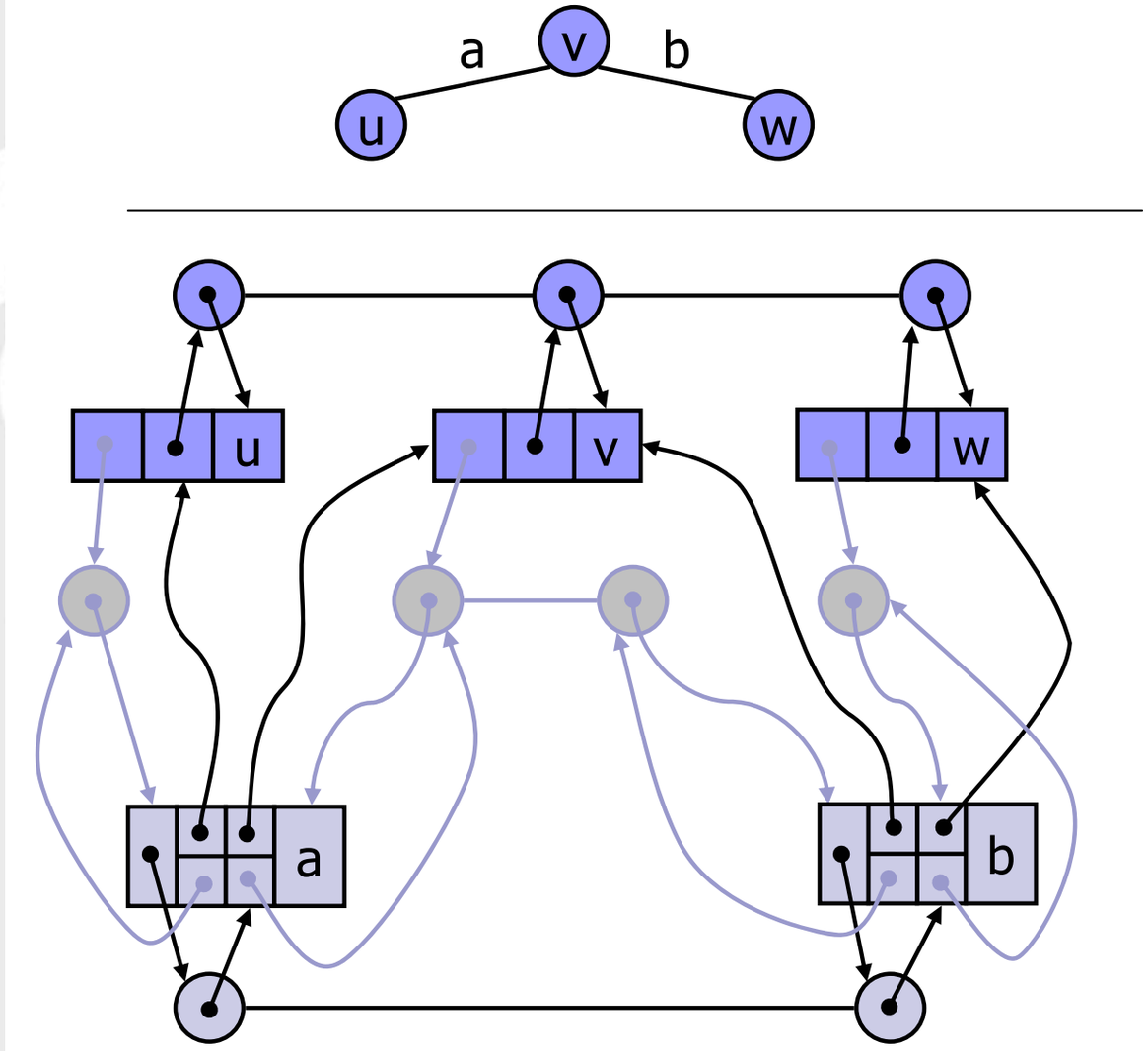
Nabolister

- En liste af kanter
- En liste af knuder
- Knuder
  - Objekt (indhold)
  - Liste af referencer til incidente kanter
- Kanter
  - Objekt (indhold)



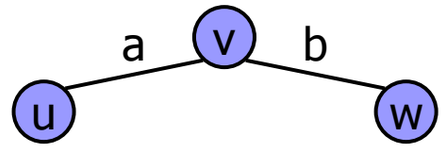
Nabolister

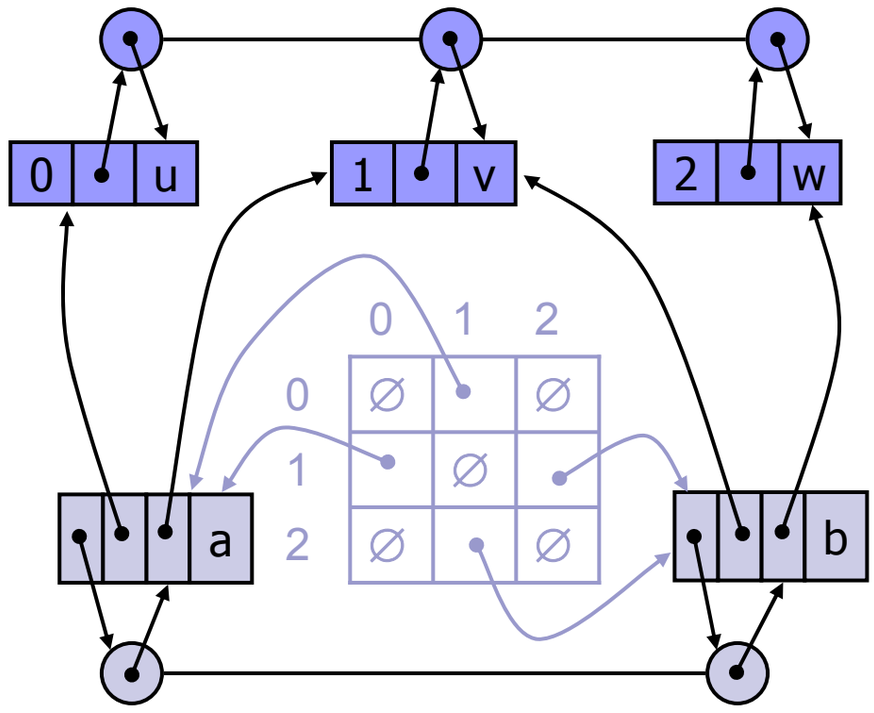
- En liste af kanter
- En liste af knuder
- Knuder
  - Objekt (indhold)
  - Liste af referencer til incidente kanter
- Kanter
  - Objekt (indhold)
  - Liste af referencer til endepunkterne



Incidensmatrix

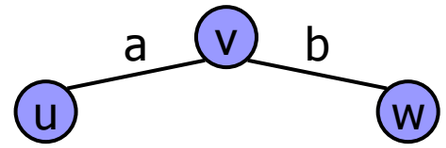
Incidensmatrix

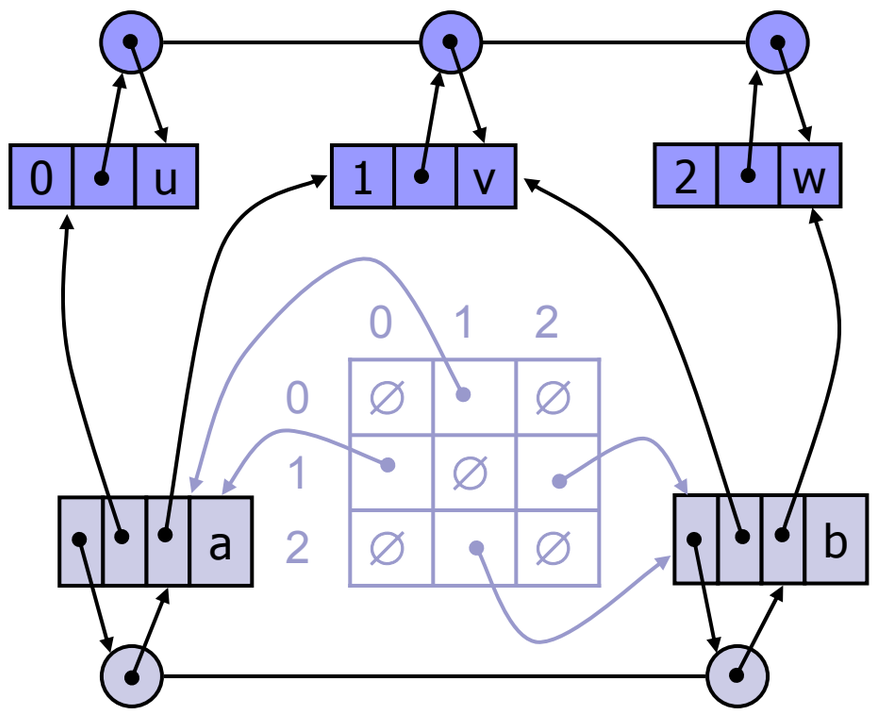




Incidensmatrix

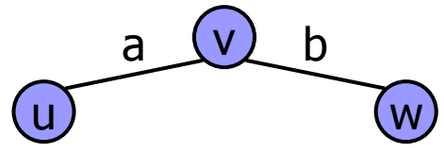
- En liste af kanter

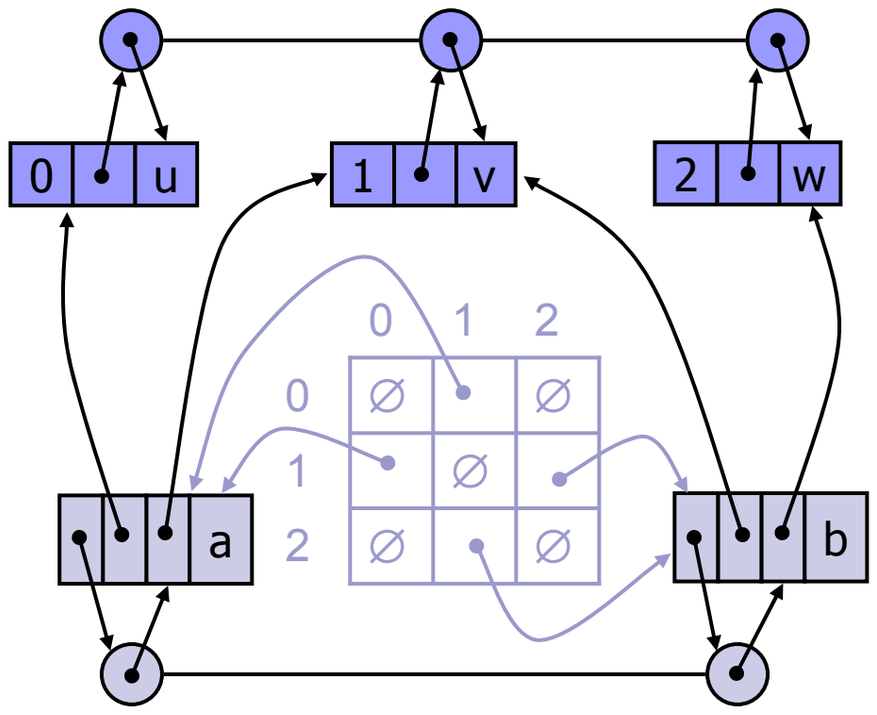




Incidensmatrix

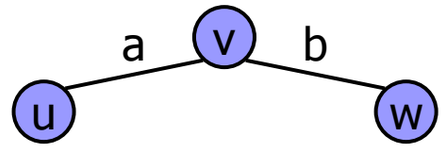
- En liste af kanter
- En liste af knuder

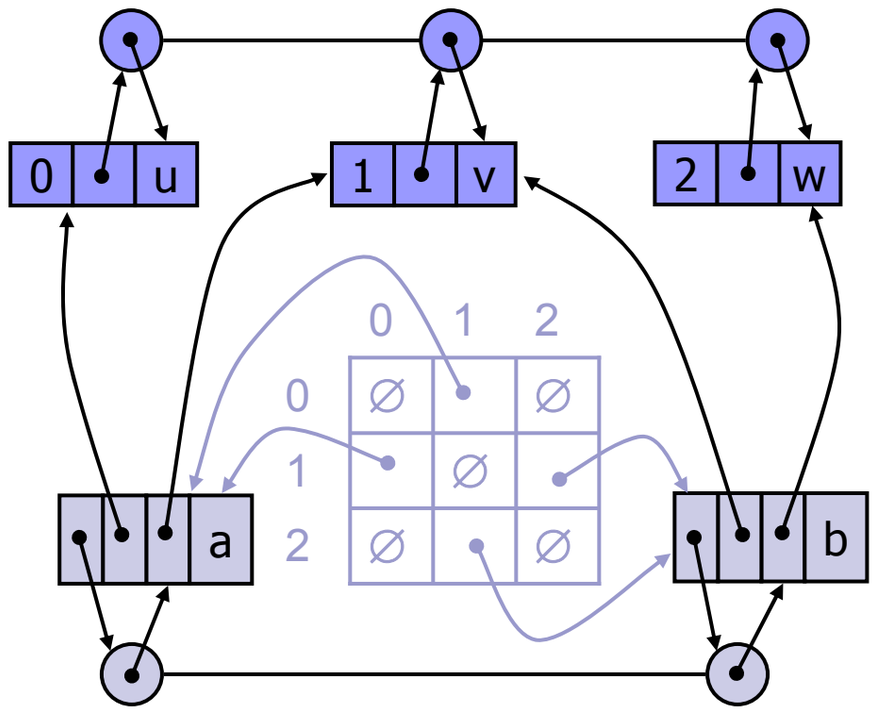




Incidensmatrix

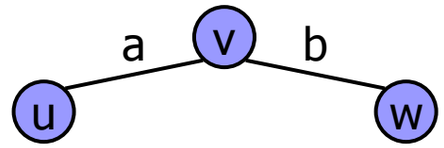
- En liste af kanter
- En liste af knuder
  - Hver knude husker sit index i listen

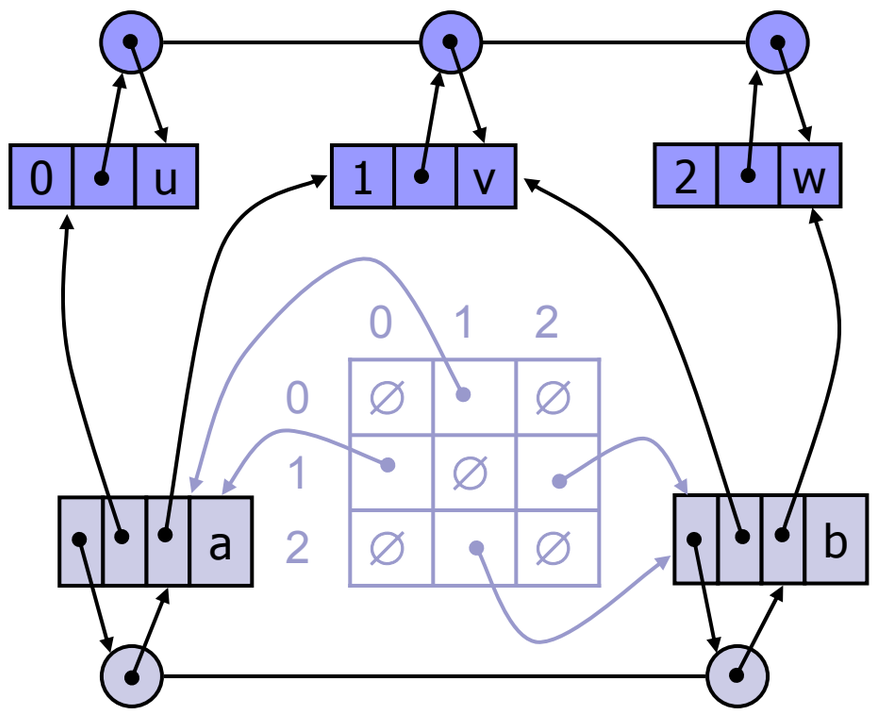




Incidensmatrix

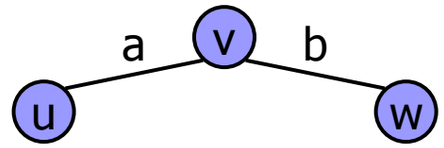
- En liste af kanter
- En liste af knuder
  - Hver knude husker sit index i listen
- En matrice

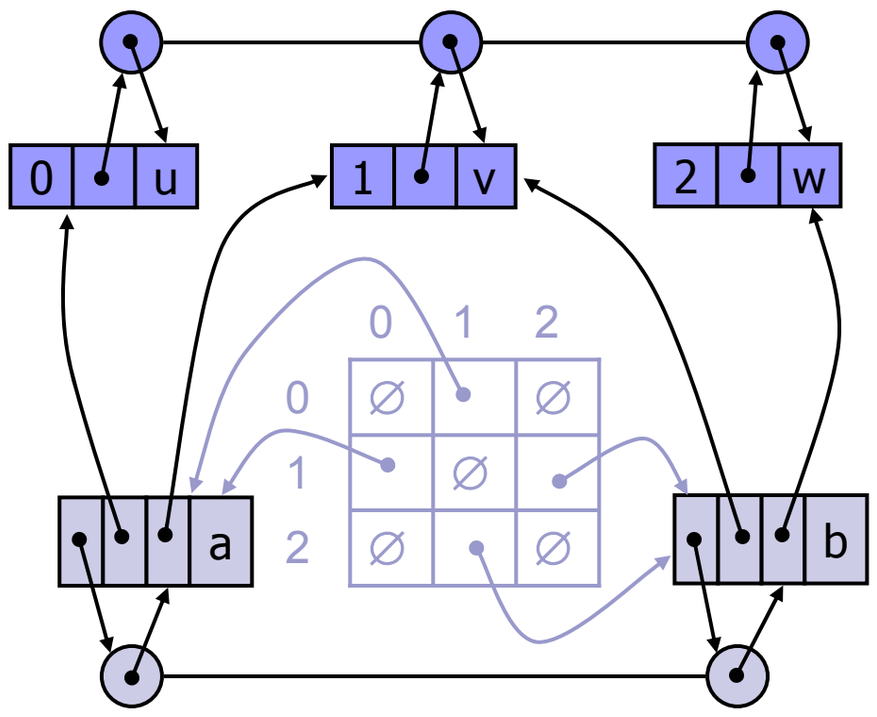




Incidensmatrix

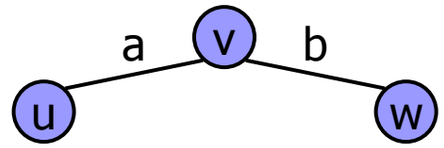
- En liste af kanter
- En liste af knuder
  - Hver knude husker sit index i listen
- En matrice
  - Index a knuderne i hver dimension

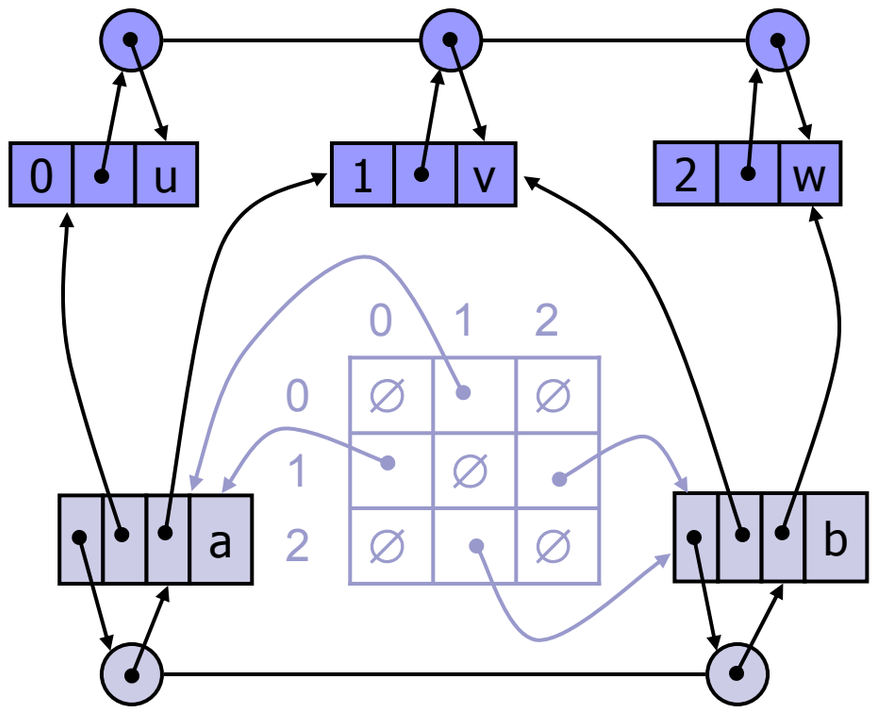




Incidensmatrix

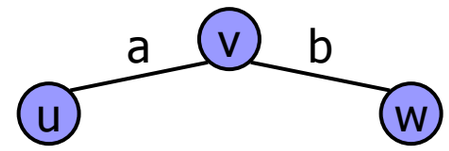
- En liste af kanter
- En liste af knuder
  - Hver knude husker sit index i listen
- En matrice
  - Index a knuderne i hver dimension
  - Reference til kant

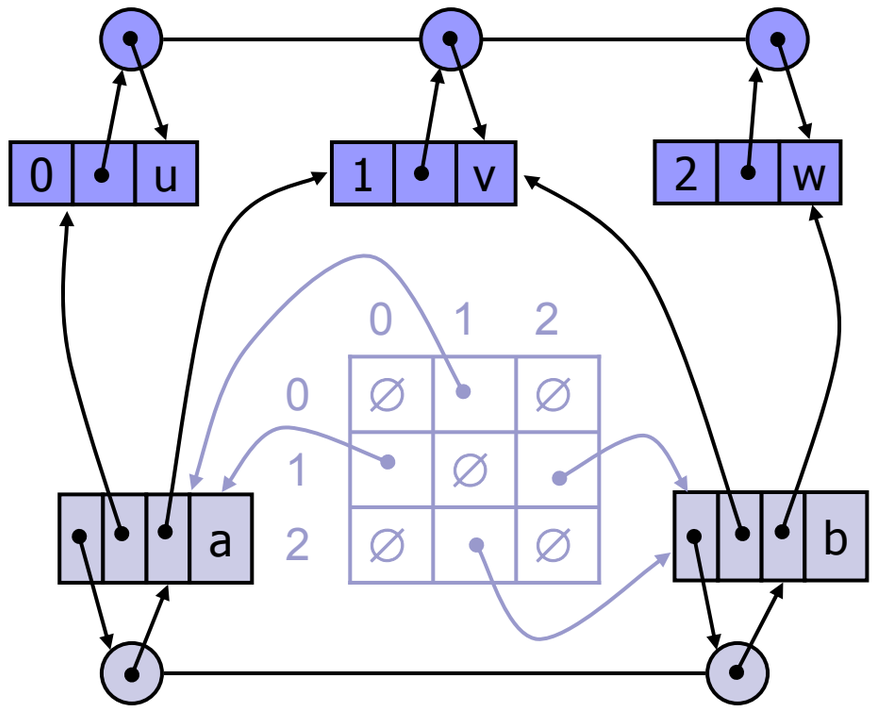




Incidensmatrix

- En liste af kanter
- En liste af knuder
  - Hver knude husker sit index i listen
- En matrice
  - Index a knuderne i hver dimension
  - Reference til kant
  - null ellers





Hastighed

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads		

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)		

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n+m$	$n+m$
incidentEdges(v)	$deg(v)$	

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n+m$	$n+m$
incidentEdges(v)	$deg(v)$	n

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)		

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)		

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)		

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	1

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	1
removeVertex(v)		

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	1
removeVertex(v)	$deg(v)$	

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	1
removeVertex(v)	$deg(v)$	$deg(v)$

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	1
removeVertex(v)	$deg(v)$	$deg(v)$
removeEdge(e)		

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	1
removeVertex(v)	$deg(v)$	$deg(v)$
removeEdge(e)	1	

Hastighed

n knuder m kanter	Naboliste	Incidensmatrix
Plads	$n + m$	$n + m$
incidentEdges(v)	$deg(v)$	n
areAdjacent(v, w)	$\min(deg(v), deg(w))$	1
insertVertex(o)	1	1
insertEdge(v, w, o)	1	1
removeVertex(v)	$deg(v)$	$deg(v)$
removeEdge(e)	1	1

Dybde-først-søgning

Dybde-først-søgning

- Definitioner
  - Delgraf
  - Sammenhængende graf
  - Træer og skove
  - Udspændende træer og skove

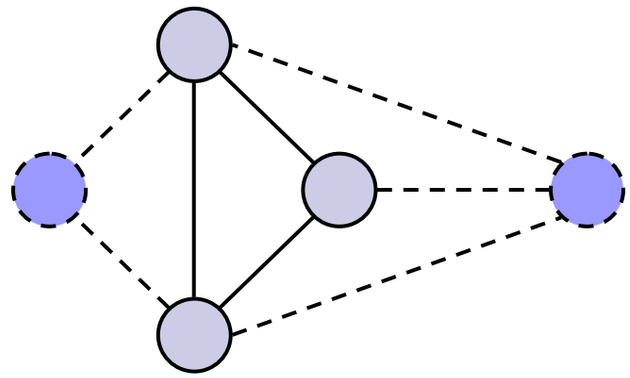
Dybde-først-søgning

- Definitioner
  - Delgraf
  - Sammenhængende graf
  - Træer og skove
  - Udspændende træer og skove
- DFS
  - Algoritmen
  - Eksempel
  - Egenskaber
  - Java-kode

Delgraf

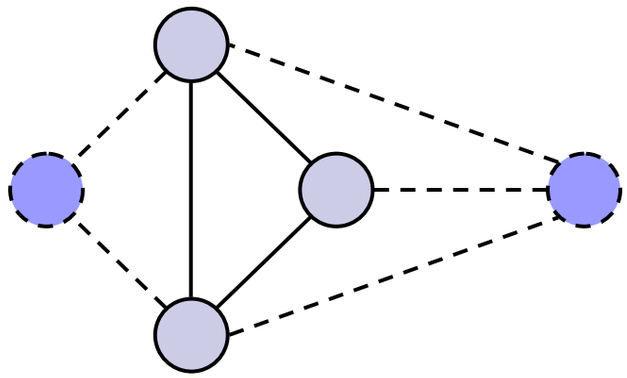
Delgraf

- En delgraf S af en graf G er en graf så:



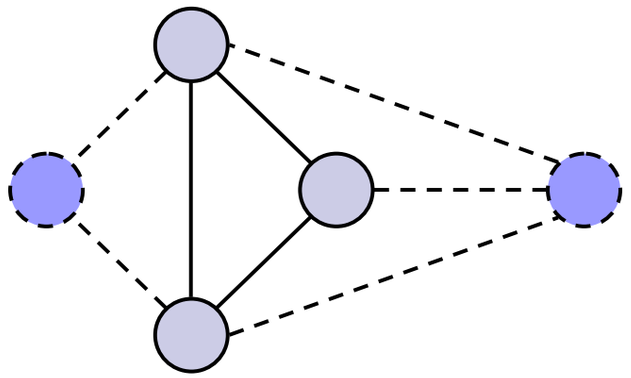
Delgraf

- En delgraf S af en graf G er en graf så:
  - Knuderne i S er en delmængde af knuderne i G



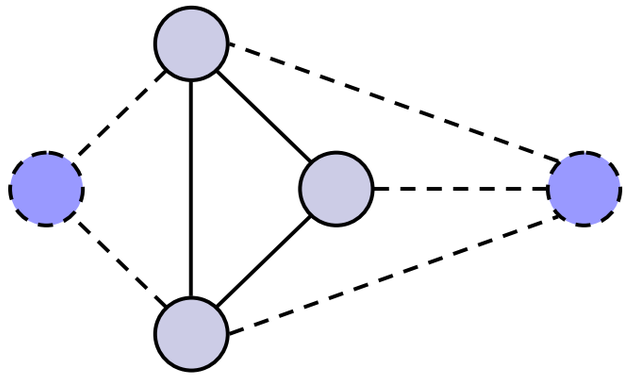
Delgraf

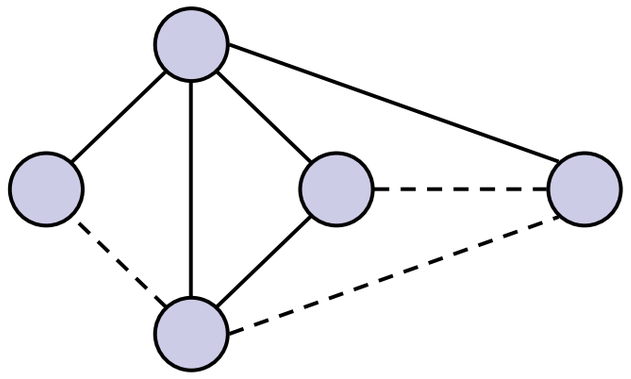
- En delgraf S af en graf G er en graf så:
  - Knuderne i S er en delmængde af knuderne i G
  - Kanterne i S er en delmængde af kanterne i G



Delgraf

- En delgraf S af en graf G er en graf så:
  - Knuderne i S er en delmængde af knuderne i G
  - Kanterne i S er en delmængde af kanterne i G
- En udspændende delgraf af G er en delgraf som indeholder alle knuderne i G

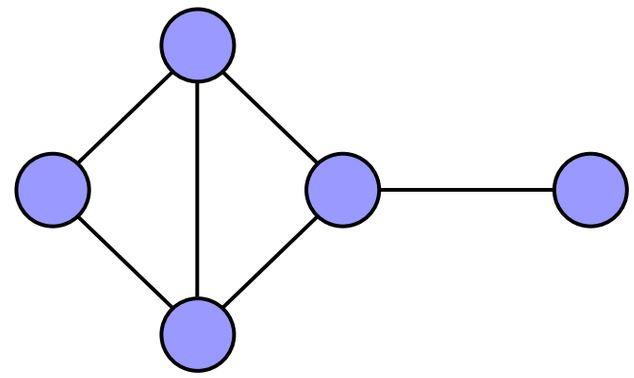




Sammenhængende graf

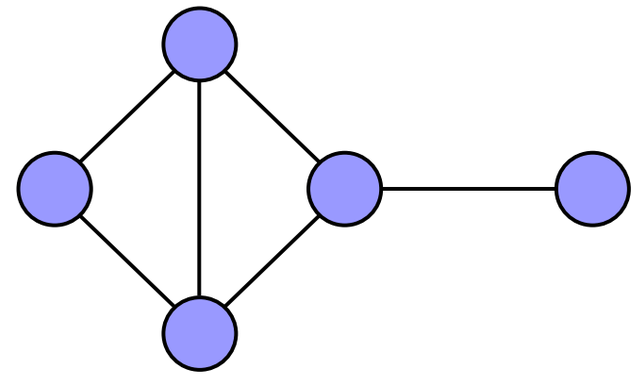
Sammenhængende graf

- En graf er sammenhængende hvis der er en sti mellem alle par af knuder

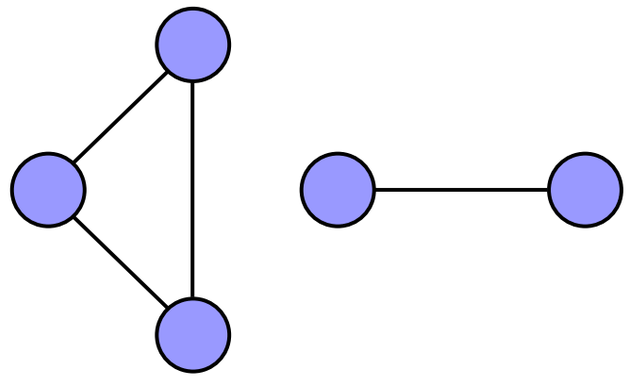


Sammenhængende graf

- En graf er sammenhængende hvis der er en sti mellem alle par af knuder



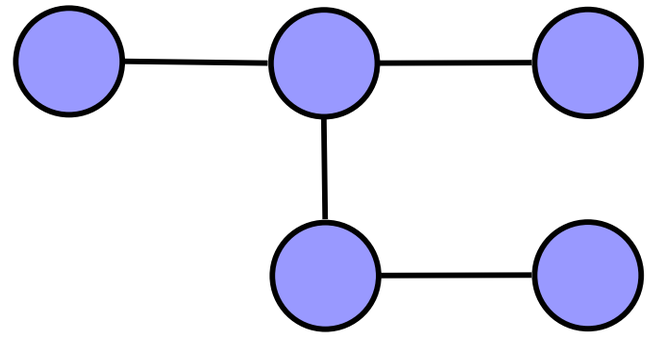
- En sammenhængskomponent af en graf G er en maksimal sammenhængende delgraf af G



Træer og skove

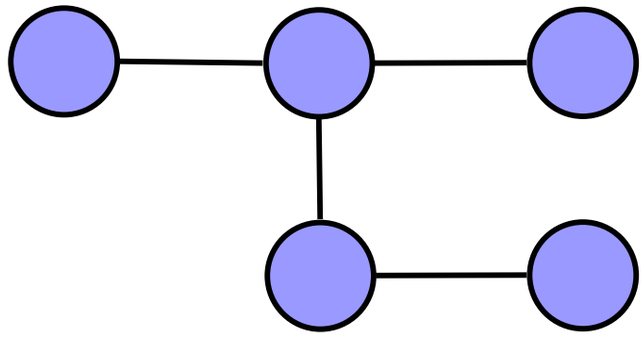
Træer og skove

- Et træ er en ikke-orienteret graf T der opfylder:



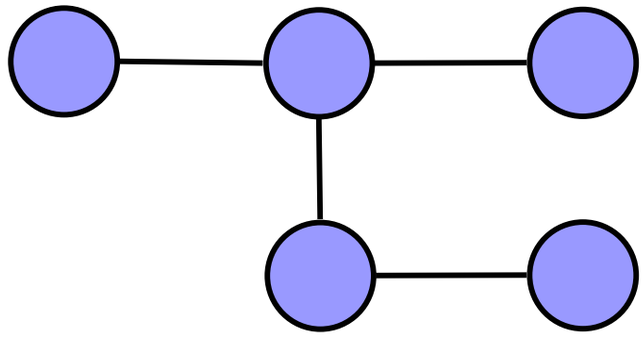
Træer og skove

- Et træ er en ikke-orienteret graf T der opfylder:
  - T er sammenhængende



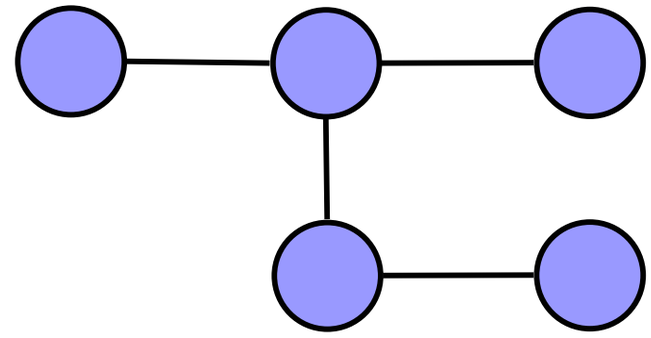
Træer og skove

- Et træ er en ikke-orienteret graf T der opfylder:
  - T er sammenhængende
  - T har ikke kredse



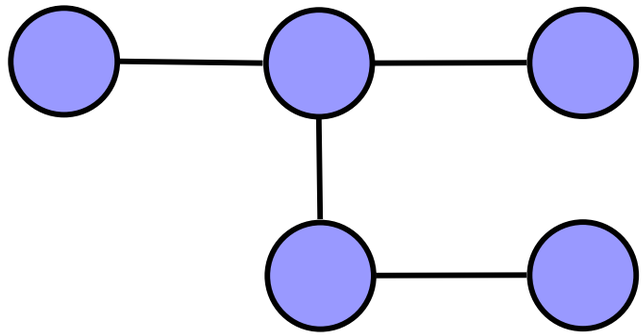
Træer og skove

- Et træ er en ikke-orienteret graf T der opfylder:
  - T er sammenhængende
  - T har ikke kredse
- Bemærk at dette er forskelligt fra definitionen af et rodet træet (som vi tidligere har kigget på)



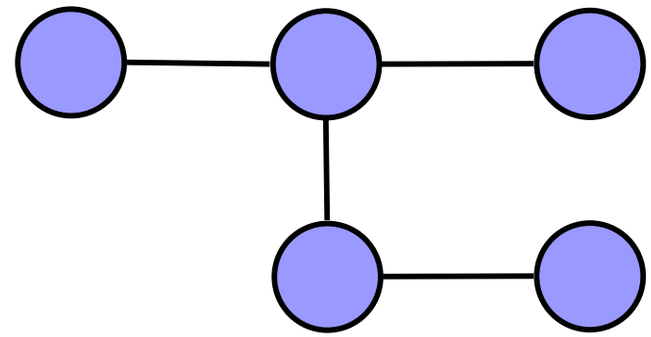
Træer og skove

- Et træ er en ikke-orienteret graf T der opfylder:
  - T er sammenhængende
  - T har ikke kredse
- Bemærk at dette er forskelligt fra definitionen af et rodet træet (som vi tidligere har kigget på)
- En skov er en ikke-orienteret graf uden kredse

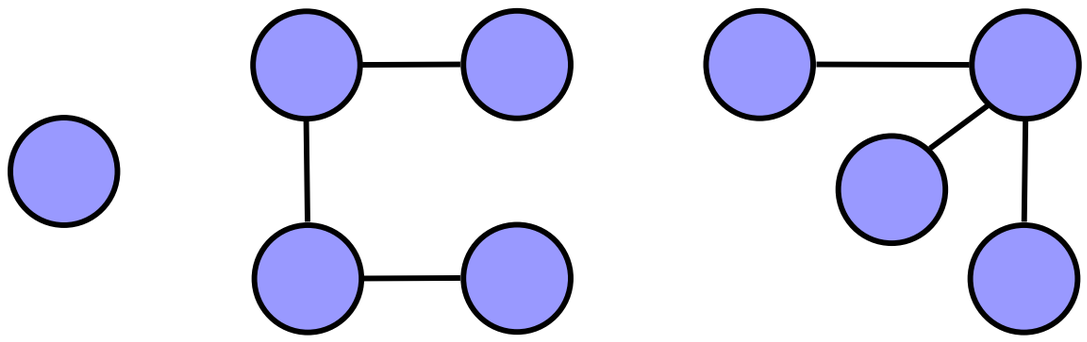


Træer og skove

- Et træ er en ikke-orienteret graf T der opfylder:
  - T er sammenhængende
  - T har ikke kredse
- Bemærk at dette er forskelligt fra definitionen af et rodet træet (som vi tidligere har kigget på)



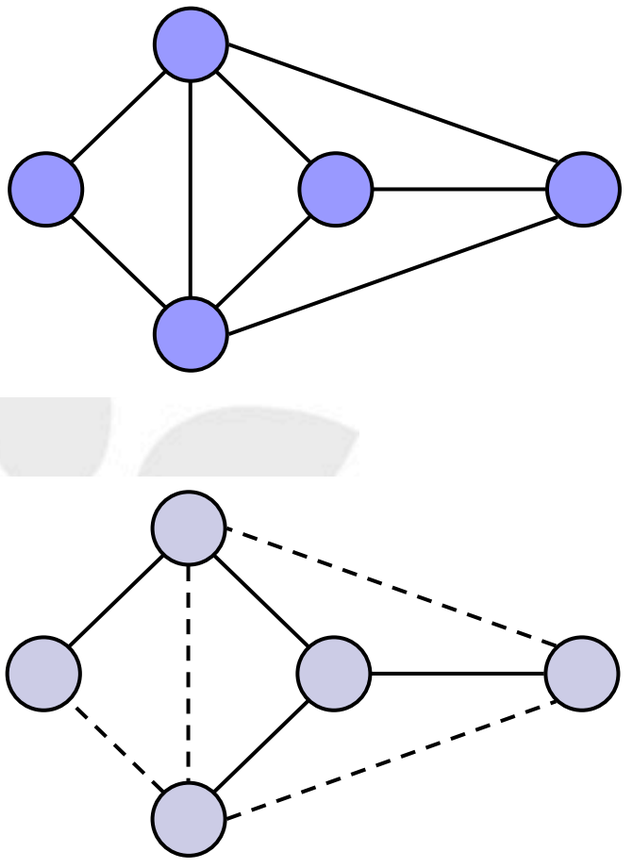
- En skov er en ikke-orienteret graf uden kredse
- Sammenhængskomponenterne af en skov er træer



Udspændende træer og skove

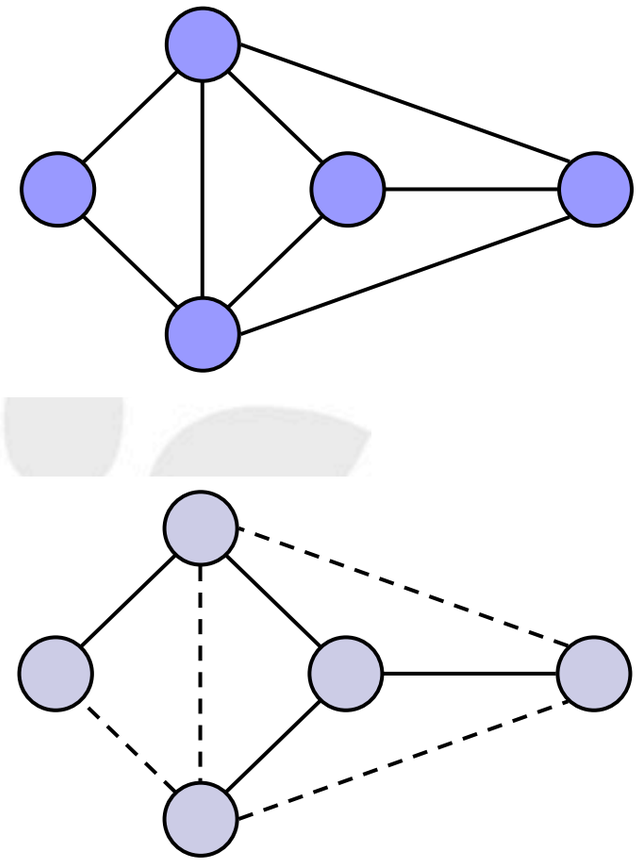
Udspændende træer og skove

- Et udspændende træ for en sammenhængende graf er en udspændende delgraf der er træ



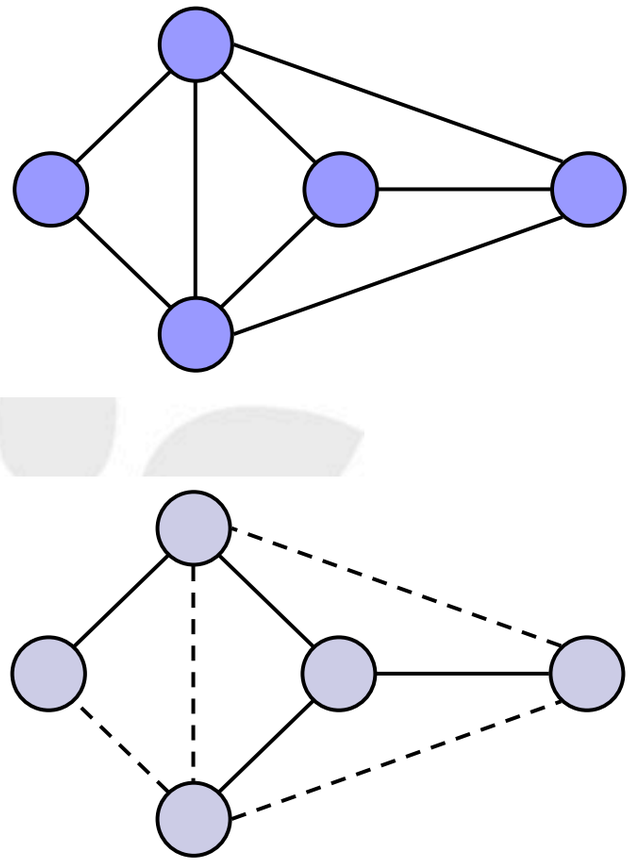
Udspændende træer og skove

- Et udspændende træ for en sammenhængende graf er en udspændende delgraf der er træ
  - Et udspændende træ er ikke unikt (pånær hvis grafen er et træ)



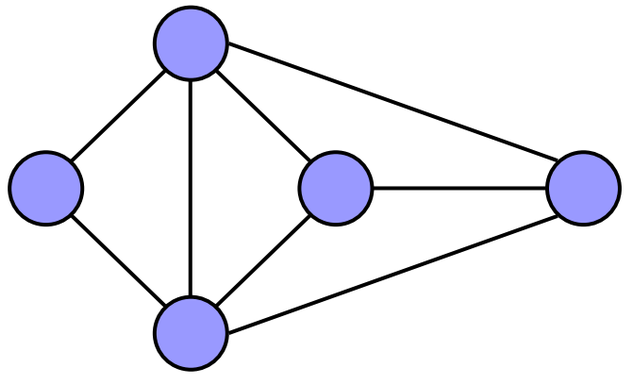
Udspændende træer og skove

- Et udspændende træ for en sammenhængende graf er en udspændende delgraf der er træ
  - Et udspændende træ er ikke unikt (pånær hvis grafen er et træ)
  - Et udspændende træ har anvendelser i kommunikationsnetværk

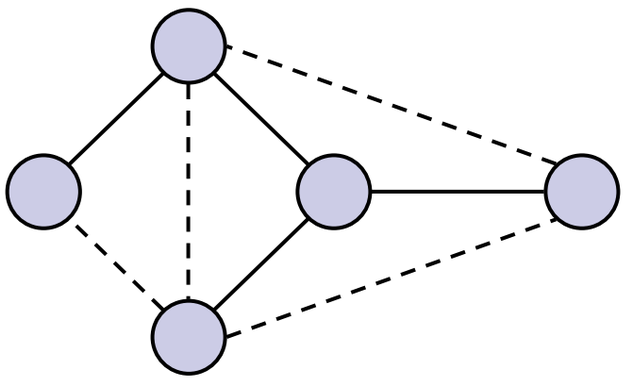


Udspændende træer og skove

- Et udspændende træ for en sammenhængende graf er en udspændende delgraf der er træ
  - Et udspændende træ er ikke unikt (pånær hvis grafen er et træ)
  - Et udspændende træ har anvendelser i kommunikationsnetværk



- En udspændende skov for en graf er en udspændende delgraf der er en skov



Dybde-først-søgning

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- DFS tager tid der er proportionalt med n+m

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- DFS tager tid der er proportionalt med n+m
- DFS kan udvides til at

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- DFS tager tid der er proportionalt med n+m
- DFS kan udvides til at
  - Beregne en sti mellem to givne knuder (hvis en sådan findes)

Dybde-først-søgning

- Dybde-først-søgning (DFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- DFS tager tid der er proportionalt med n+m
- DFS kan udvides til at
  - Beregne en sti mellem to givne knuder (hvis en sådan findes)
  - Finde en kreds i grafen (hvis der er en)

DFS-algoritmen

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"
  - For alle ubesøgte kanter e incident til v

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"
  - For alle ubesøgte kanter e incident til v
    - w = "det andet endepunkt af e"

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"
  - For alle ubesøgte kanter e incident til v
    - w = "det andet endepunkt af e"
    - Hvis w ikke er besøgt

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"
  - For alle ubesøgte kanter e incident til v
    - w = "det andet endepunkt af e"
    - Hvis w ikke er besøgt
      - Giv e mærket "besøgt"

DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"
  - For alle ubesøgte kanter e incident til v
    - w = "det andet endepunkt af e"
    - Hvis w ikke er besøgt
      - Giv e mærket "besøgt"
      - DFS(G, w)

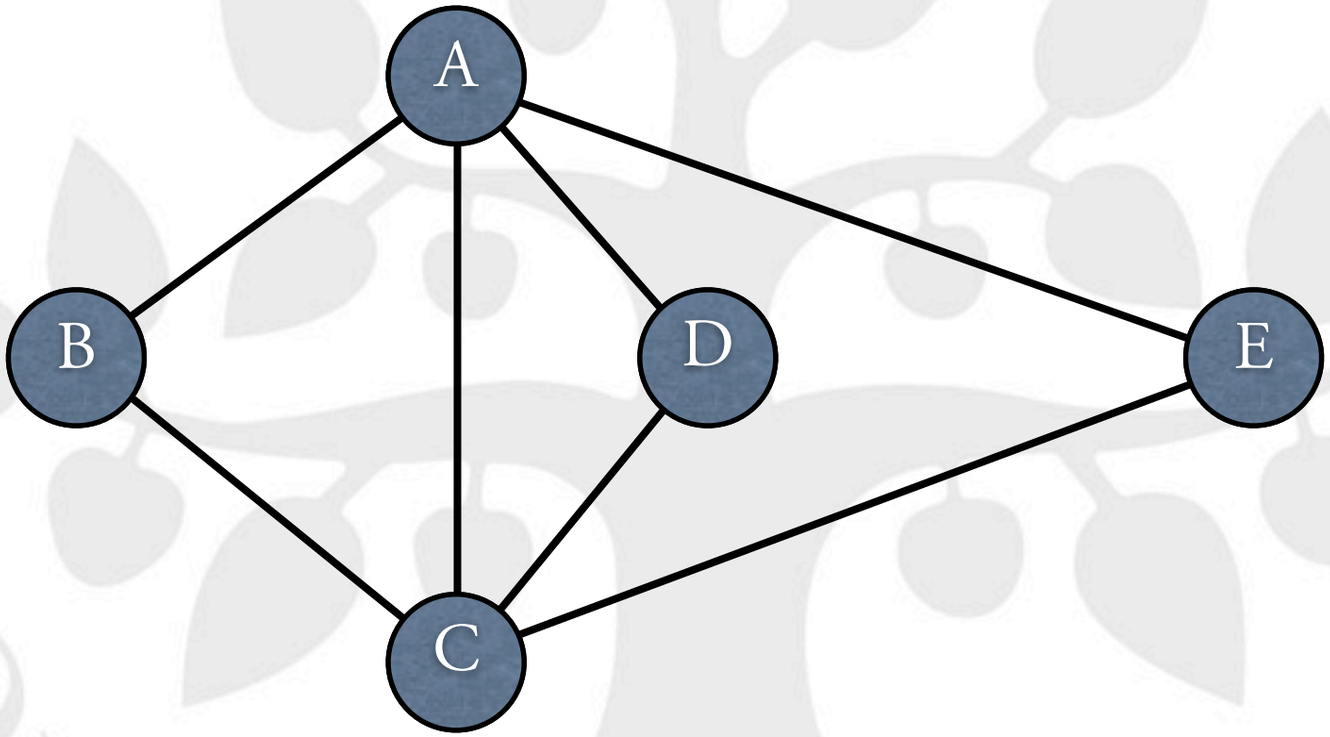
DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"
  - For alle ubesøgte kanter e incident til v
    - w = "det andet endepunkt af e"
    - Hvis w ikke er besøgt
      - Giv e mærket "besøgt"
      - DFS(G, w)
    - Ellers

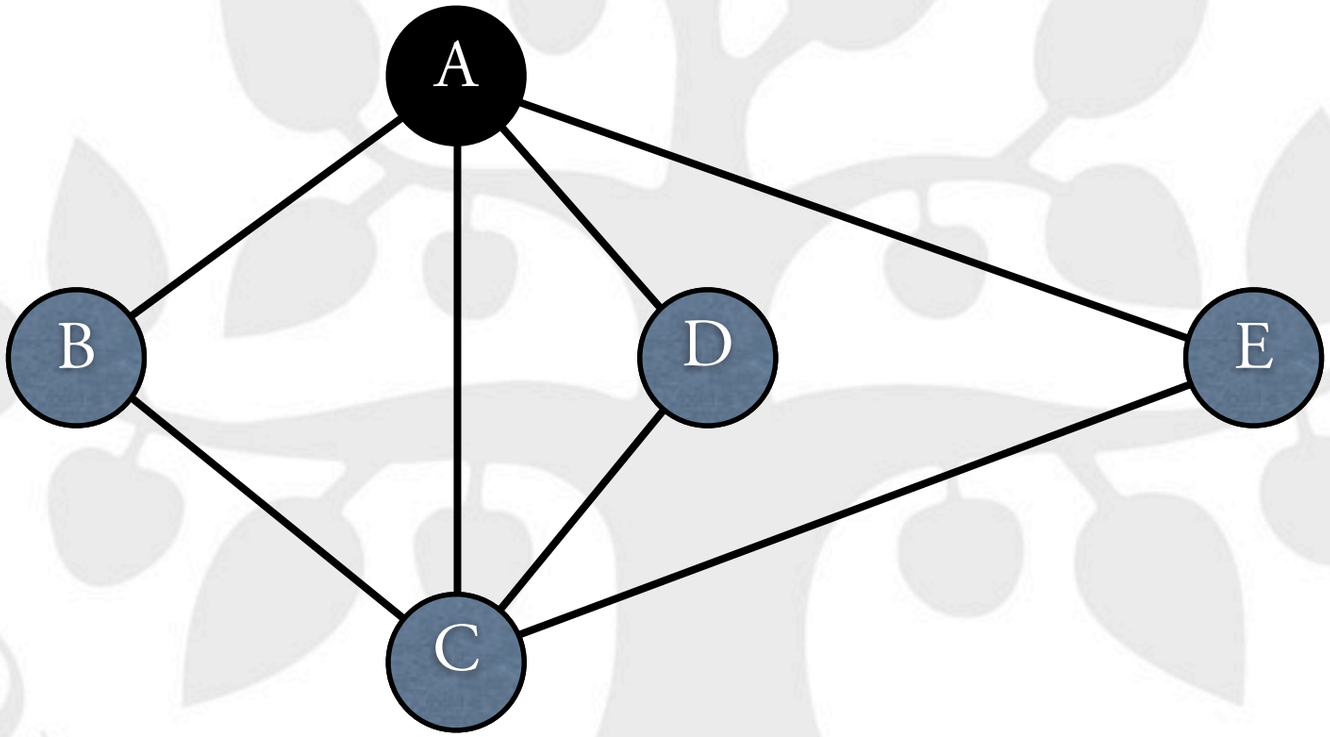
DFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - DFS(G, v)
- DFS(G, v):
  - Giv v mærket "besøgt"
  - For alle ubesøgte kanter e incident til v
    - w = "det andet endepunkt af e"
    - Hvis w ikke er besøgt
      - Giv e mærket "besøgt"
      - DFS(G, w)
    - Ellers
      - Giv e mærket "tidligere besøgt knude"

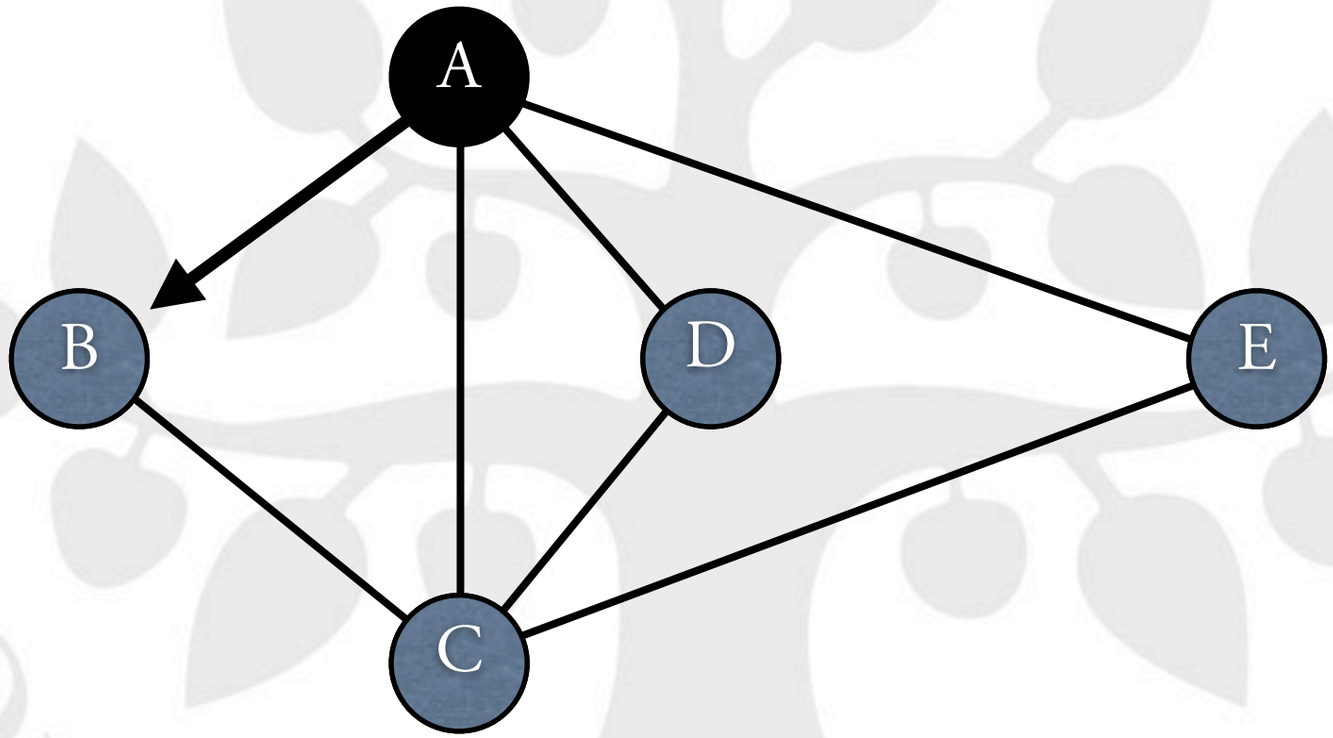
Eksempel



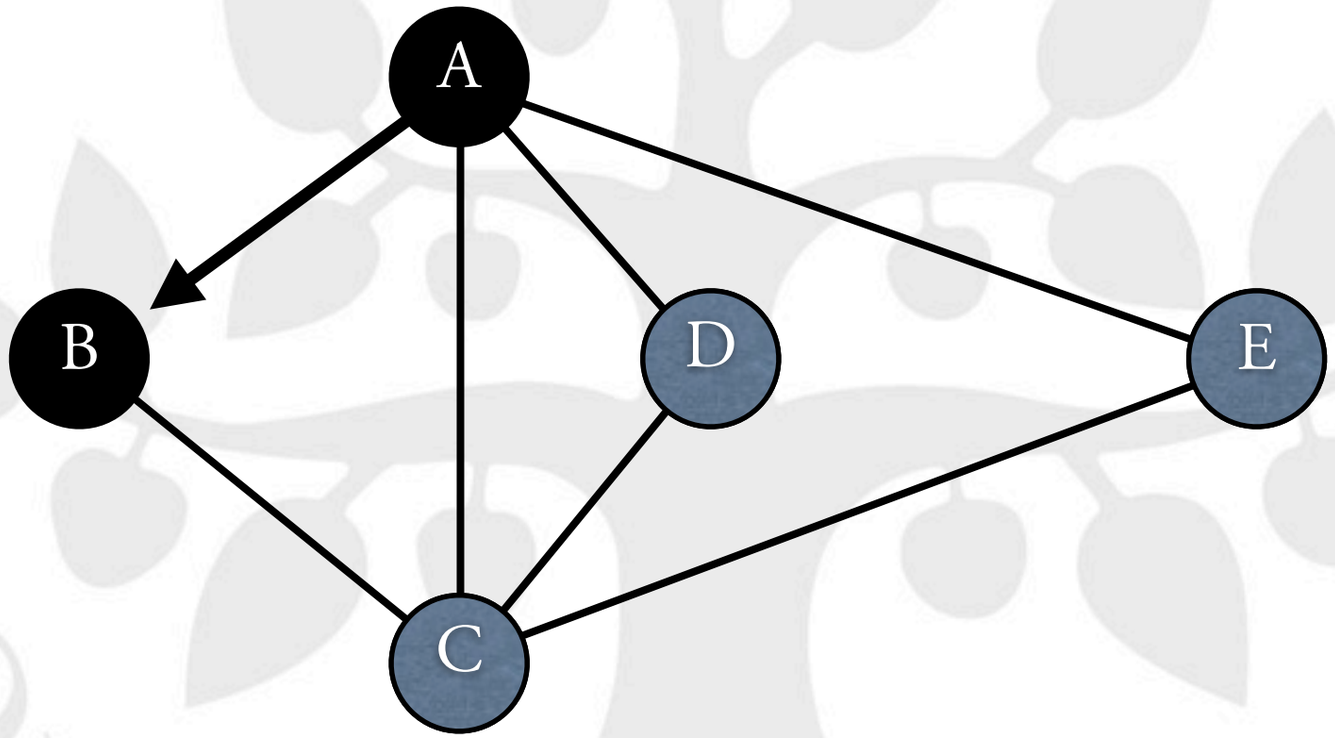
Eksempel



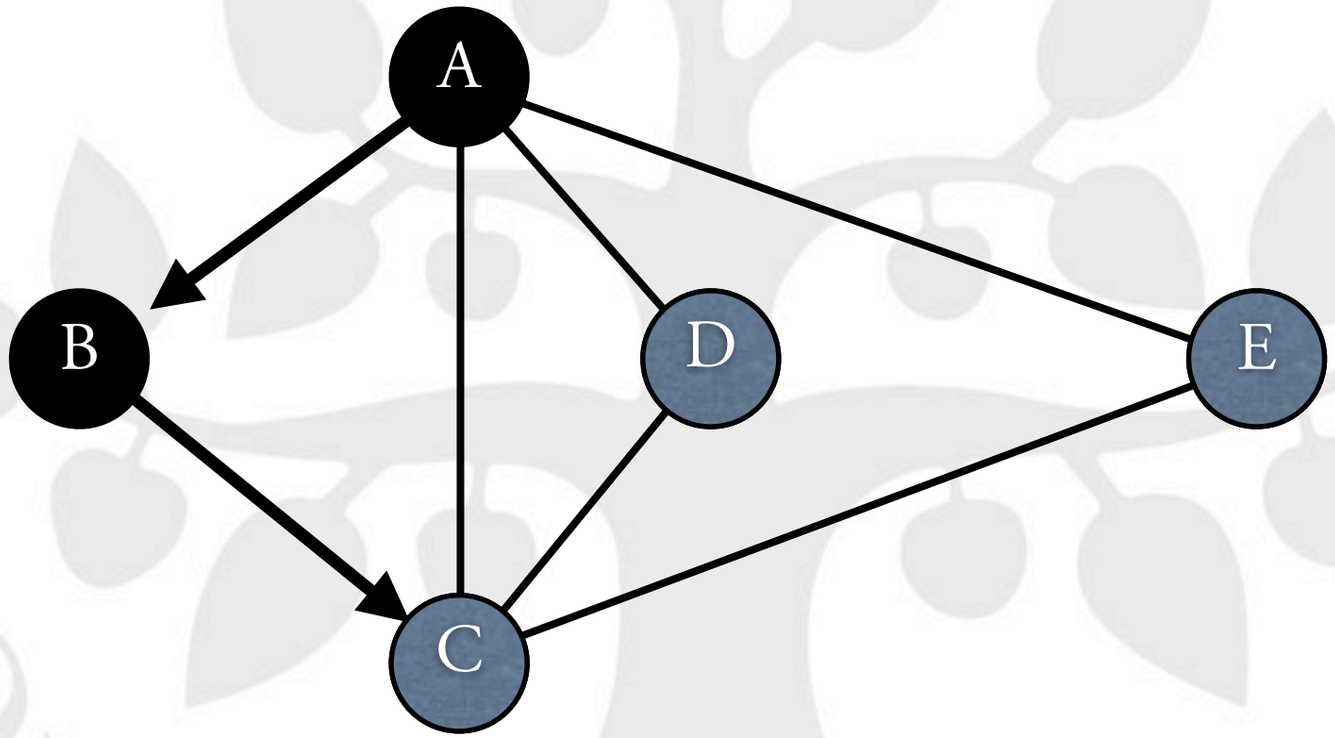
Eksempel



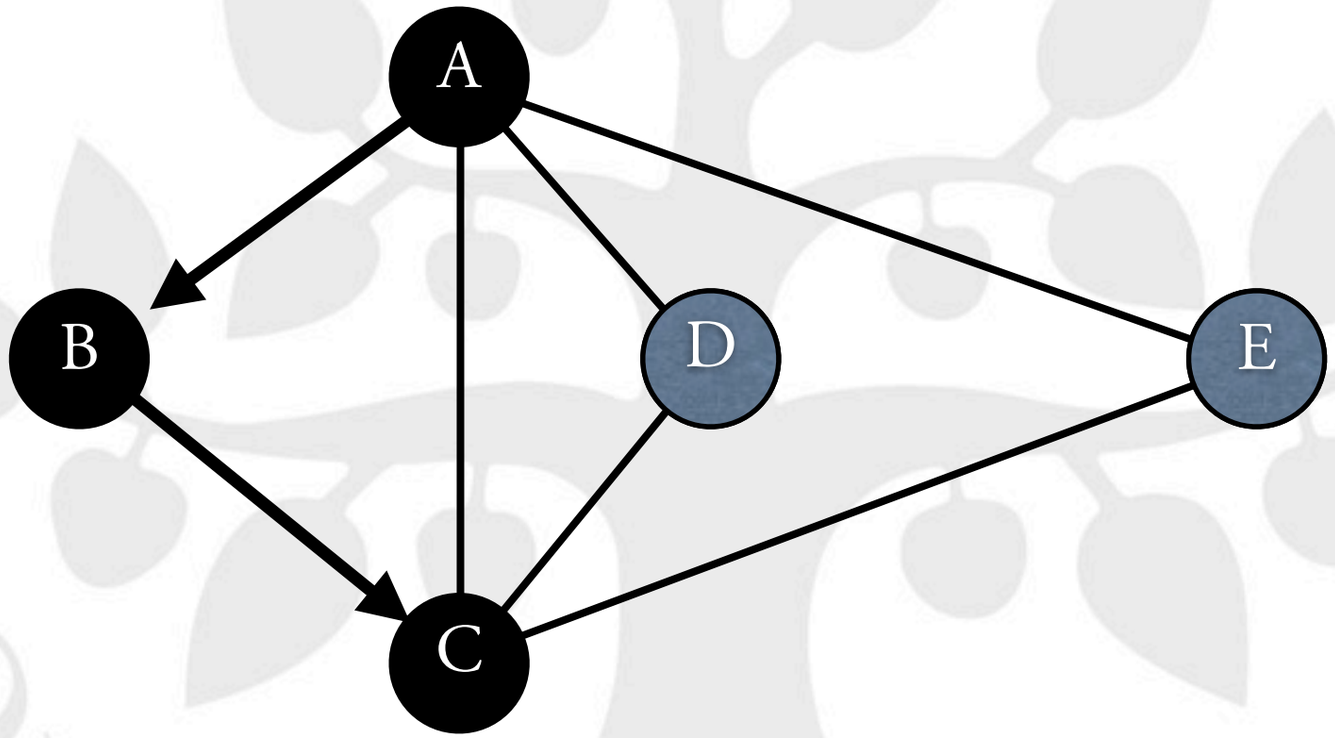
Eksempel



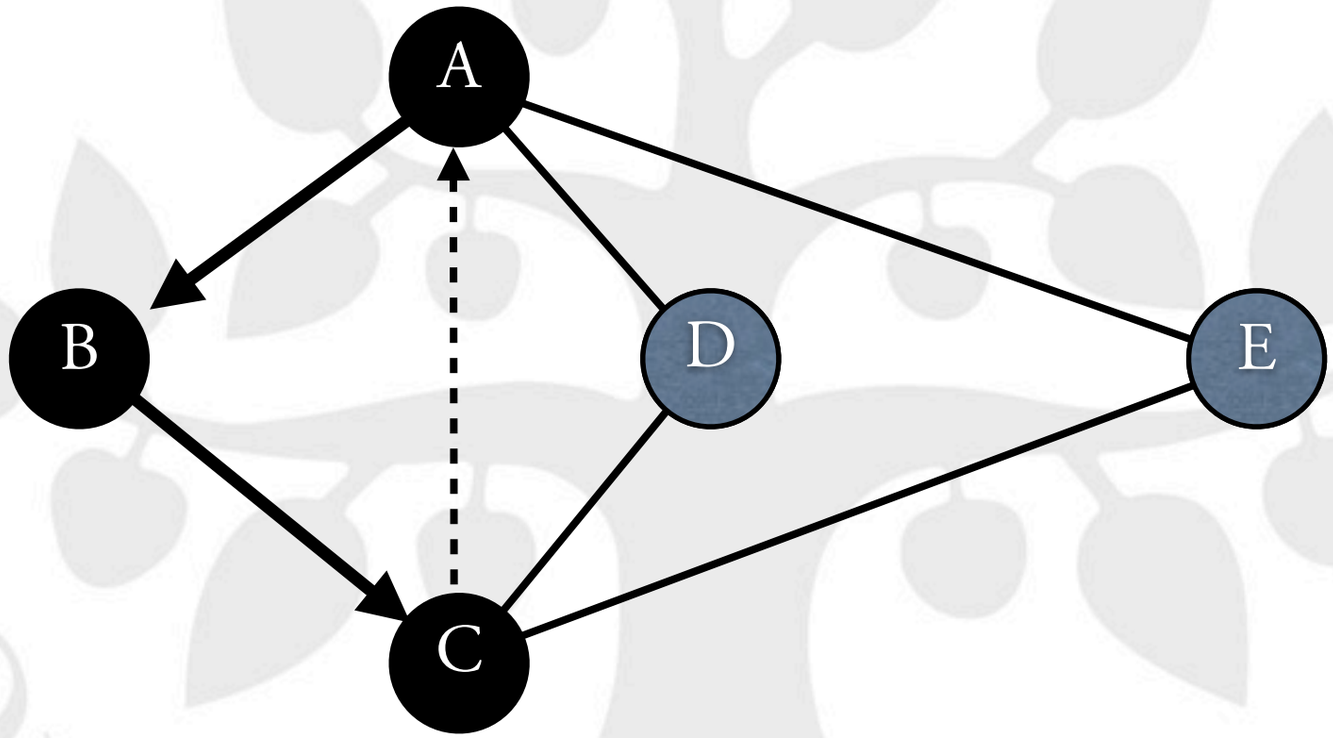
Eksempel



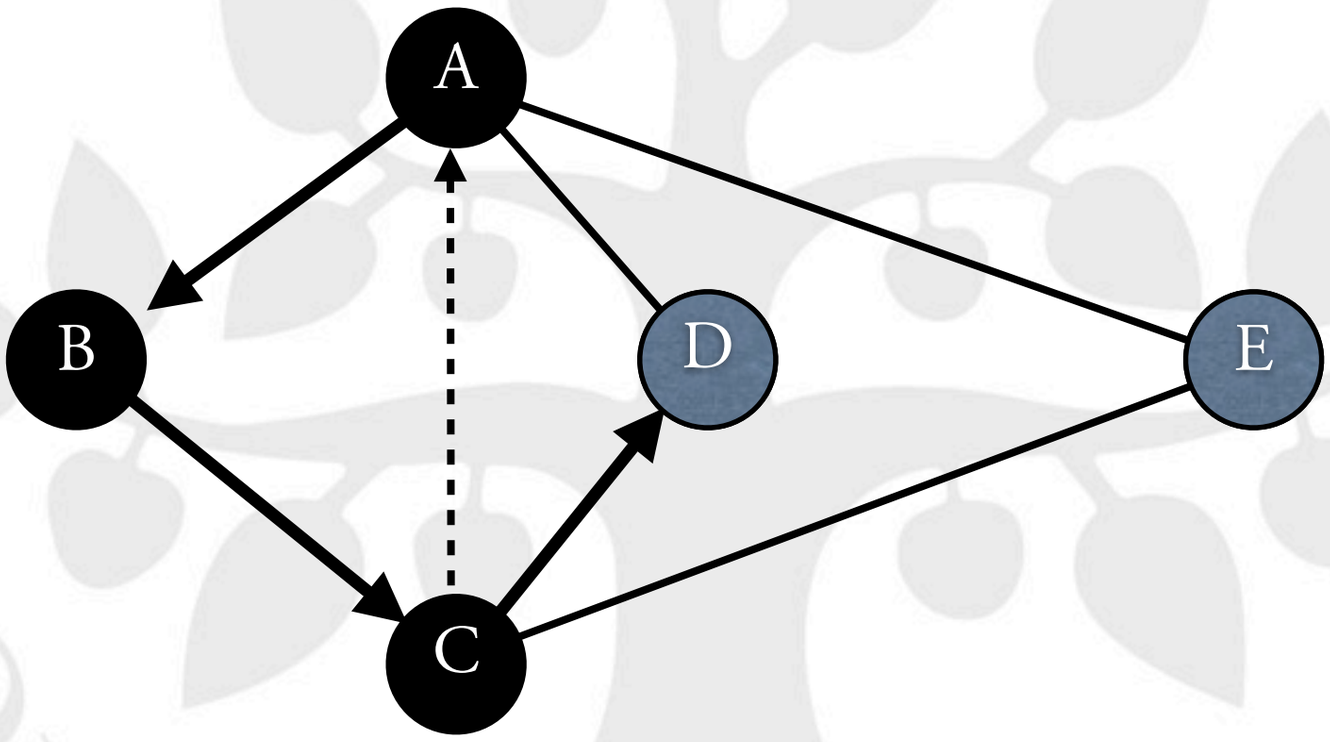
Eksempel



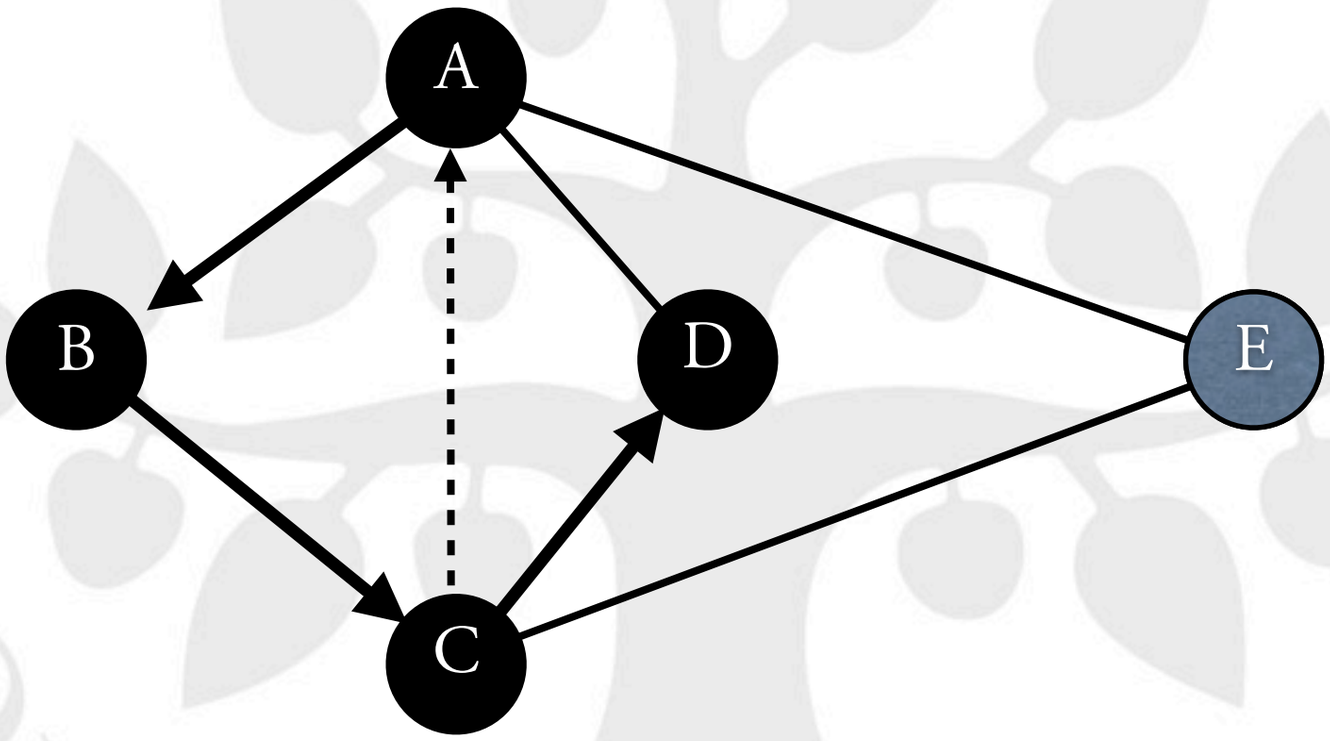
Eksempel



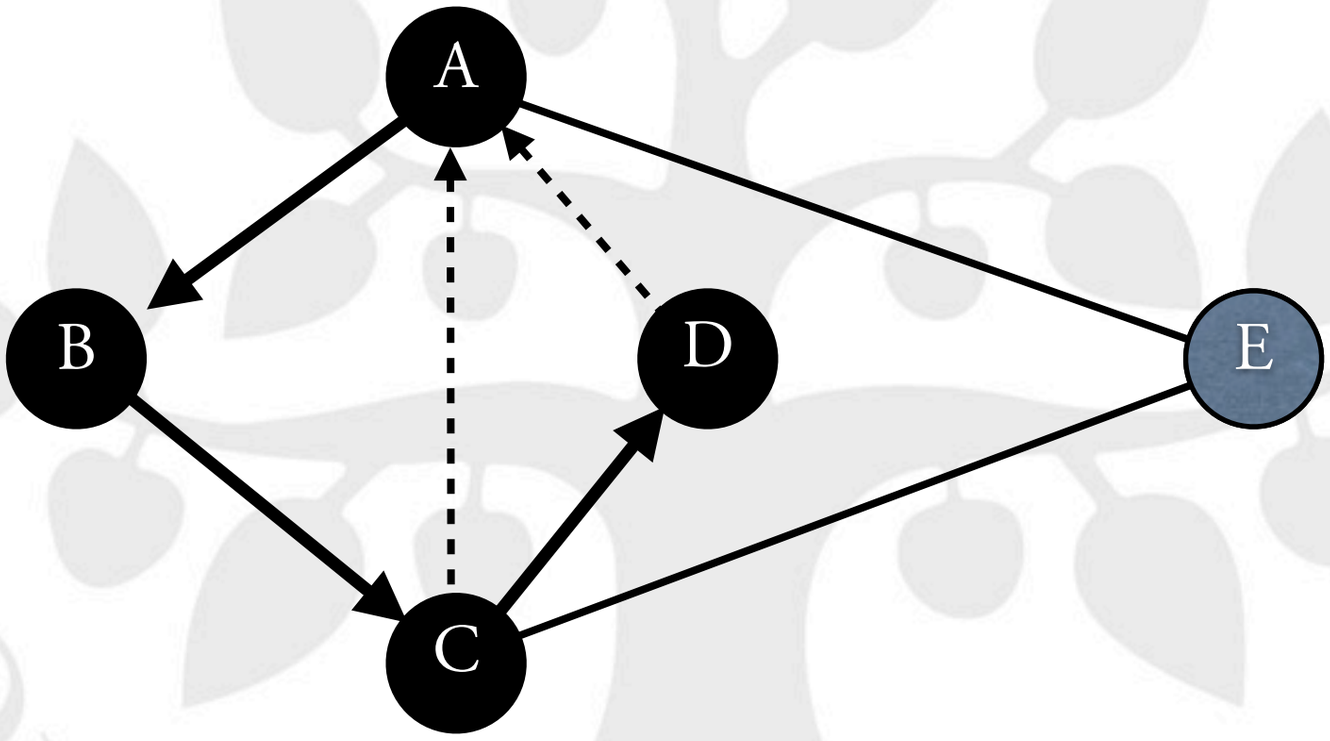
Eksempel



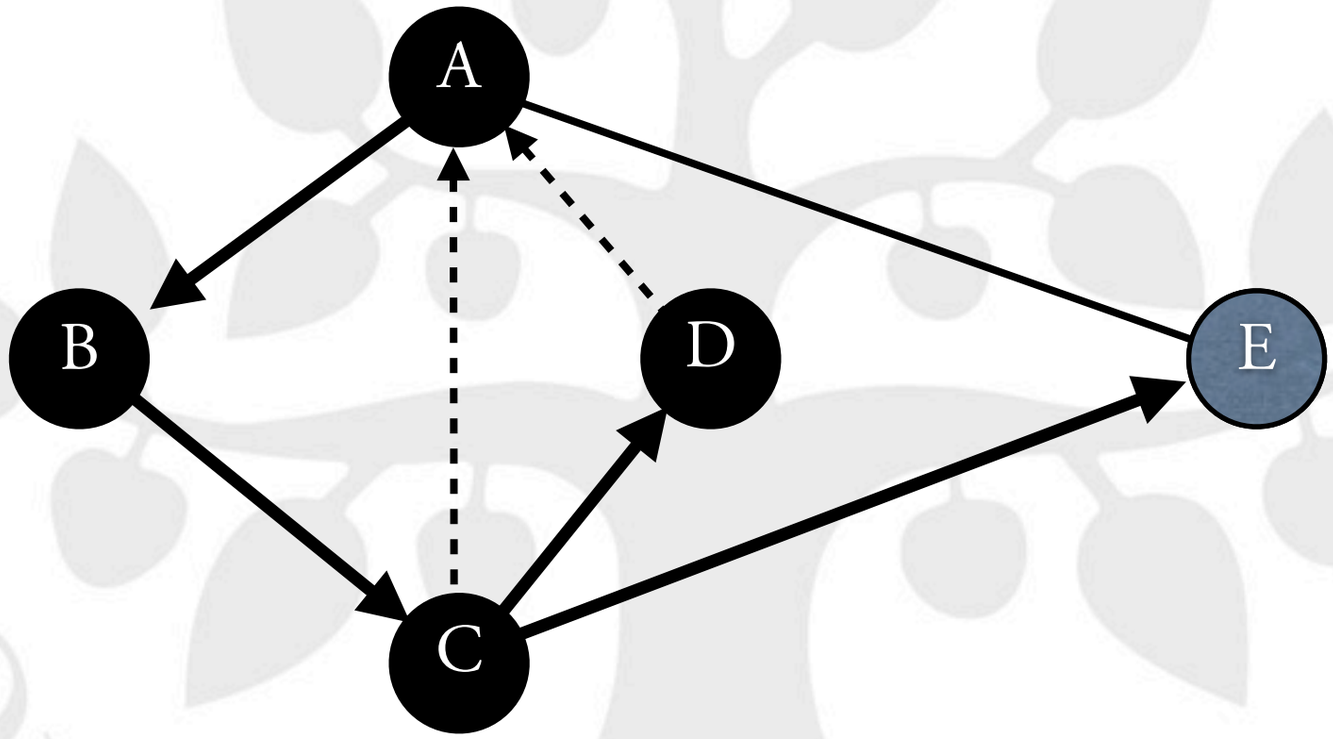
Eksempel



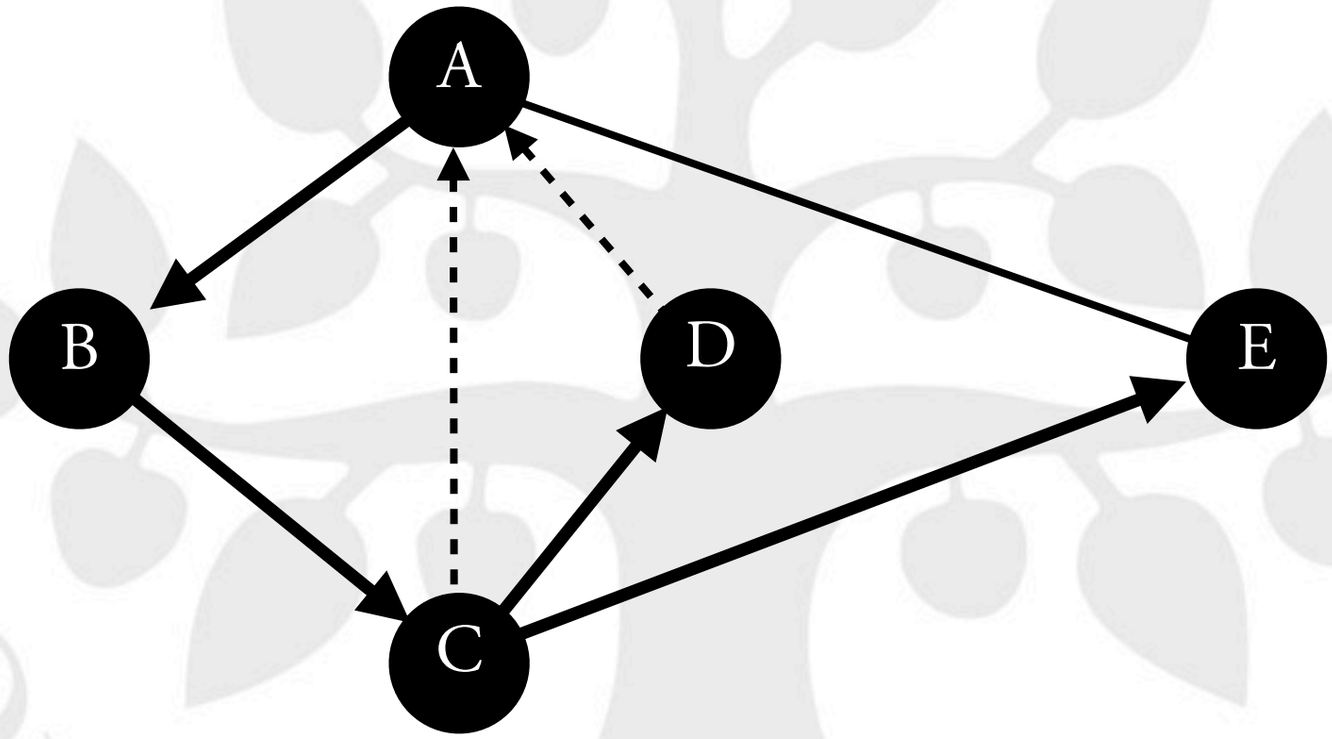
Eksempel



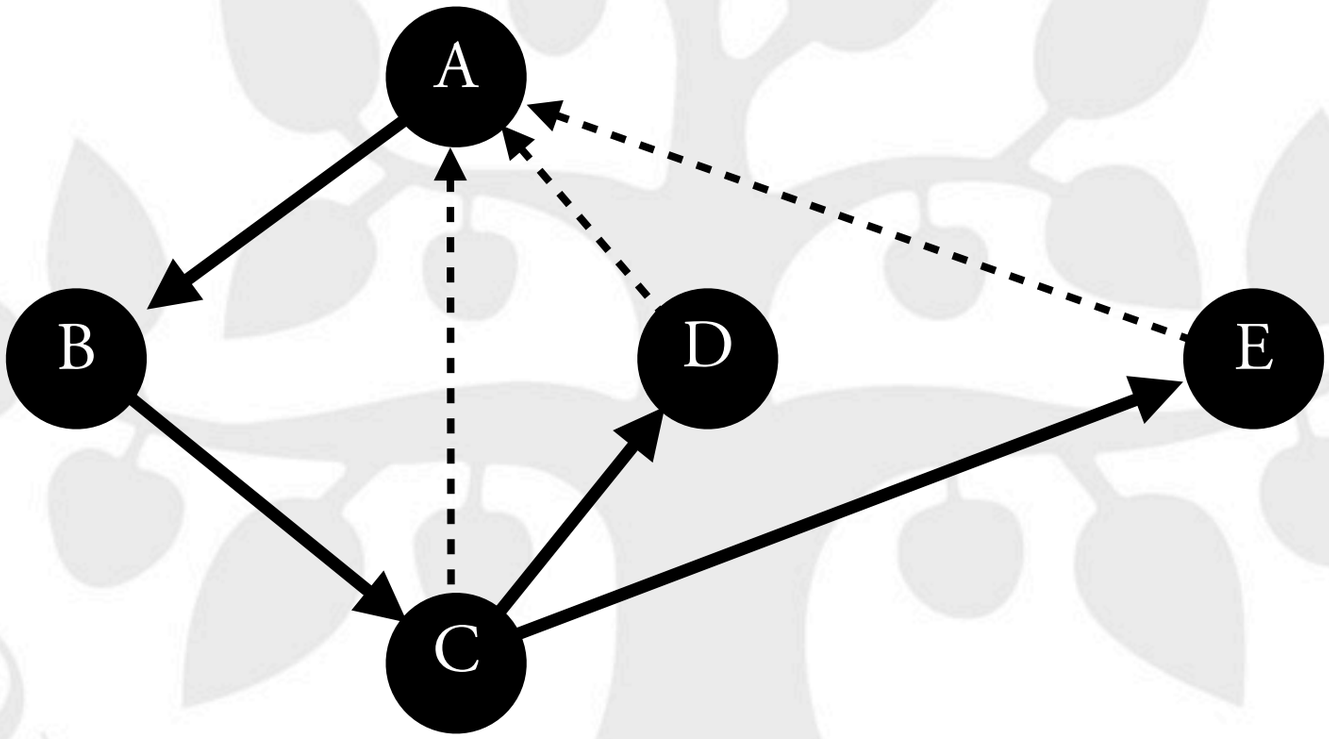
Eksempel



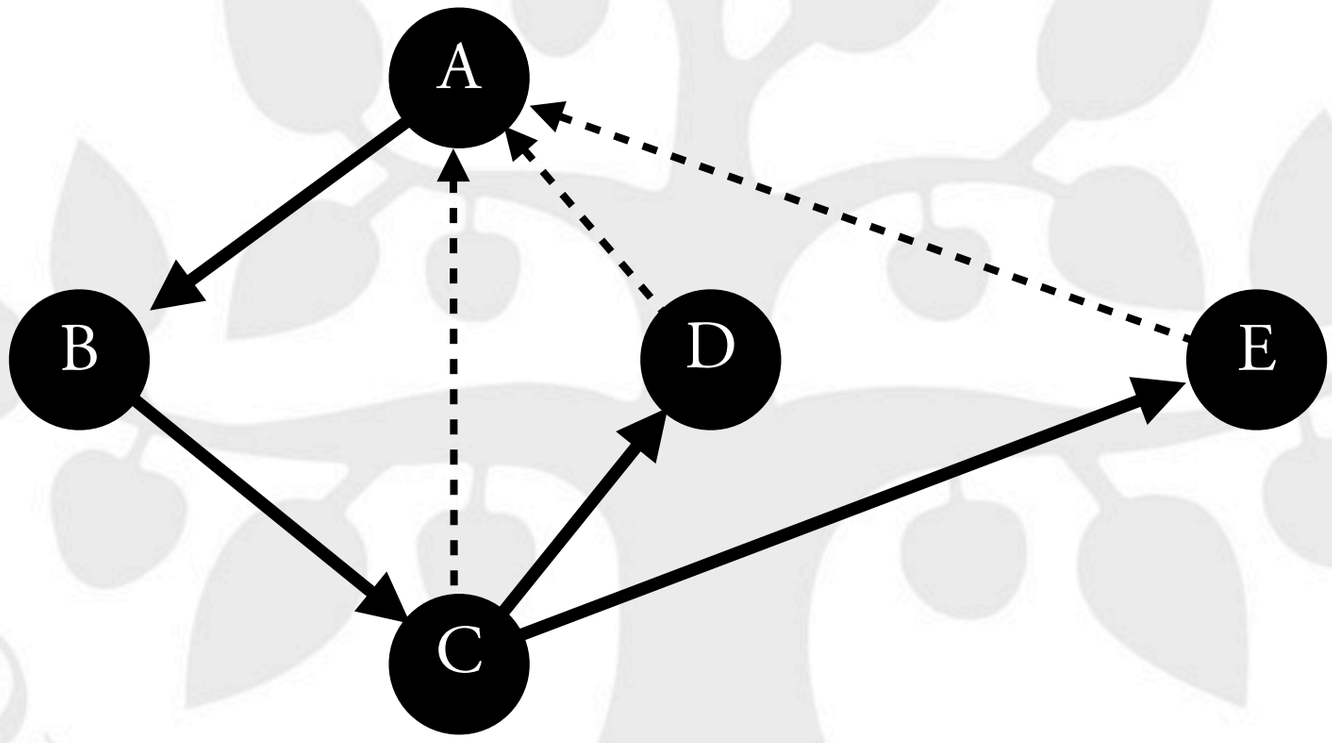
Eksempel



Eksempel

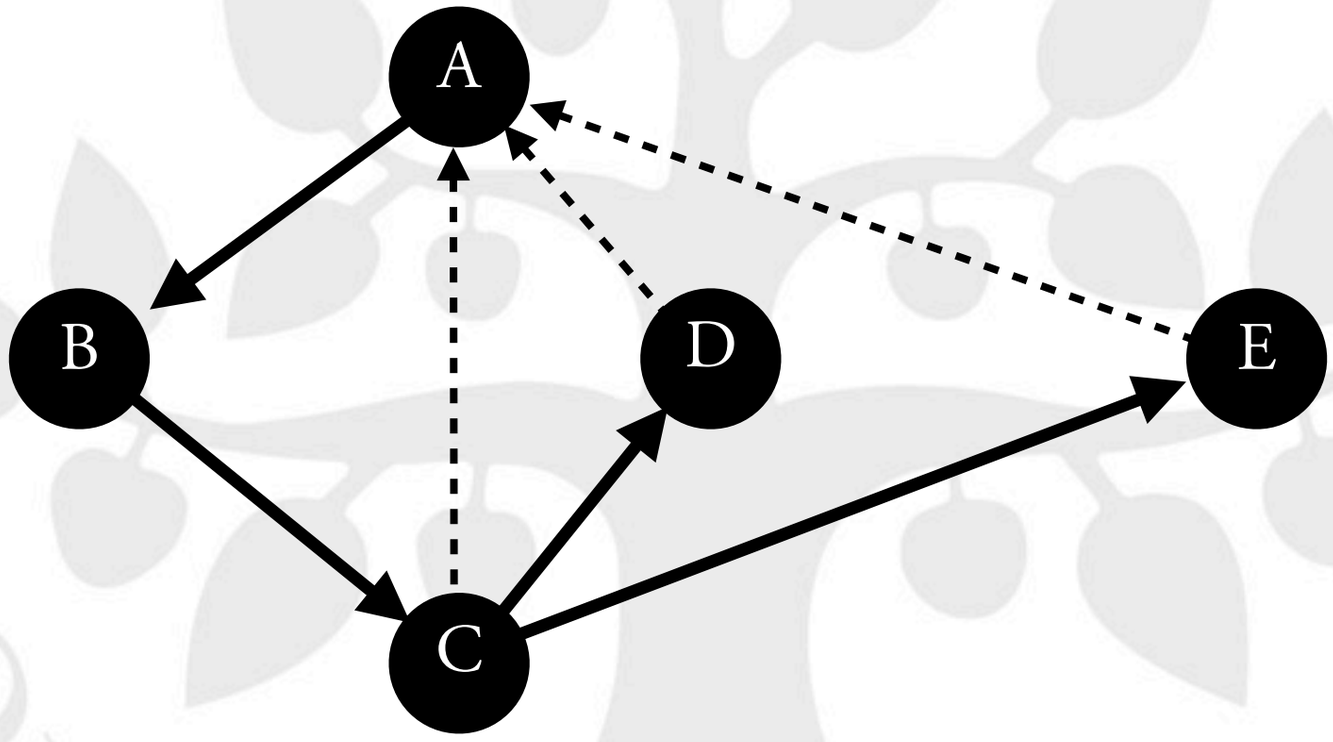


Eksempel



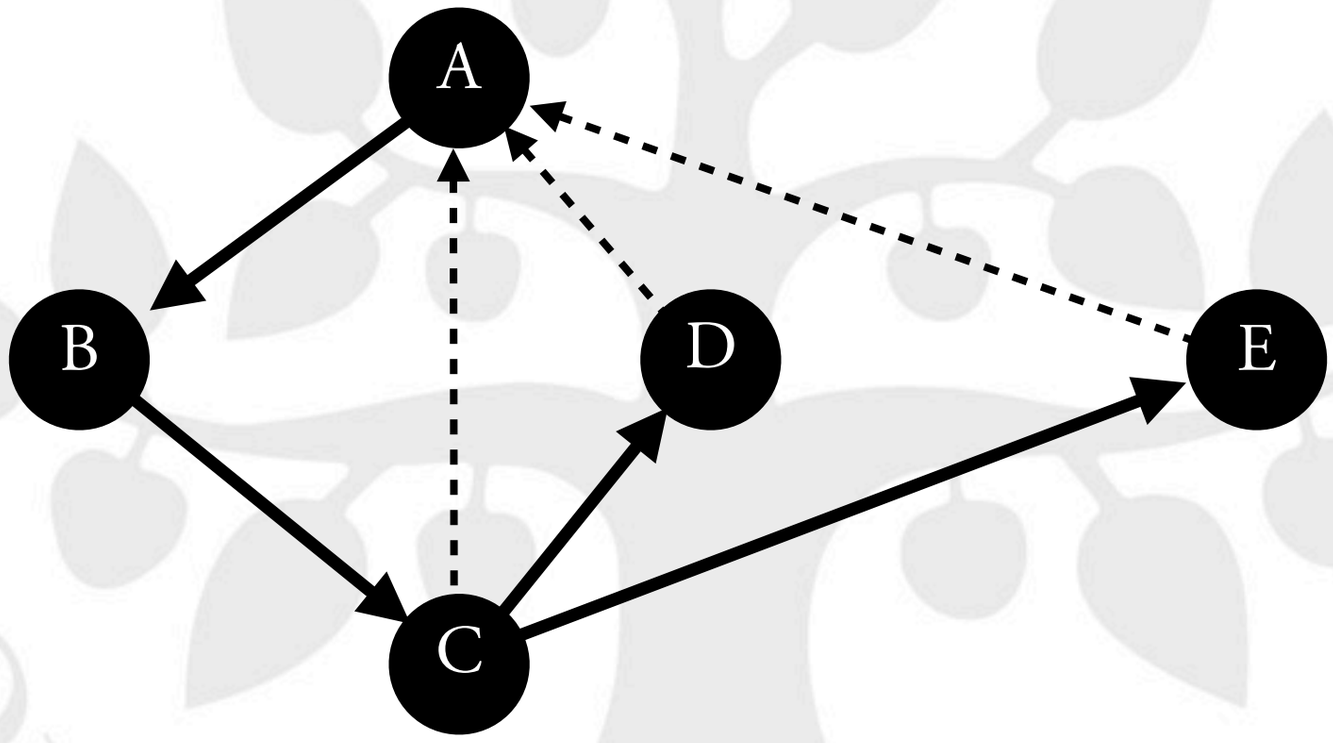
- Hvordan kan DFS udvides til at

Eksempel



- Hvordan kan DFS udvides til at
  - finde en kreds i grafen?

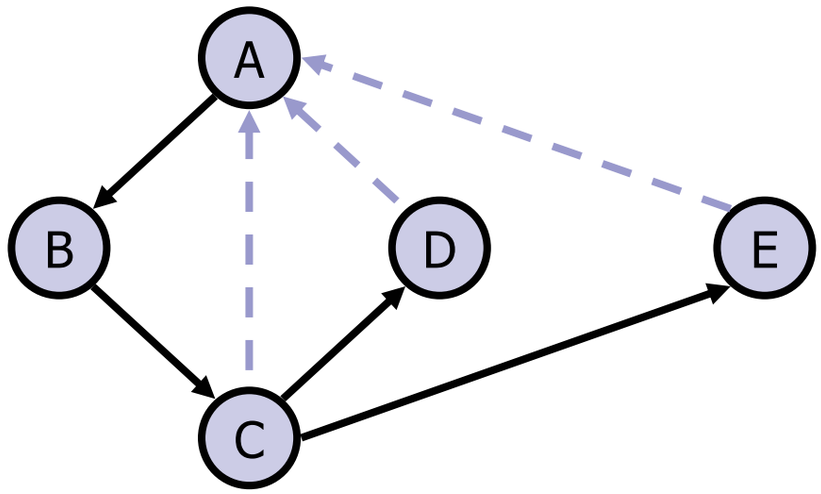
Eksempel



- Hvordan kan DFS udvides til at
  - finde en kreds i grafen?
  - beregne en sti mellem to givne knuder?

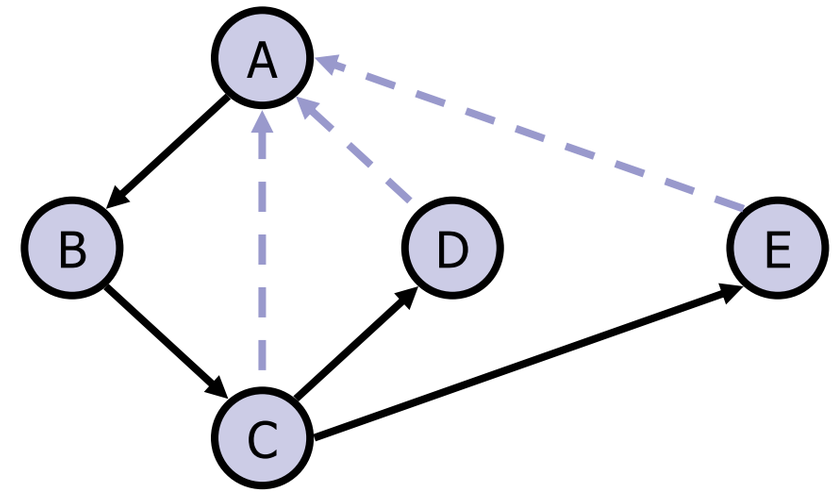
DFS's egenskaber

DFS's egenskaber



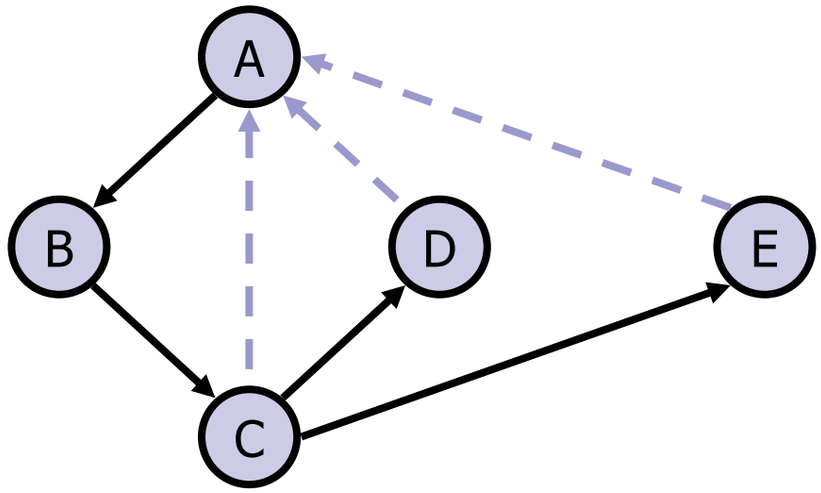
DFS's egenskaber

- Egenskab 1



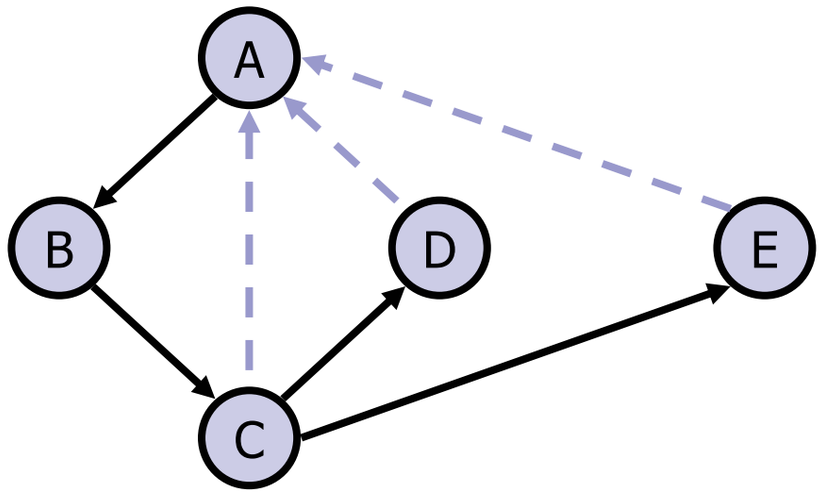
DFS's egenskaber

- Egenskab 1
  - DFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af



DFS's egenskaber

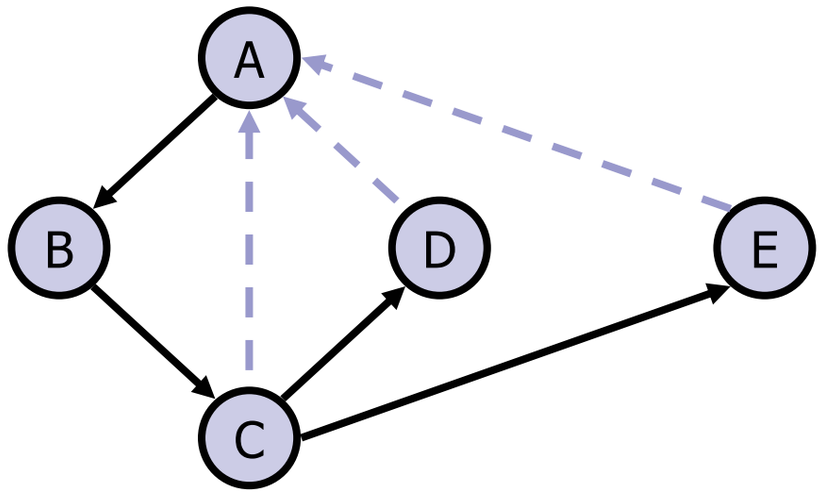
- Egenskab 1
  - DFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af



- Egenskab 2

DFS's egenskaber

- Egenskab 1
  - DFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af



- Egenskab 2
  - De kanter DFS(G, v) mærker som "besøgt" danner et udspændende træ af den sammenhængskomponent v er en del af

Node.java

```
import java.util.ArrayList;
public class Node {
    private ArrayList<Node> neighbours;
    private String name;
    private boolean mark;

    public Node( String n ) {
        neighbours = new ArrayList<Node>();
        name = n;
        mark = false;
    }
    public void addNeighbour( Node v ) {
        neighbours.add( v );
    }
    public ArrayList<Node> getNeighbours() {
        return neighbours;
    }
    public void setMark( boolean value ) {
        mark = value;
    }
    public boolean getMark() {
        return mark;
    }
    public String toString() {
        return name;
    }
}
```

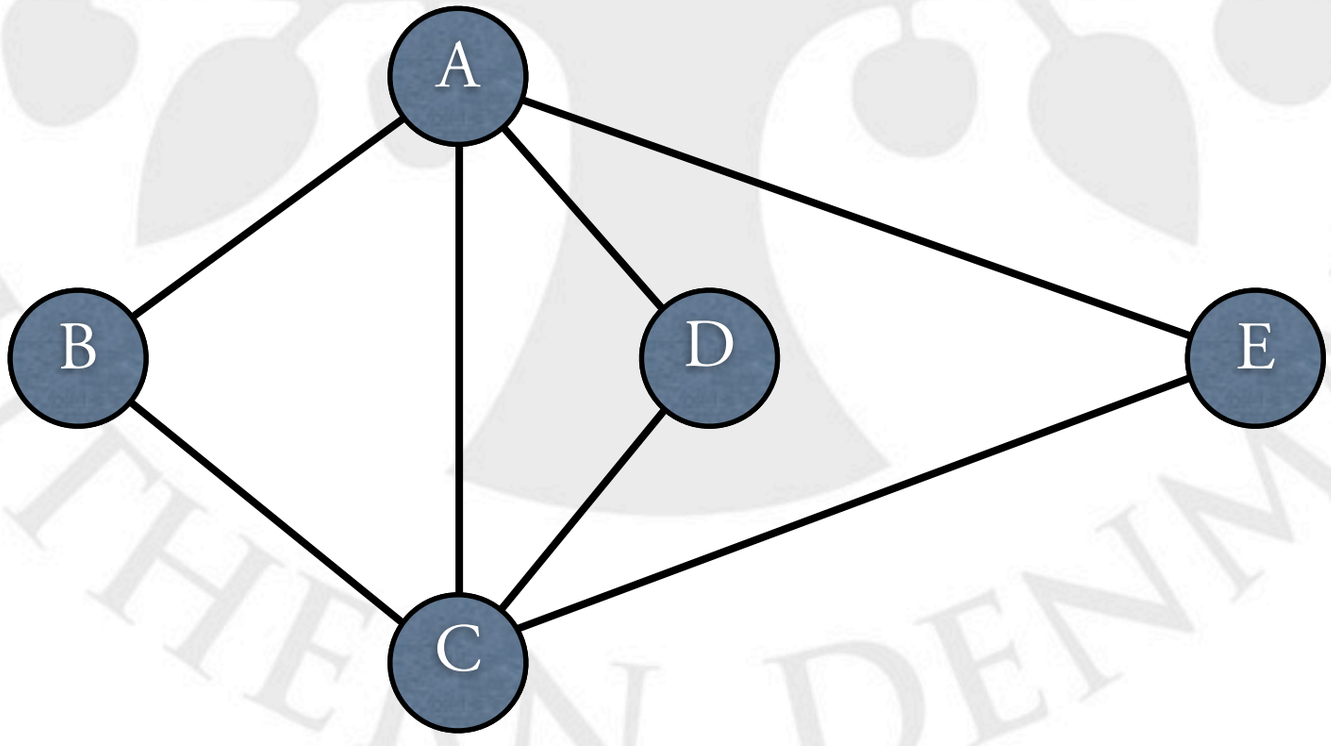
DFS.java

```
import java.util.ArrayList;
public class DFS {
    public static void main( String[] args ) {
        Node a = new Node( "A" );
        Node b = new Node( "B" );
        Node c = new Node( "C" );
        Node d = new Node( "D" );
        Node e = new Node( "E" );

        a.addNeighbour( b );
        a.addNeighbour( c );
        a.addNeighbour( d );
        a.addNeighbour( e );

        b.addNeighbour( a );
        b.addNeighbour( c );

        c.addNeighbour( a );
        c.addNeighbour( b );
        c.addNeighbour( d );
        c.addNeighbour( e );

        d.addNeighbour( a );
        d.addNeighbour( c );

        e.addNeighbour( a );
        e.addNeighbour( c );

        DFS( a );
    }
```

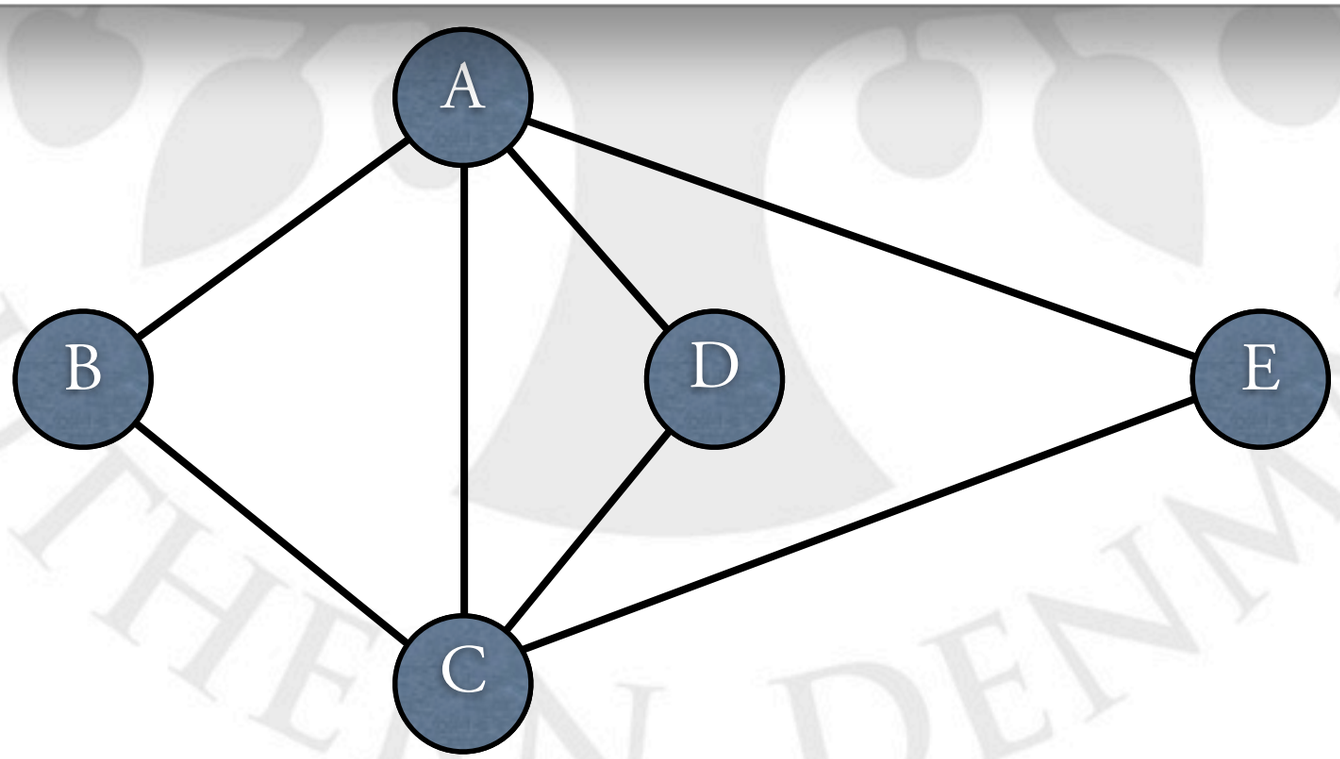
DFS.java (cont.)

```
    public static void DFS( Node v ) {
        v.setMark( true );
        for( Node w : v.getNeighbours() ) {
            if( !w.getMark() ) {
                System.out.println( v + ": visiting " + w + " next" );
                DFS( w );
            } else {
                System.out.println( v + ": " + w + " already visited" );
            }
        }
    }
}
```

Output



```
Terminal — bash — 66×17
MAC-SDU-00001:7 petersk$ javac DFS.java
MAC-SDU-00001:7 petersk$ java DFS
A: visiting B next
B: A already visited
B: visiting C next
C: A already visited
C: B already visited
C: visiting D next
D: A already visited
D: C already visited
C: visiting E next
E: A already visited
E: C already visited
A: C already visited
A: D already visited
A: E already visited
MAC-SDU-00001:7 petersk$
```



Brede-først-søgning

- BFS
  - Algoritmen
  - Eksempel
  - Egenskaber
  - Java-kode

Brede-først-søgning

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- BFS tager tid der er proportionalt med n+m

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- BFS tager tid der er proportionalt med n+m
- BFS kan udvides til at

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- BFS tager tid der er proportionalt med n+m
- BFS kan udvides til at
  - Beregne den korteste sti mellem to givne knuder (hvis der findes en sti)

Brede-først-søgning

- Brede-først-søgning (BFS) er en generel teknik til at gennemsøge en graf
  - Besøger alle knuder og kanter
  - Afgør om grafen er sammenhægende
  - Beregner sammenhængskomponenterne
  - Beregner en udspændende skov for grafen
- BFS tager tid der er proportionalt med n+m
- BFS kan udvides til at
  - Beregne den korteste sti mellem to givne knuder (hvis der findes en sti)
  - Finde en kreds i grafen (hvis der er en)

BFS-algoritmen

BFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne

BFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder

BFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G

BFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt

BFS-algoritmen

- Algoritmen sætter "mærker" på knuderne og kanterne
- Sæt mærket "ubesøgt" på alle kanter og knuder
- For alle knuder v i G
  - Hvis v er ubesøgt
    - BFS(G, v)

BFS-algoritmen

BFS-algoritmen

- BFS(G, v):

BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder

BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"

BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"
  - Indsæt v i q

BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"
  - Indsæt v i q
  - Sålænge q ikke er tom

BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"
  - Indsæt v i q
  - Sålænge q ikke er tom
    - u = den næste knude i q

BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"
  - Indsæt v i q
  - Sålænge q ikke er tom
    - u = den næste knude i q
    - Besøg u

BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"
  - Indsæt v i q
  - Sålænge q ikke er tom
    - u = den næste knude i q
    - Besøg u
  - For alle ikke-sete naboer w til u

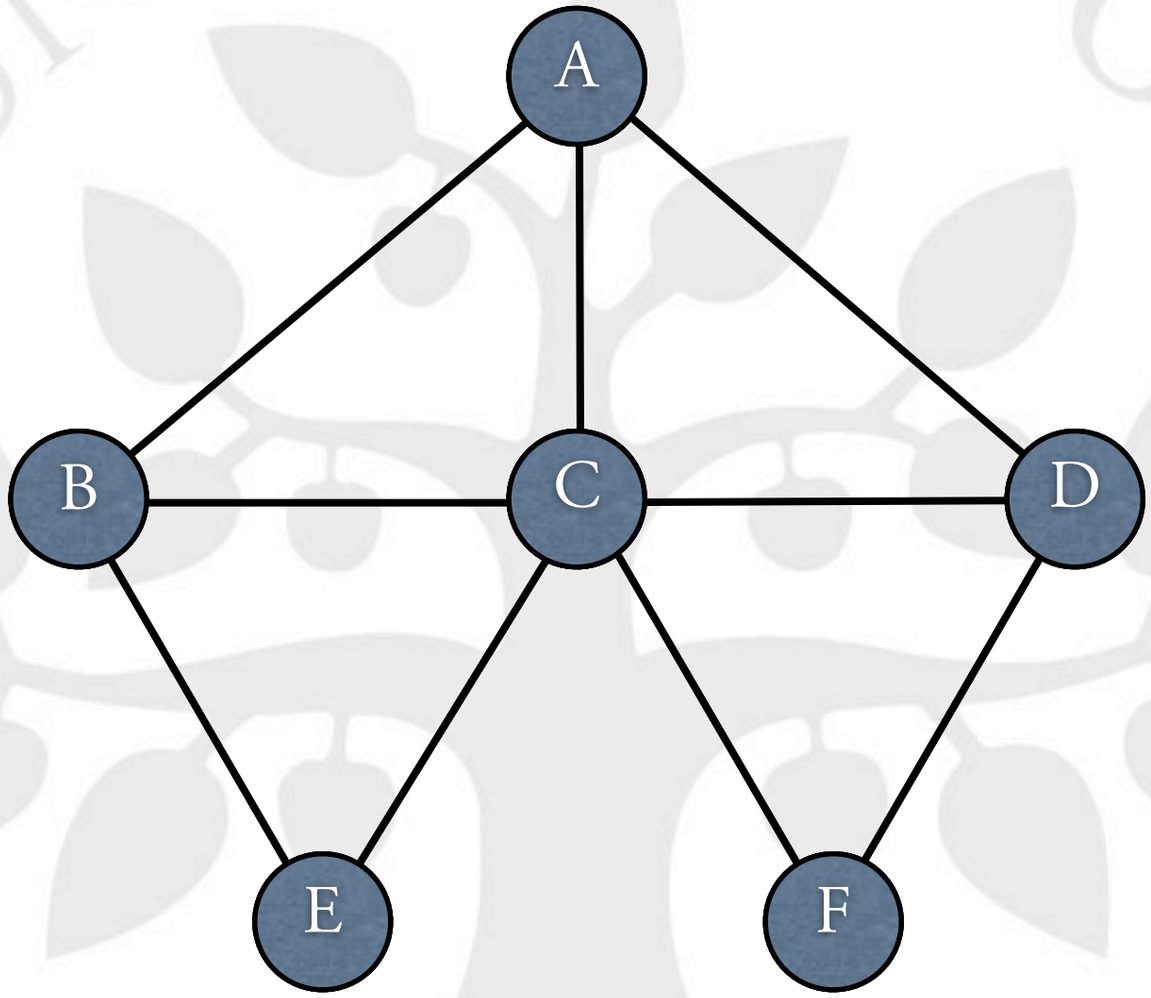
BFS-algoritmen

- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"
  - Indsæt v i q
  - Sålænge q ikke er tom
    - u = den næste knude i q
    - Besøg u
    - For alle ikke-sete naboer w til u
      - Marker w som "set"

BFS-algoritmen

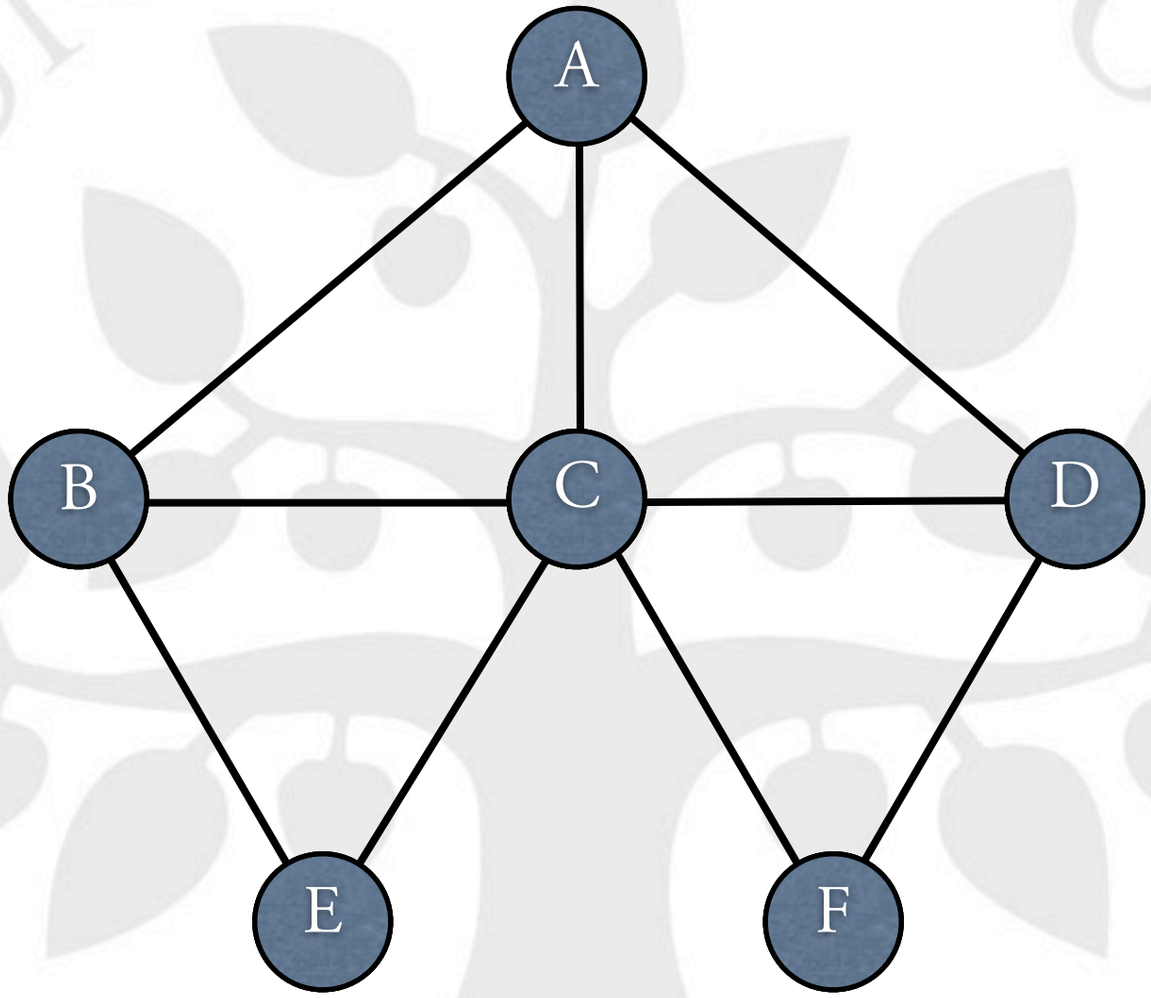
- BFS(G, v):
  - q = ny tom kø (FIFO) af knuder
  - Marker v som "set"
  - Indsæt v i q
  - Sålænge q ikke er tom
    - u = den næste knude i q
    - Besøg u
    - For alle ikke-sete naboer w til u
      - Marker w som "set"
      - Tilføj w til q

Eksempel

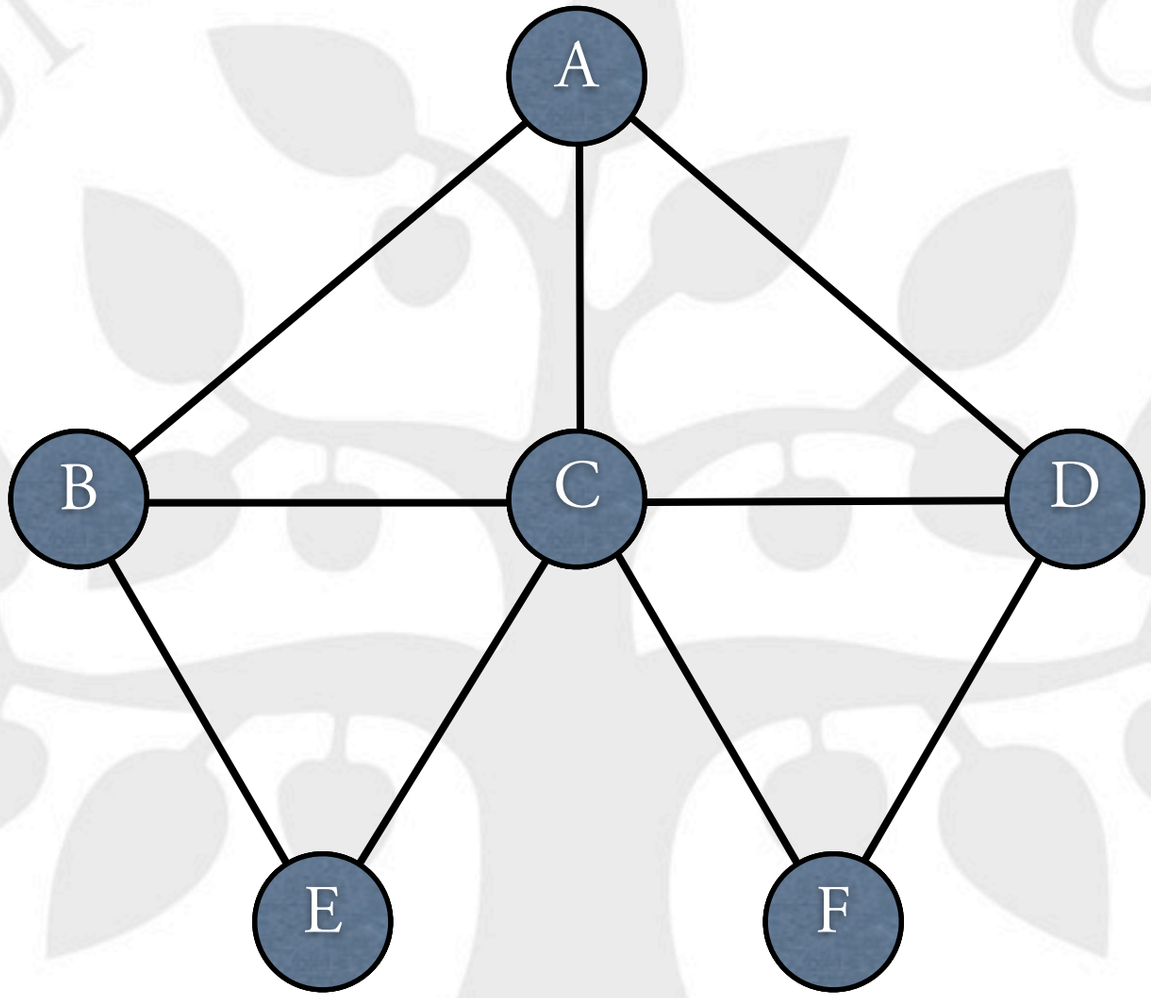


A

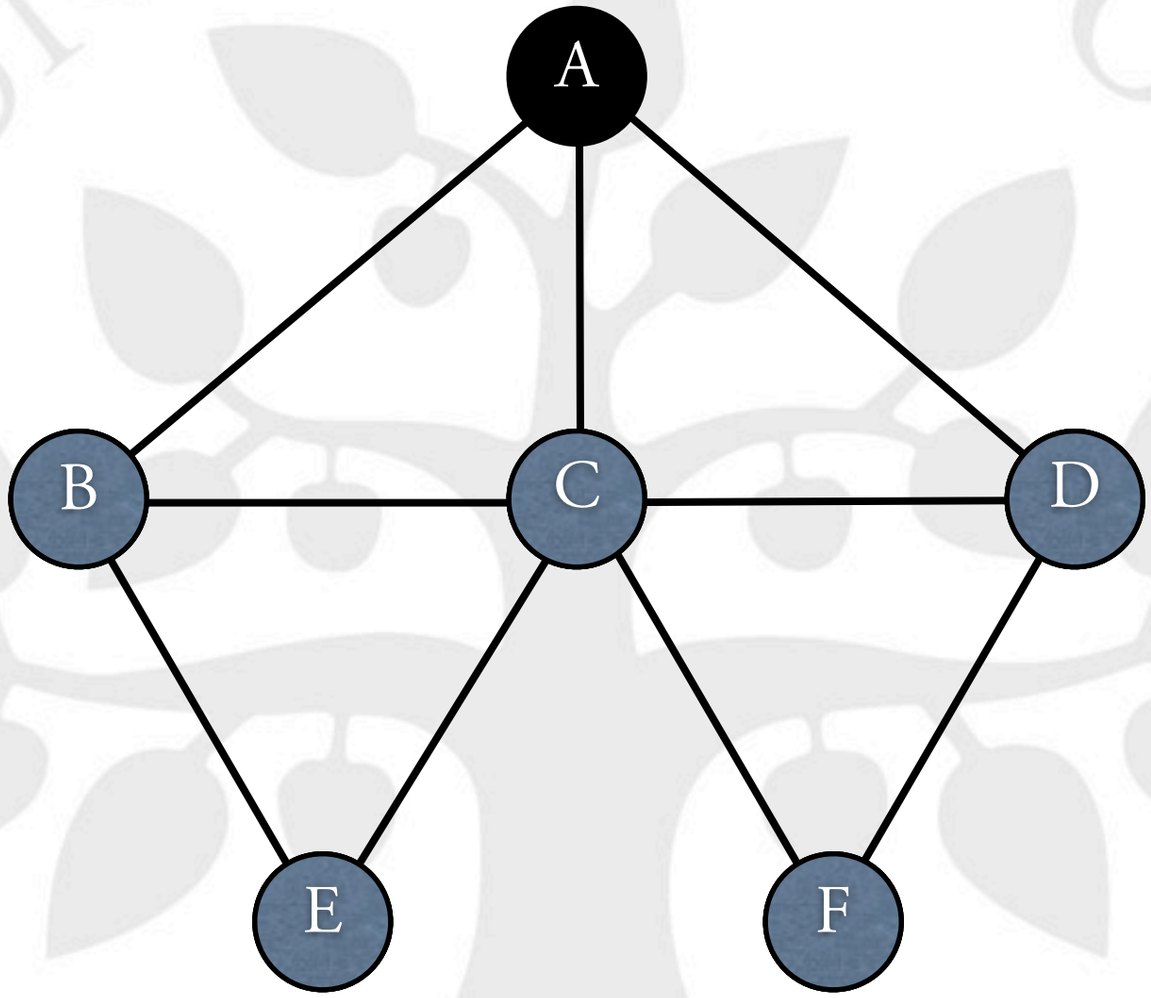
Eksempel



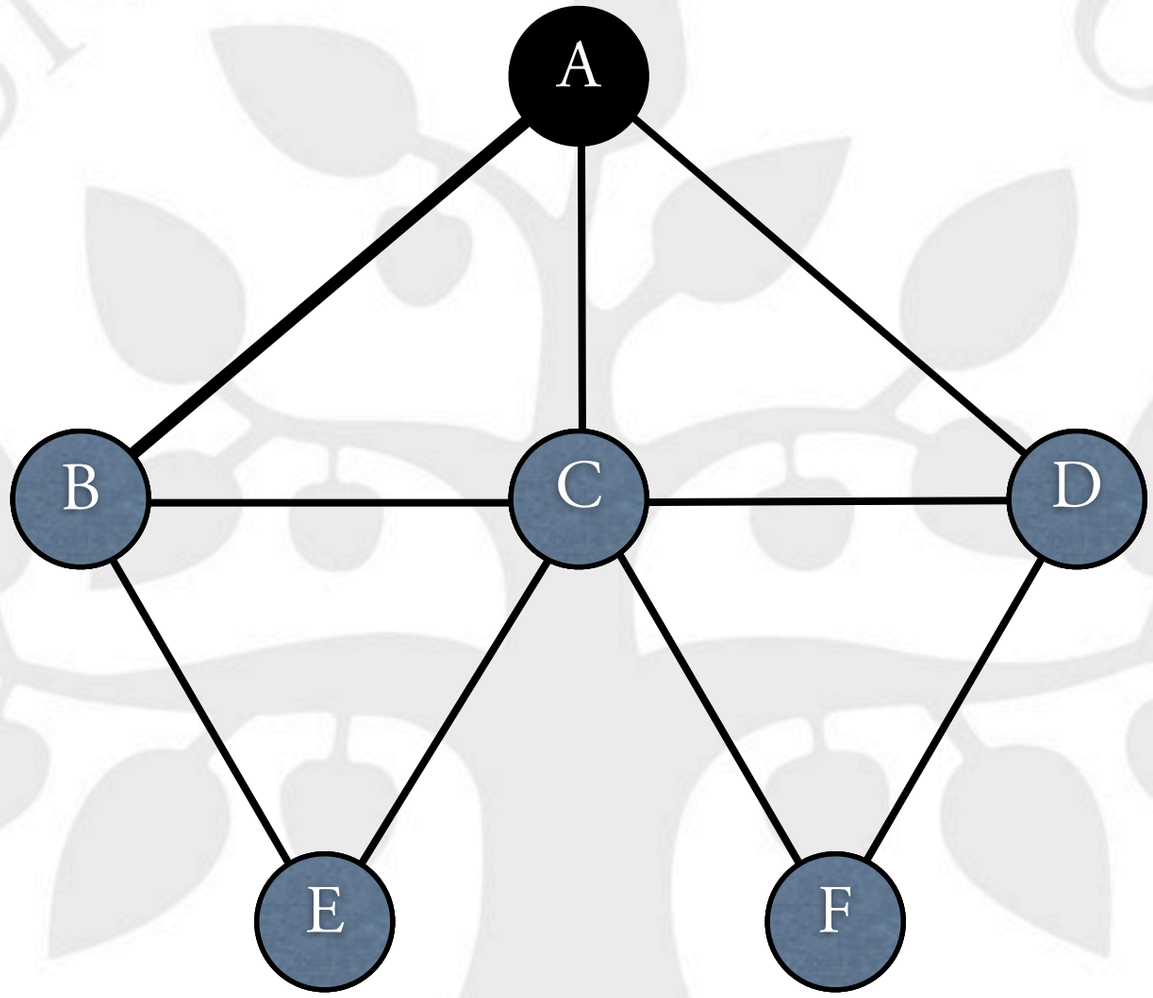
Eksempel



Eksempel

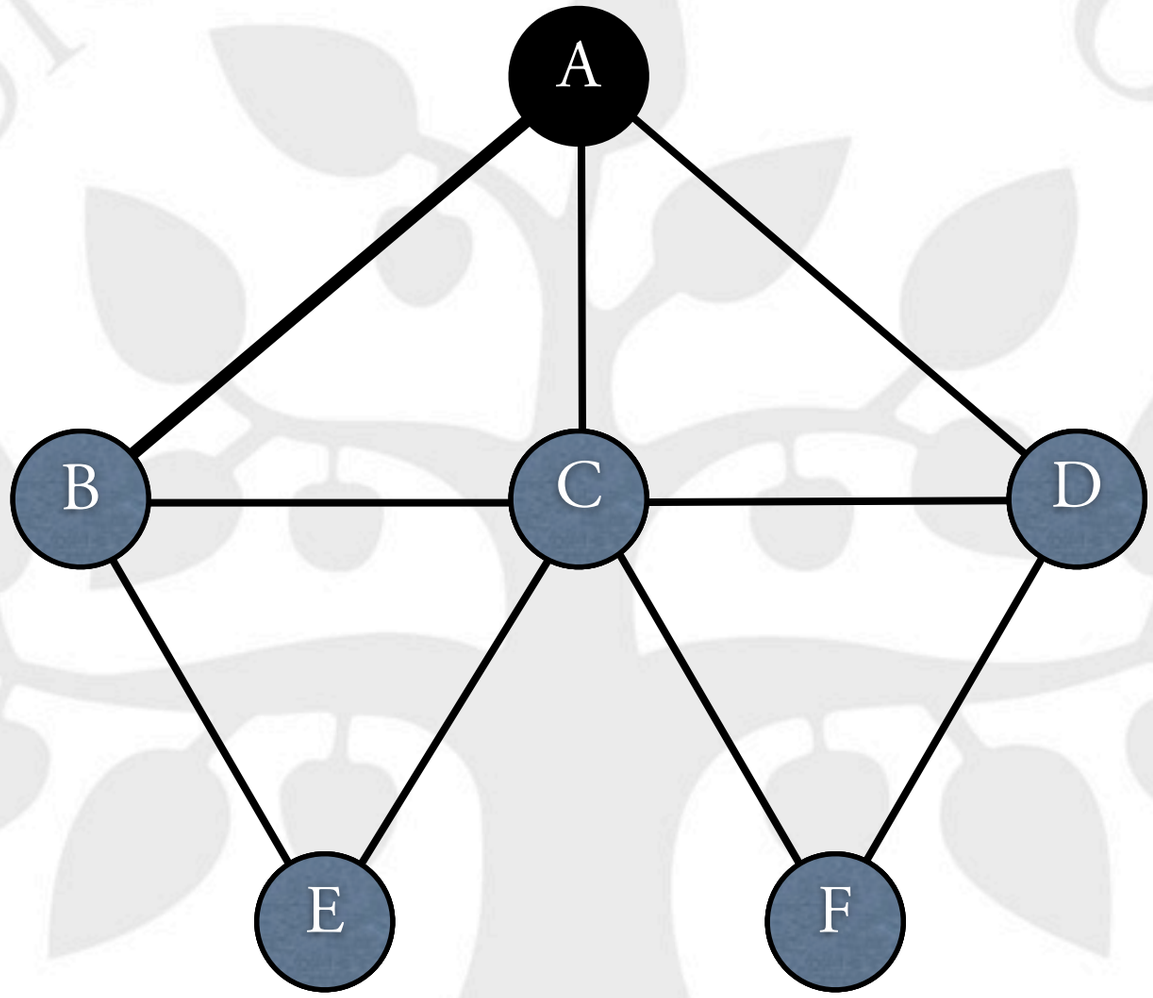


Eksempel

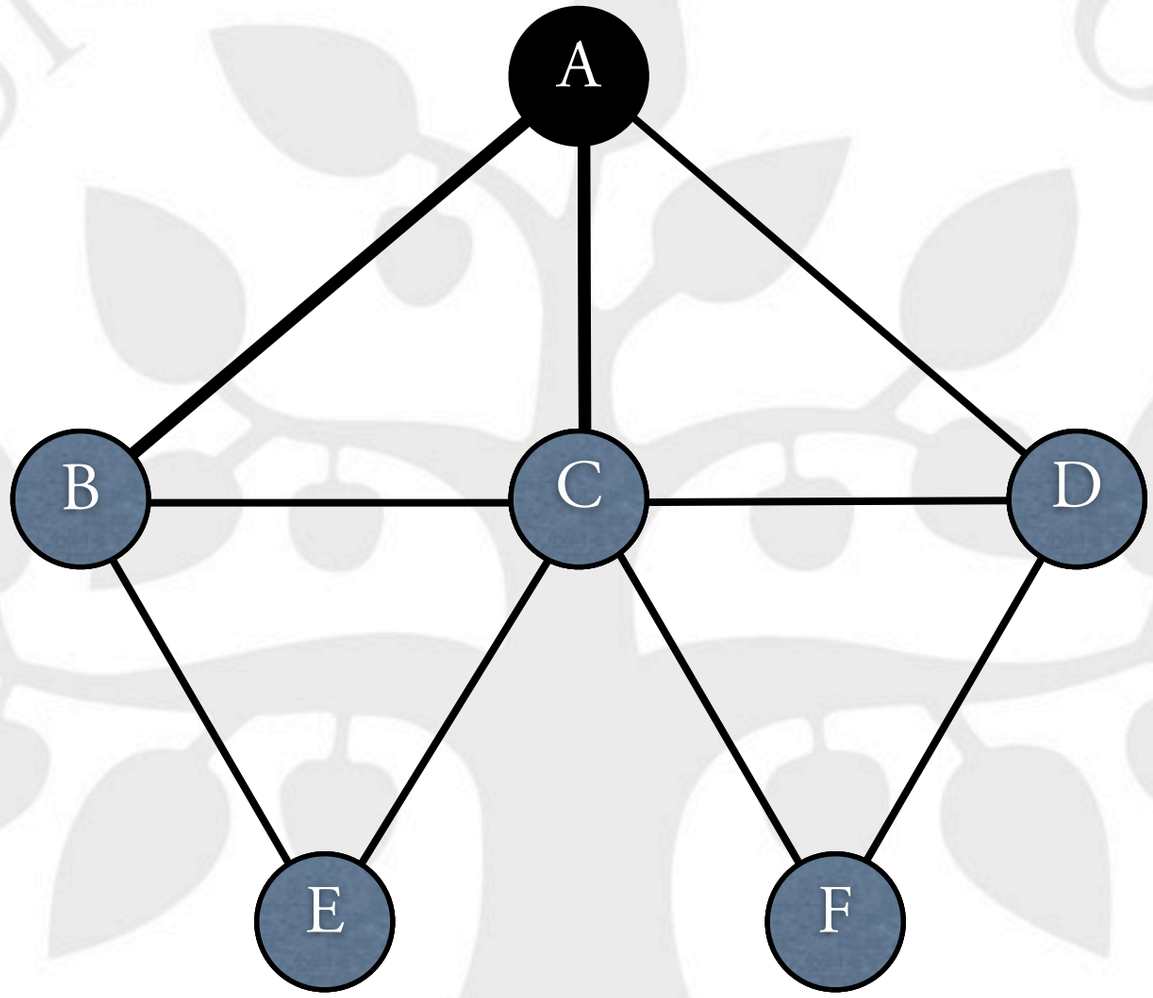


B

Eksempel

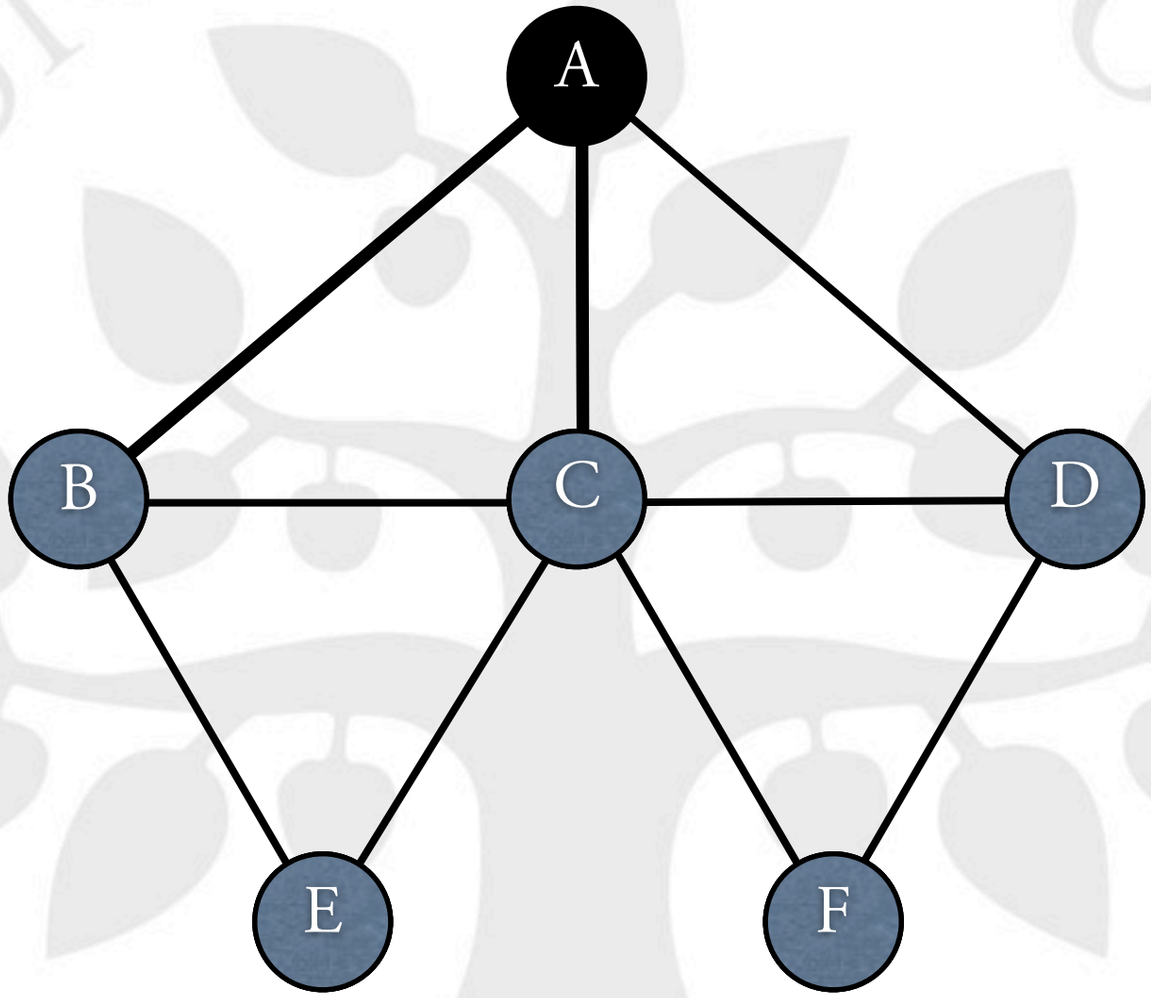


Eksempel



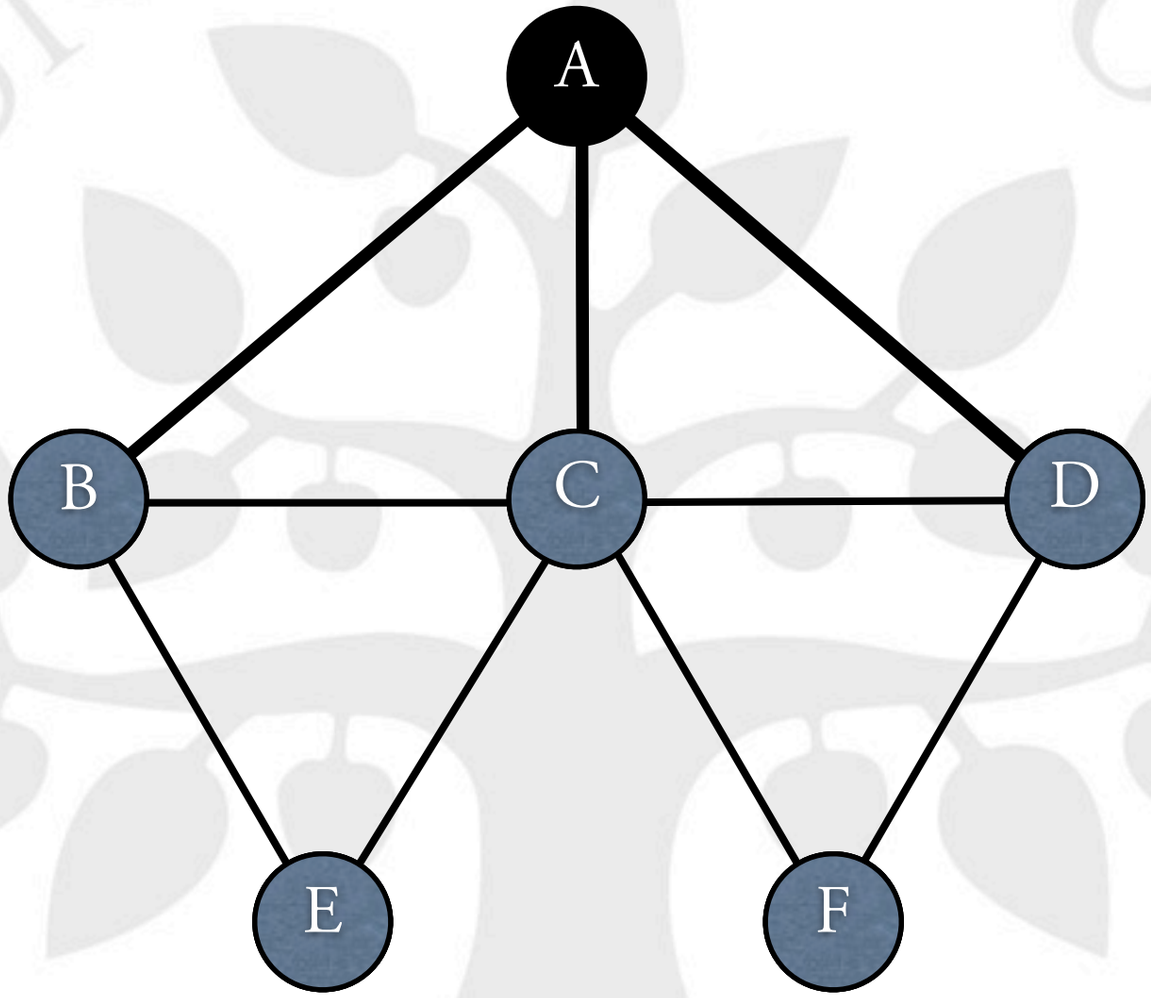
B C

Eksempel



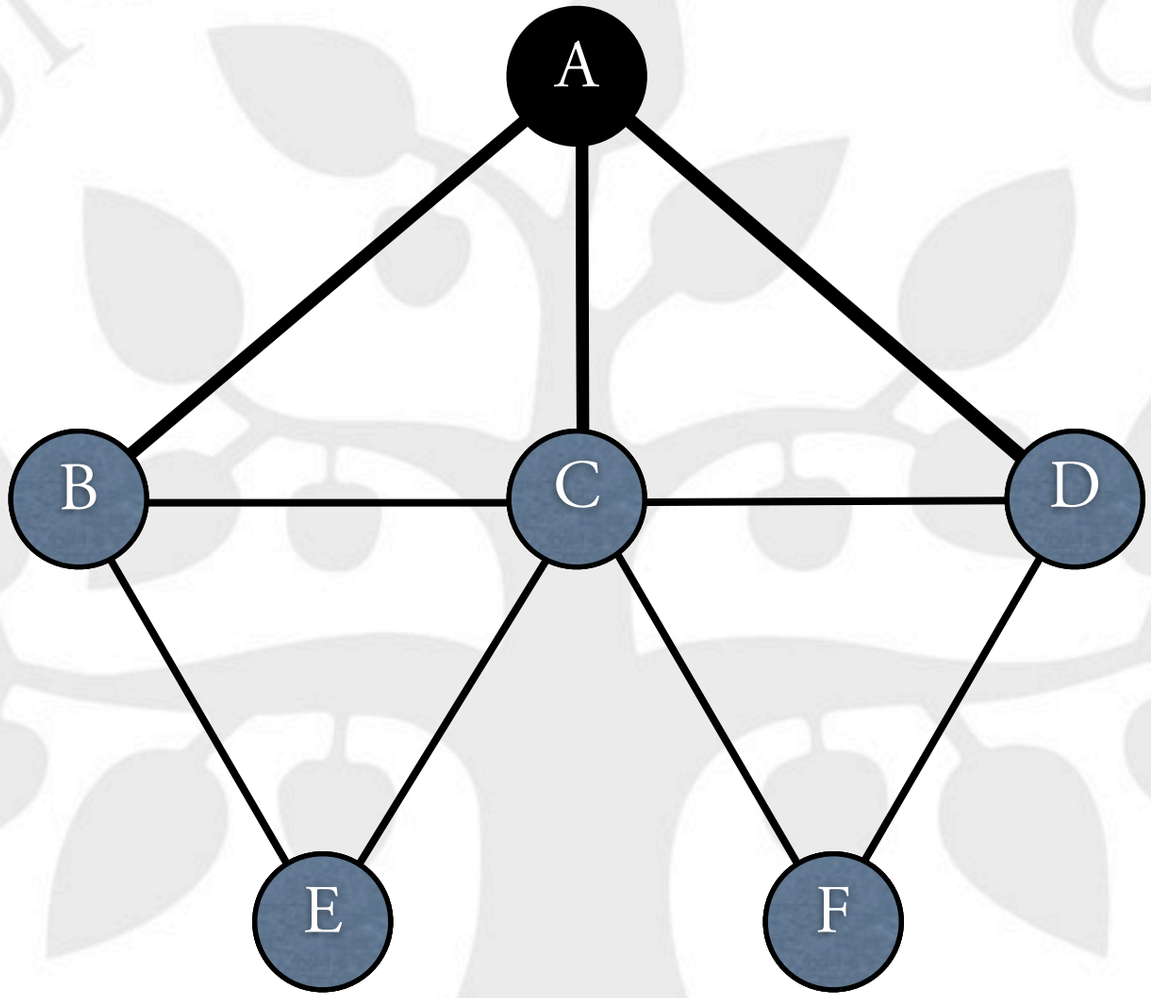
B C

Eksempel



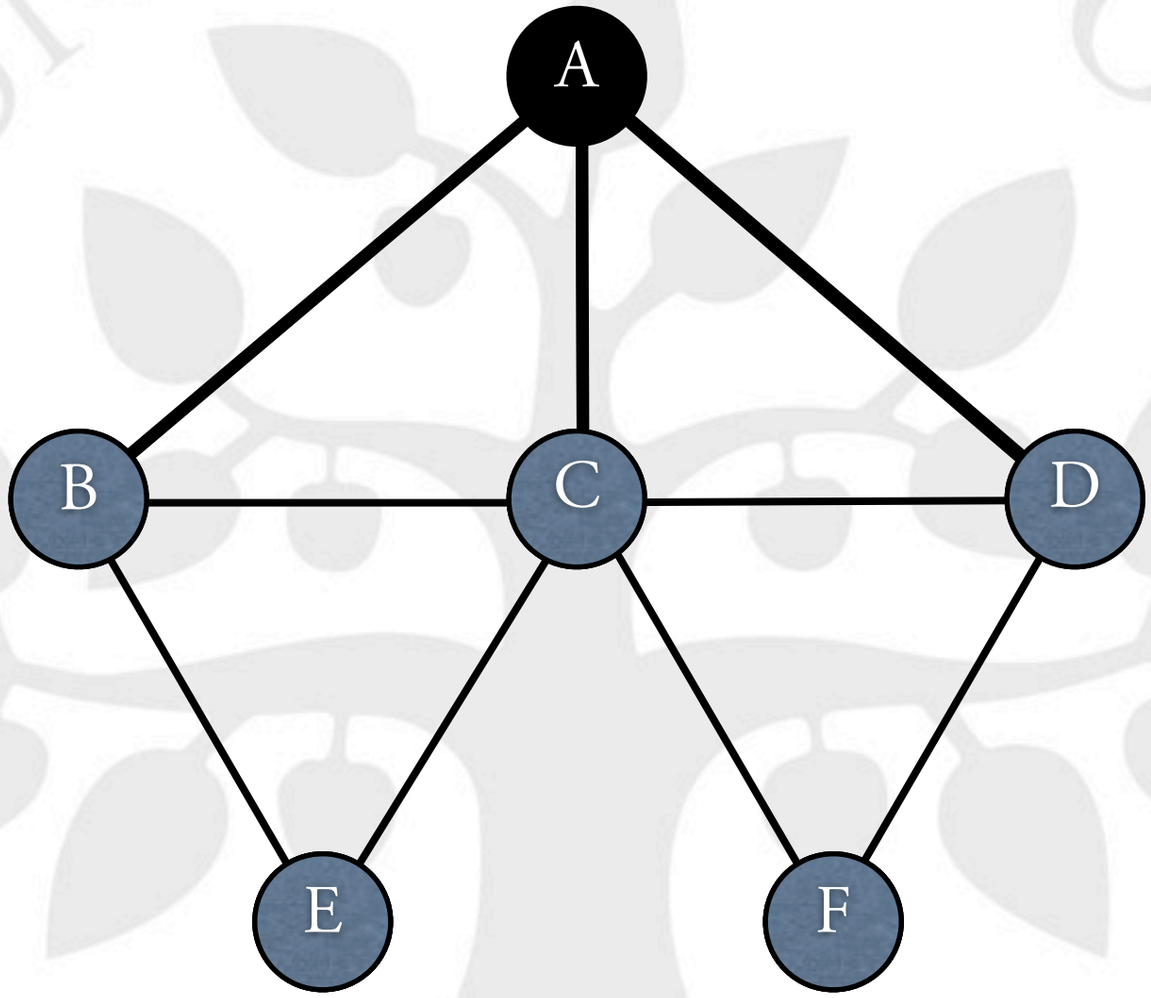
B C D

Eksempel



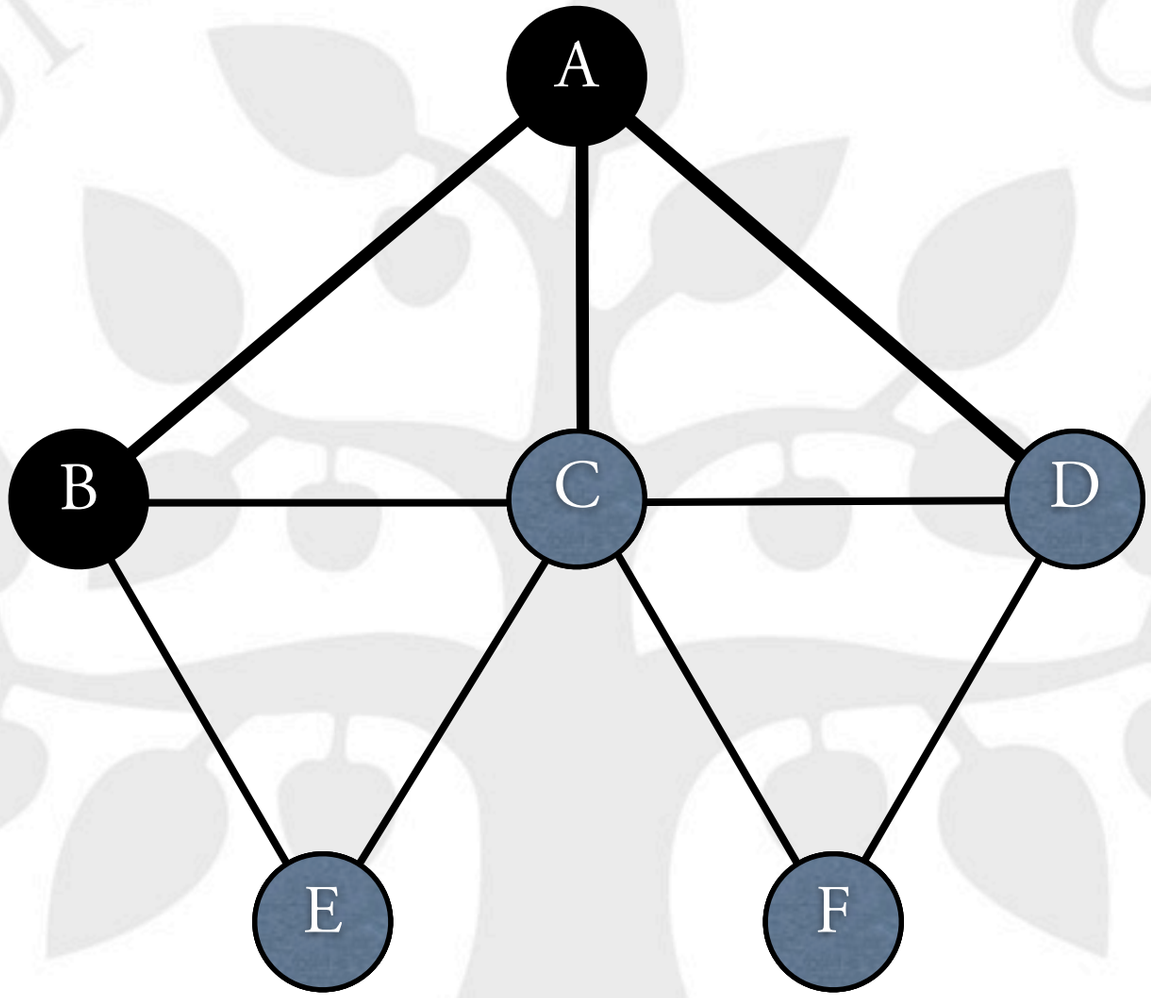
C D

Eksempel



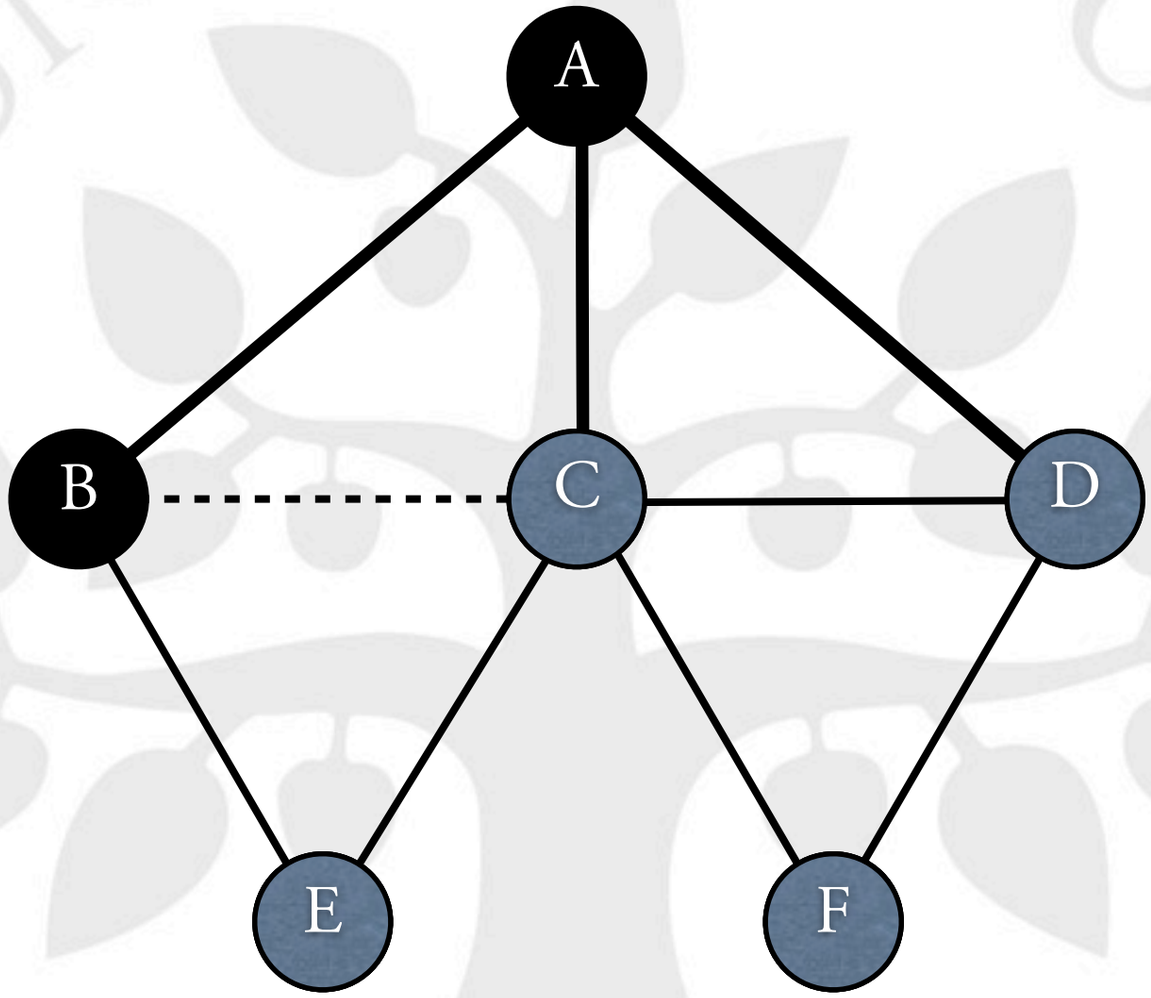
C D

Eksempel



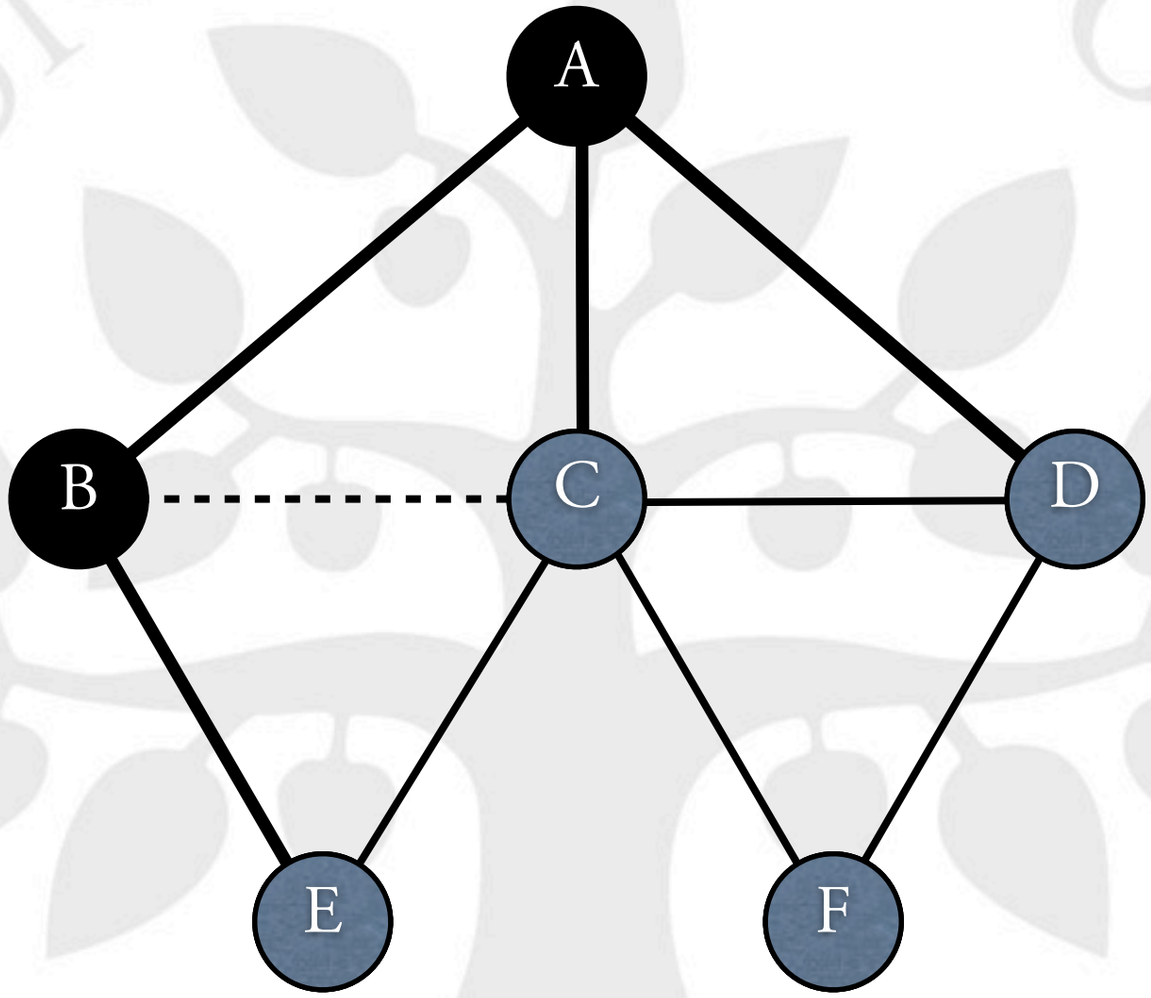
C D

Eksempel



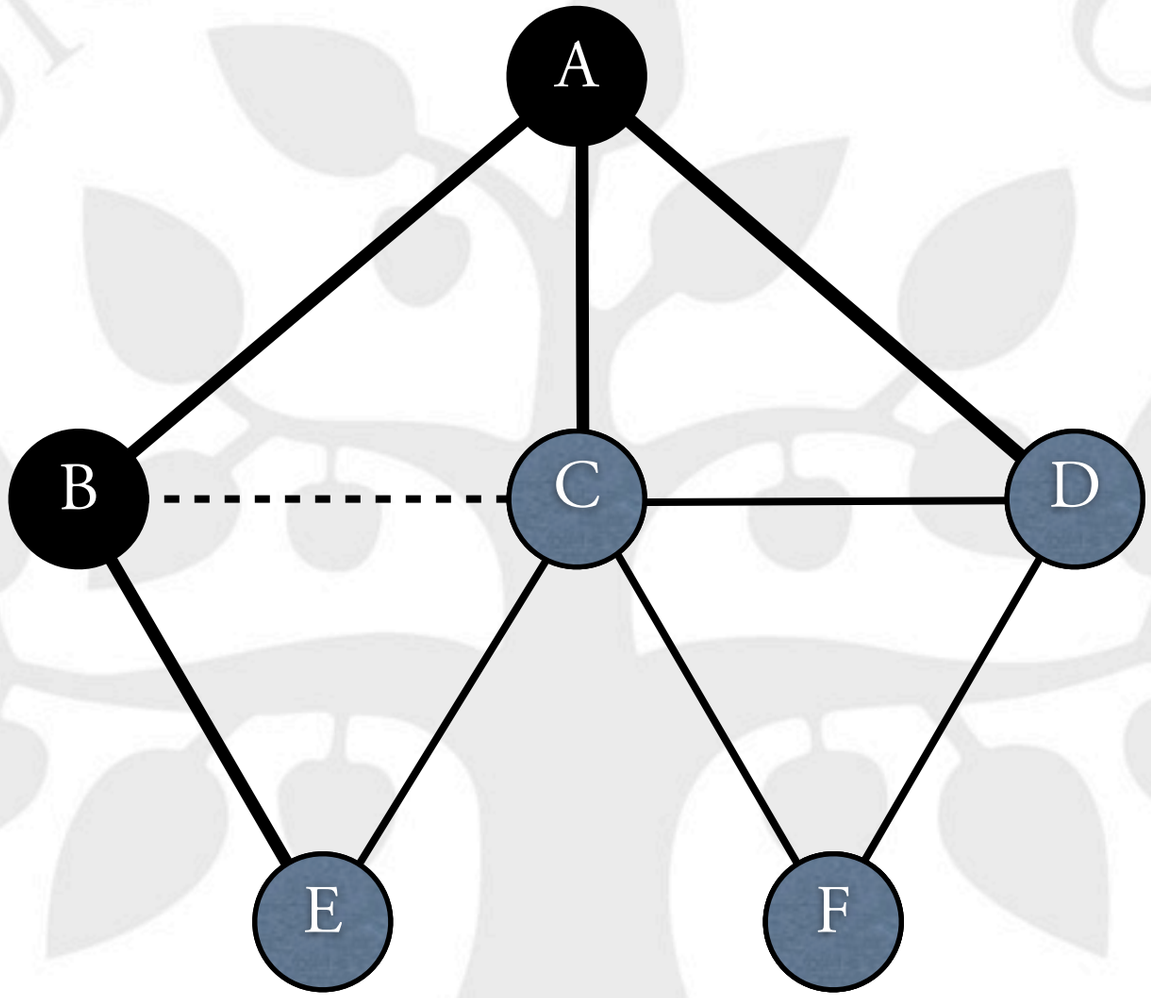
C D

Eksempel

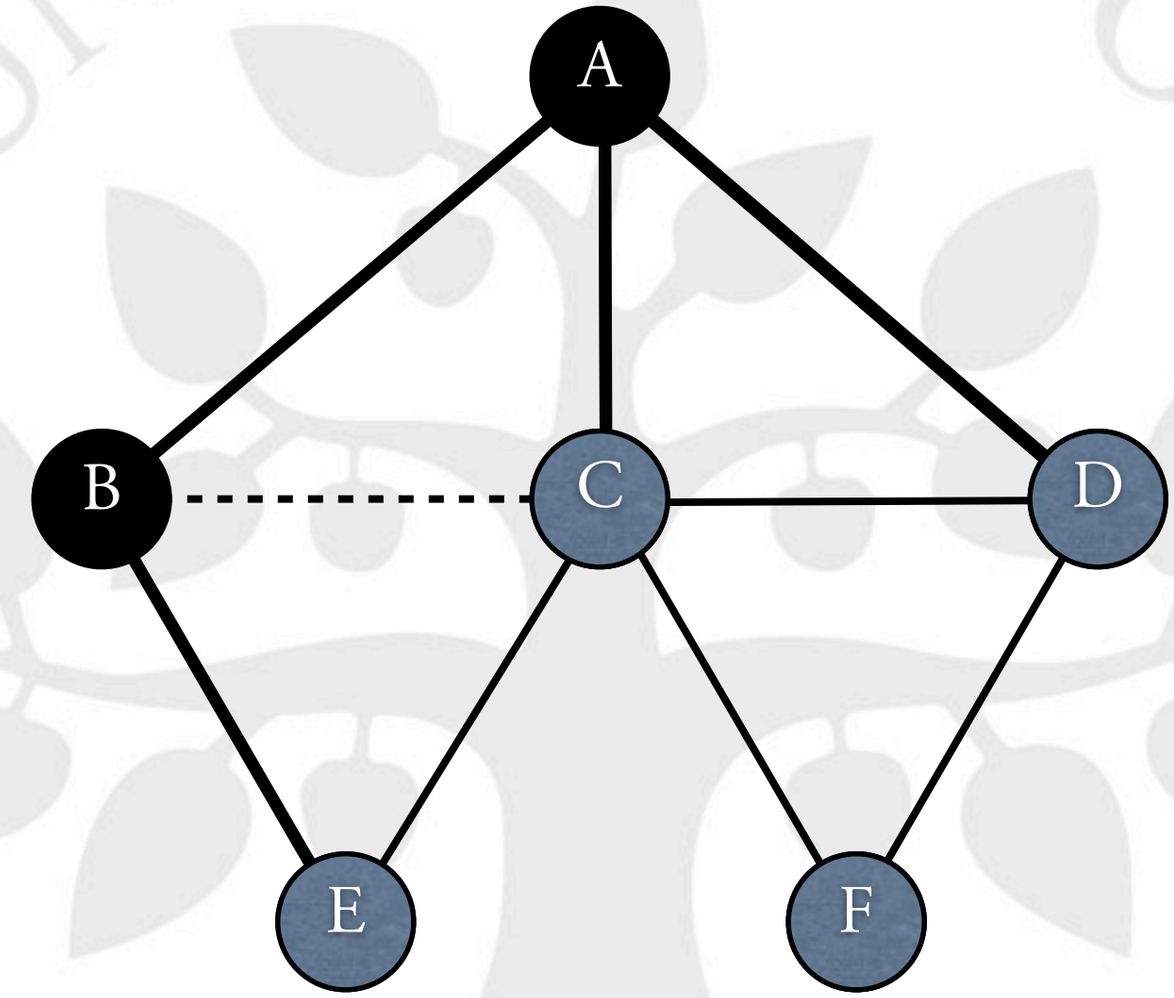


C D E

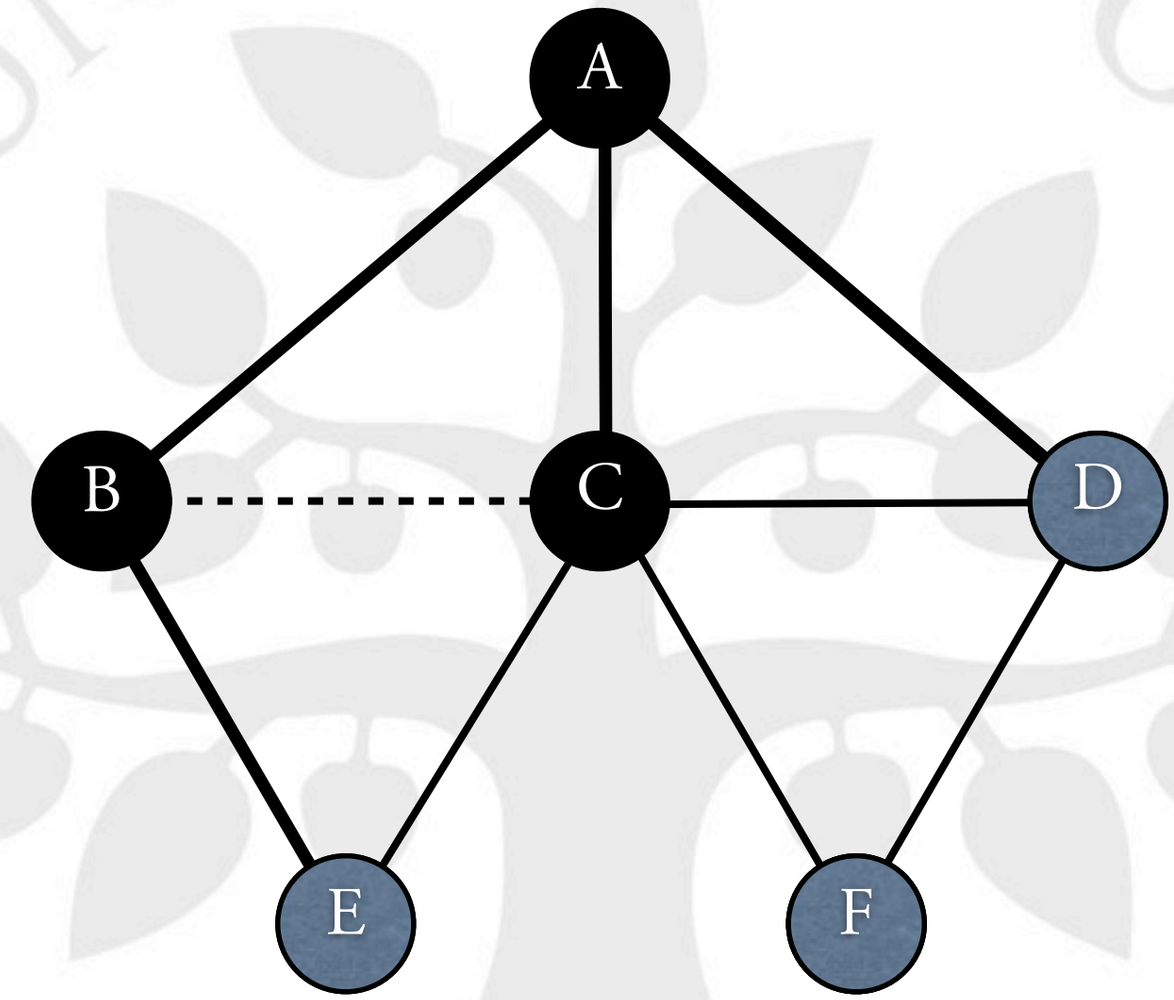
Eksempel



Eksempel

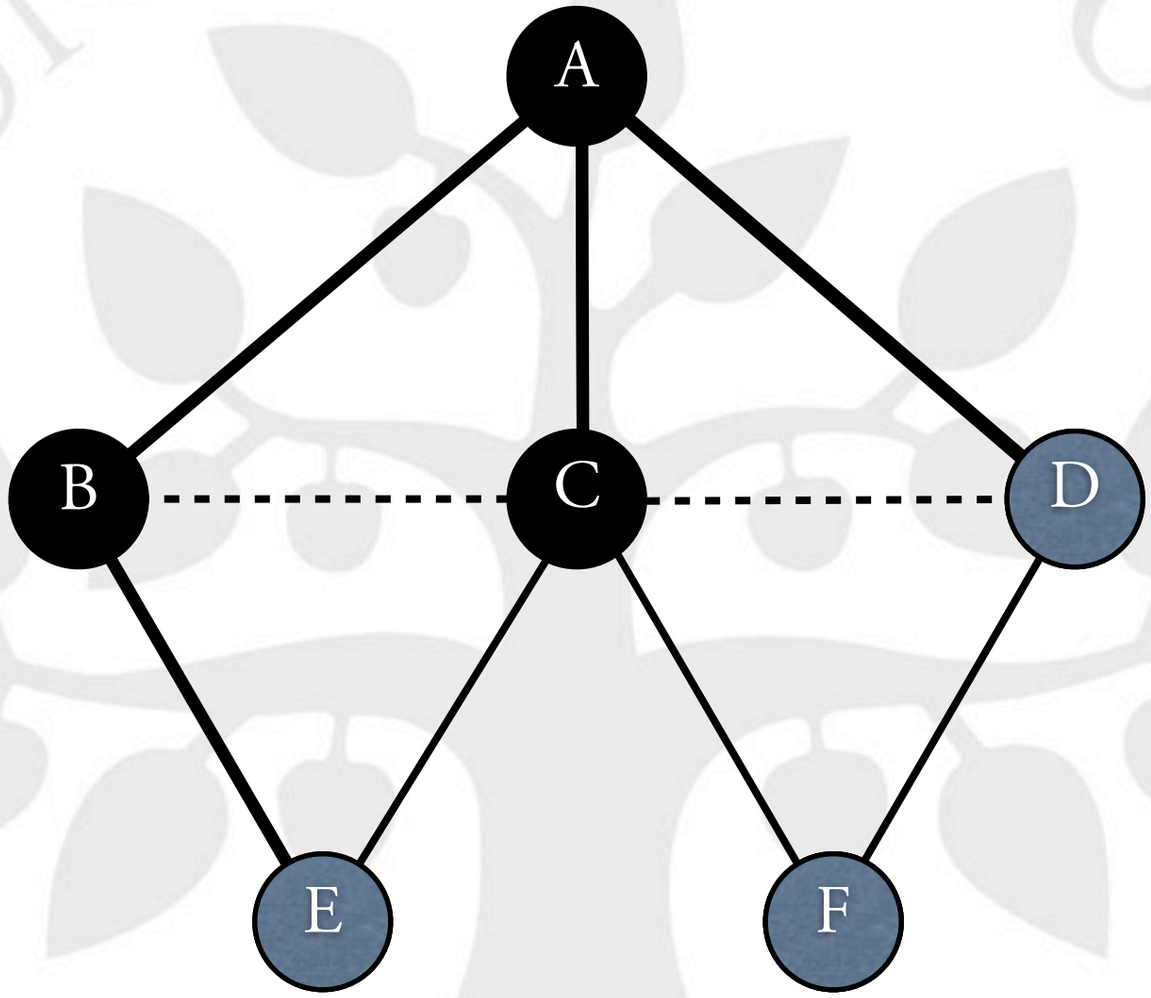


Eksempel

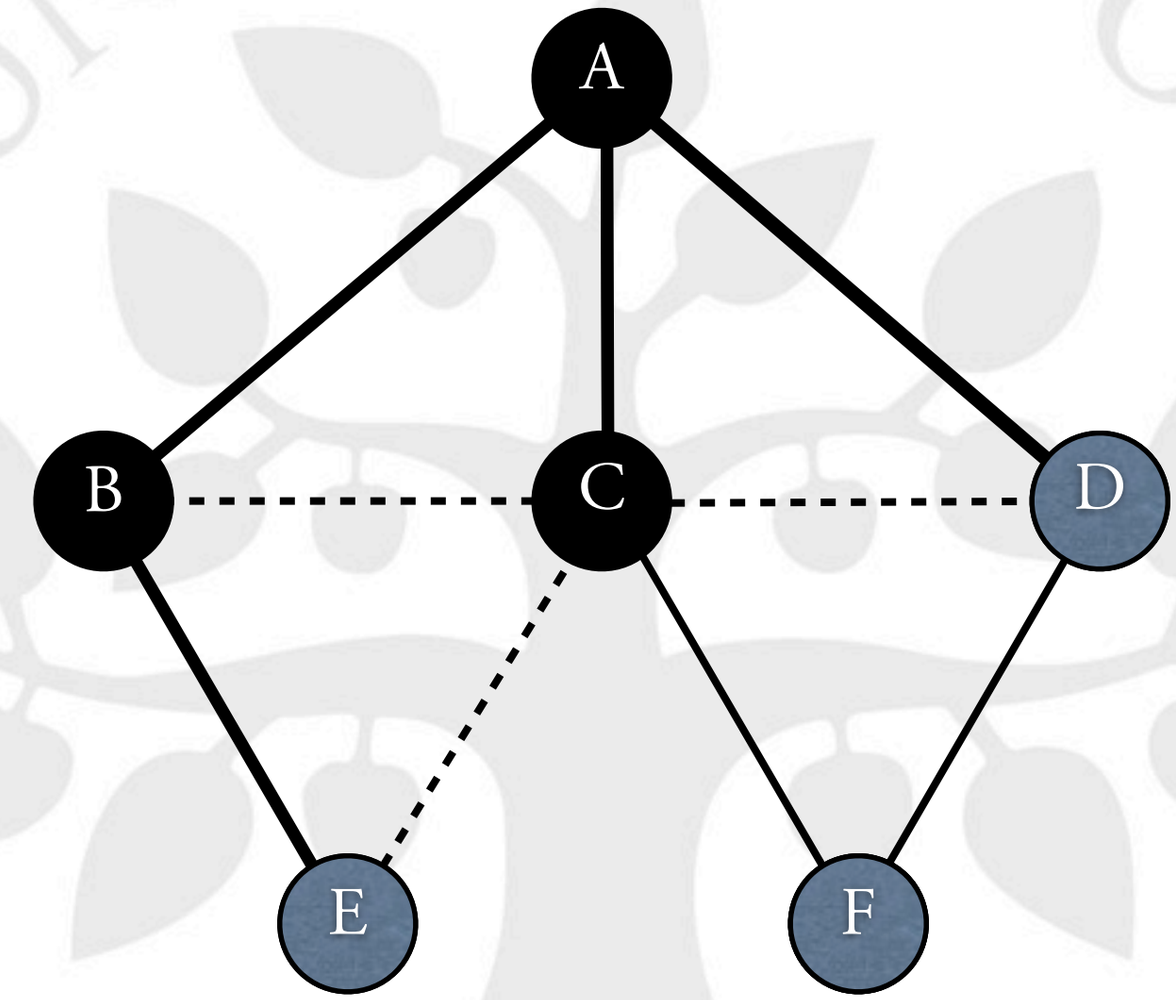


D E

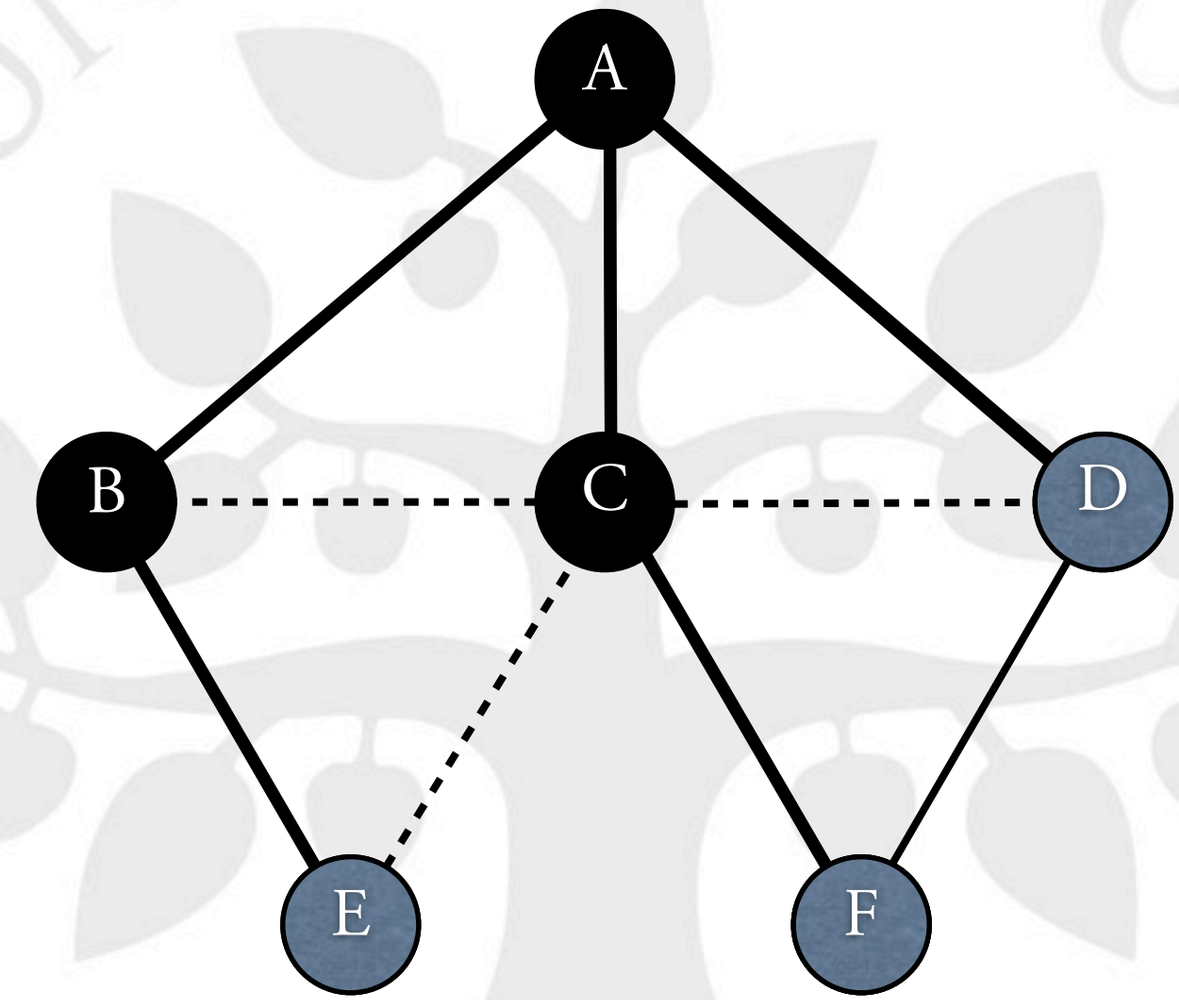
Eksempel



Eksempel

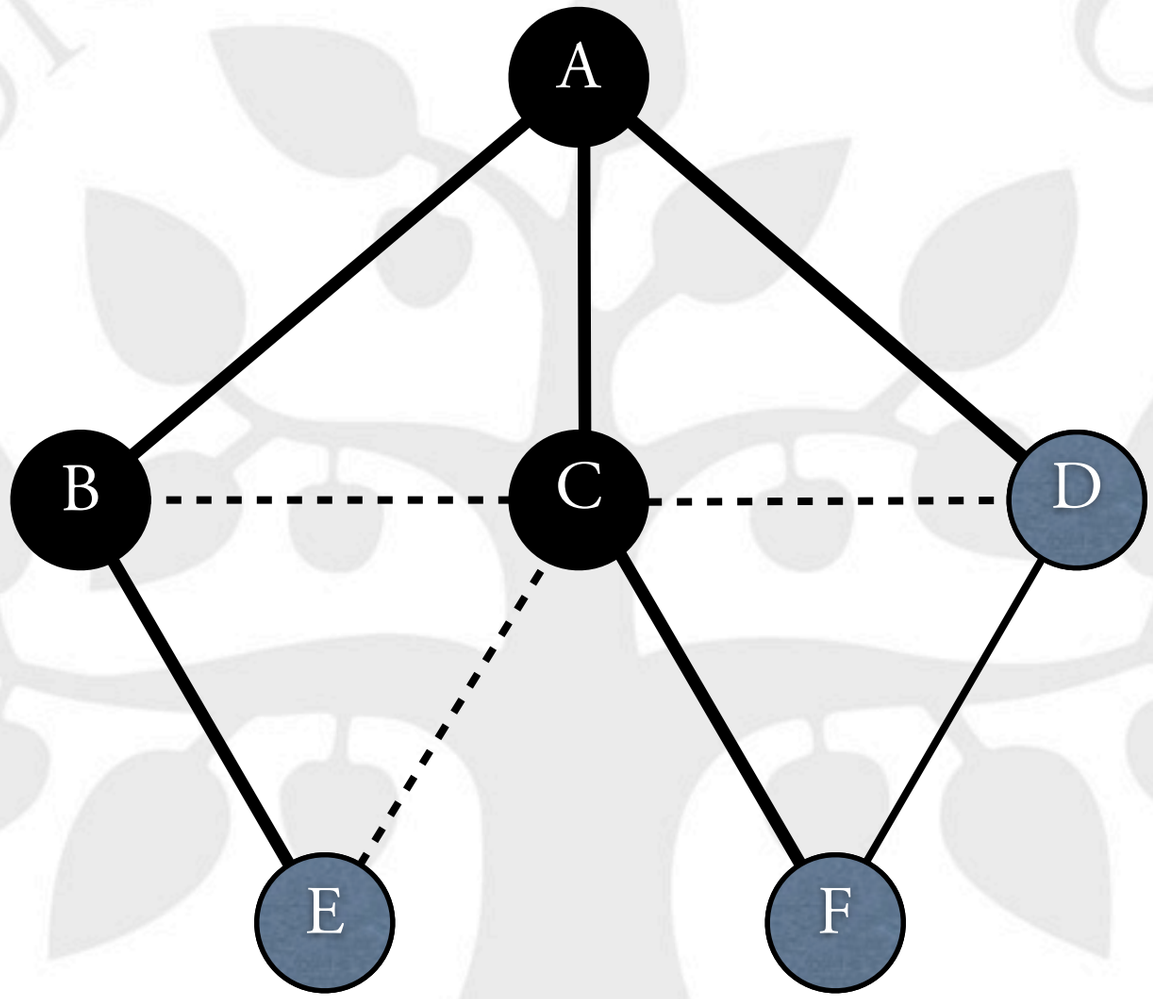


Eksempel

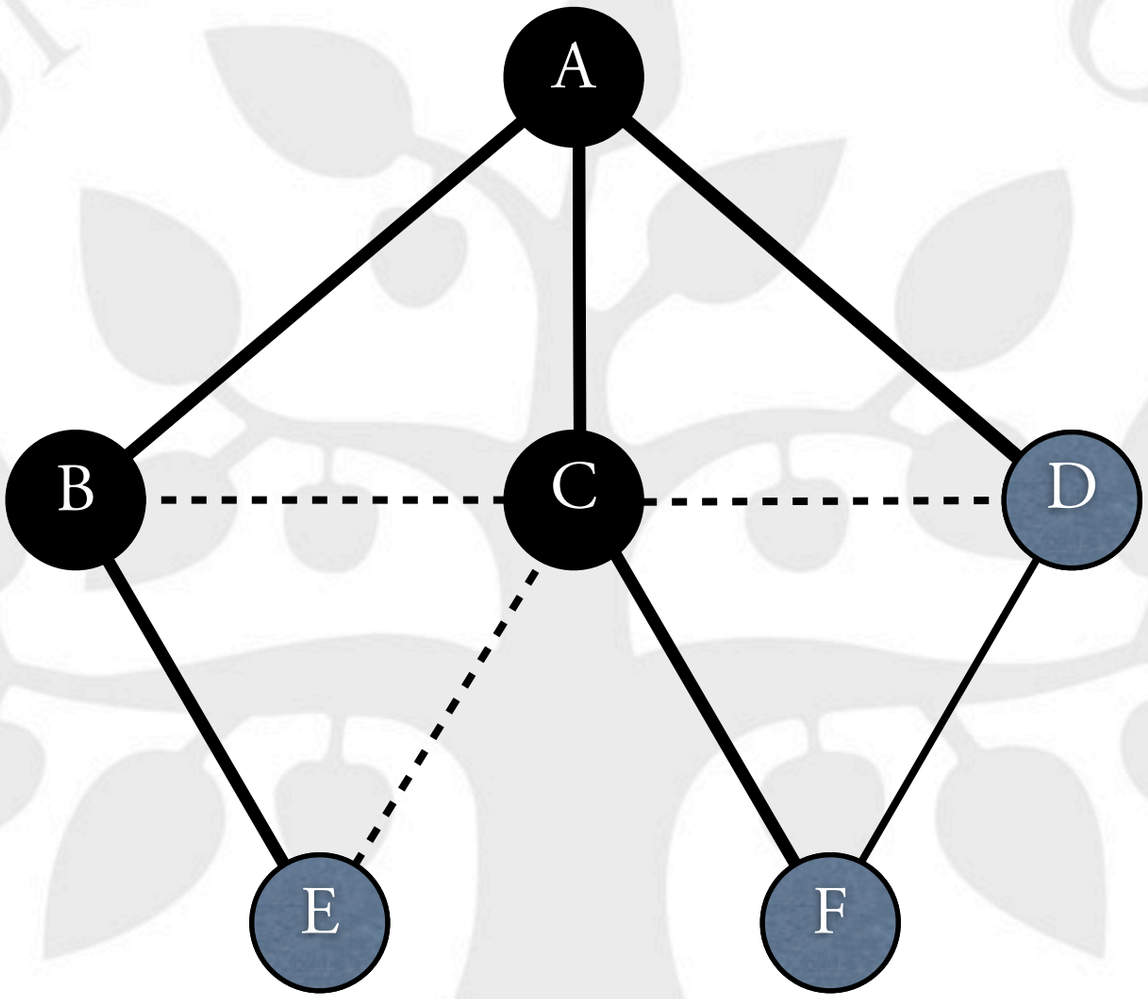


D E F

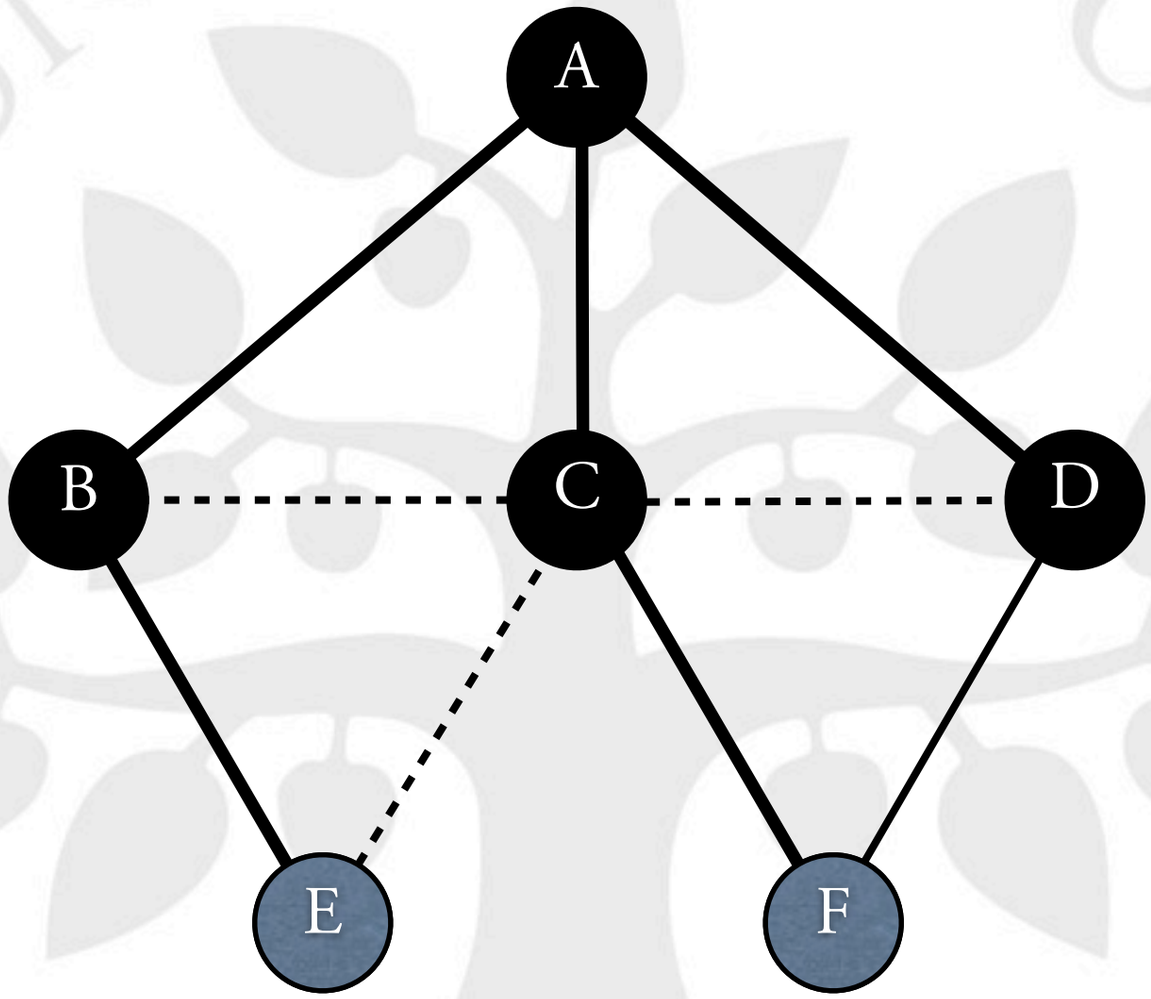
Eksempel



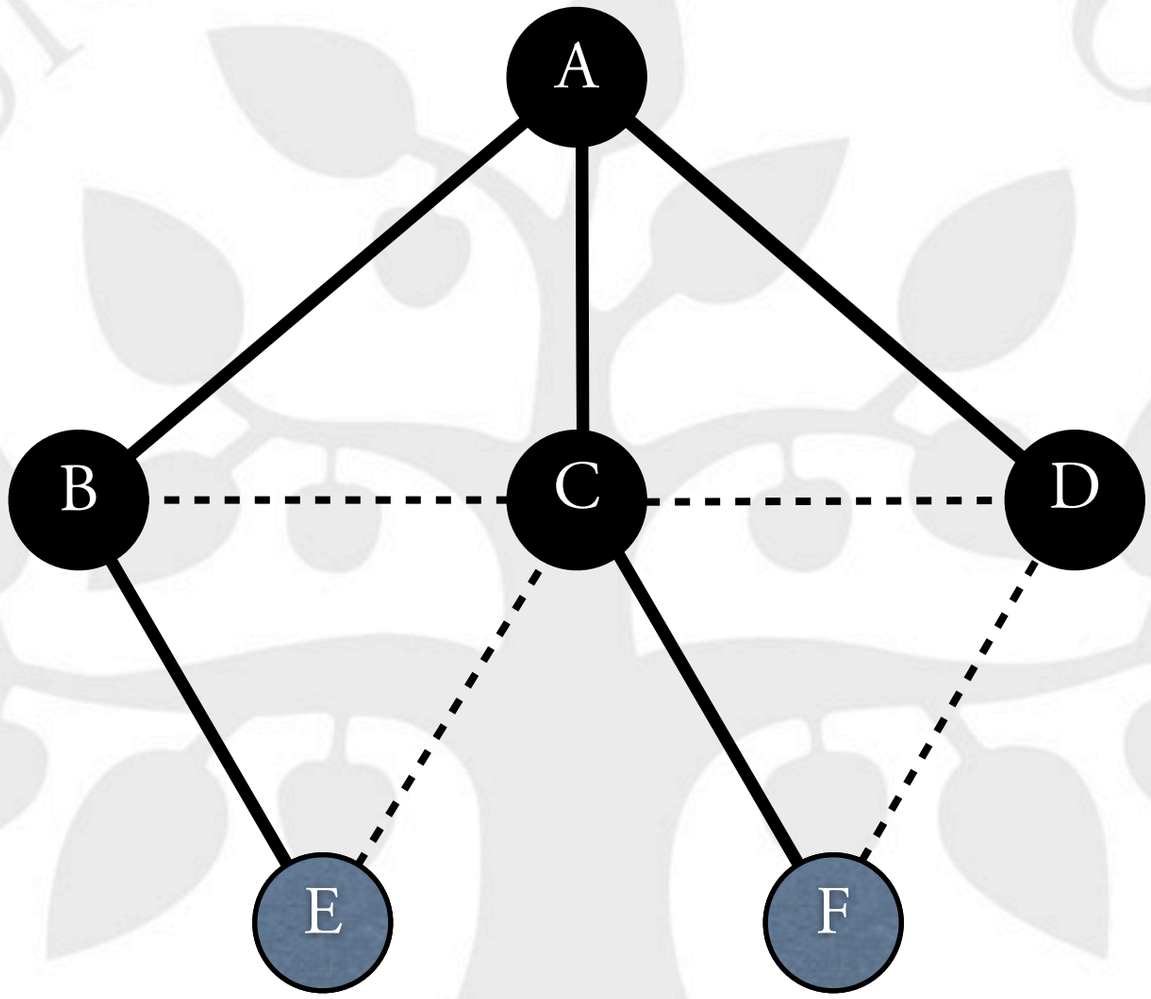
Eksempel



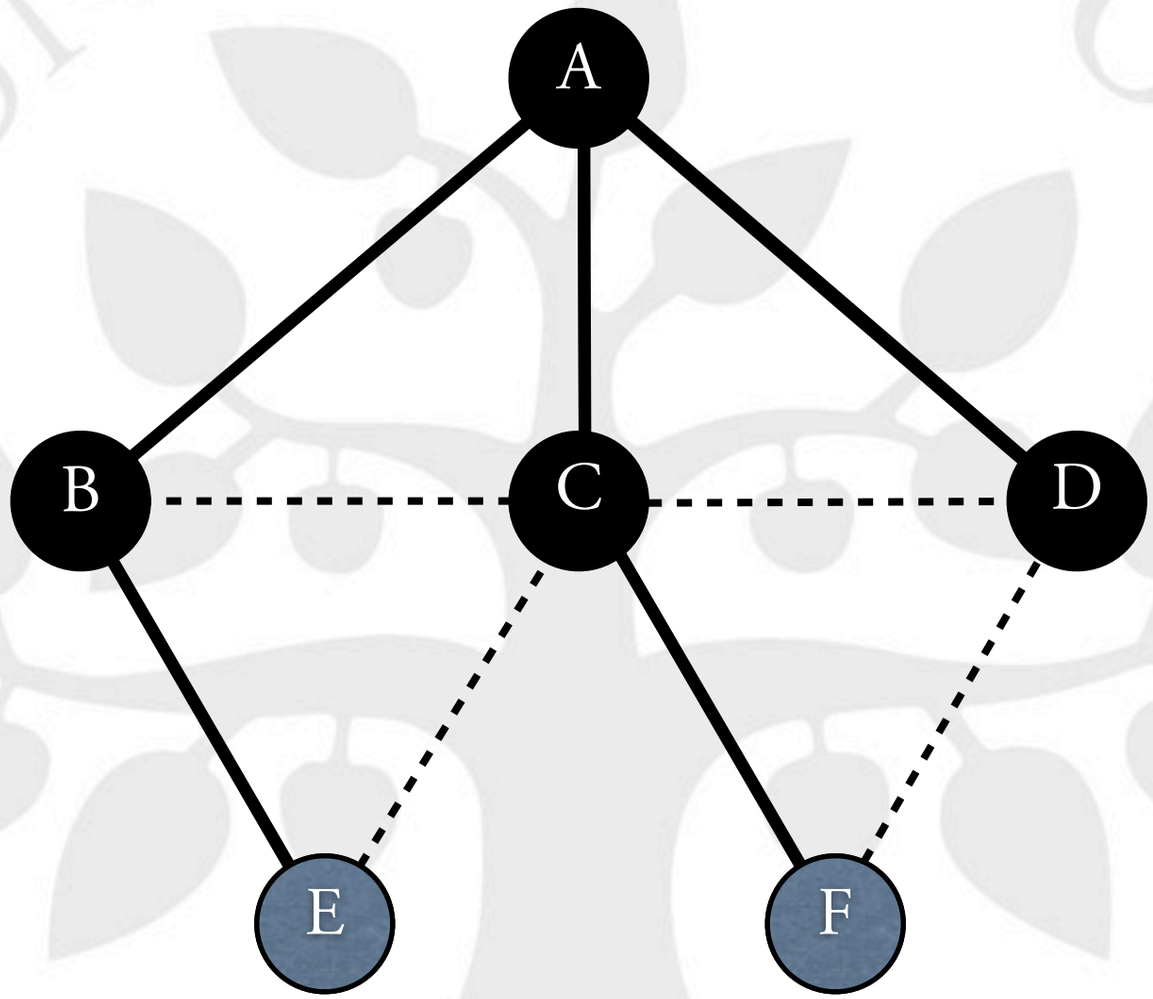
Eksempel



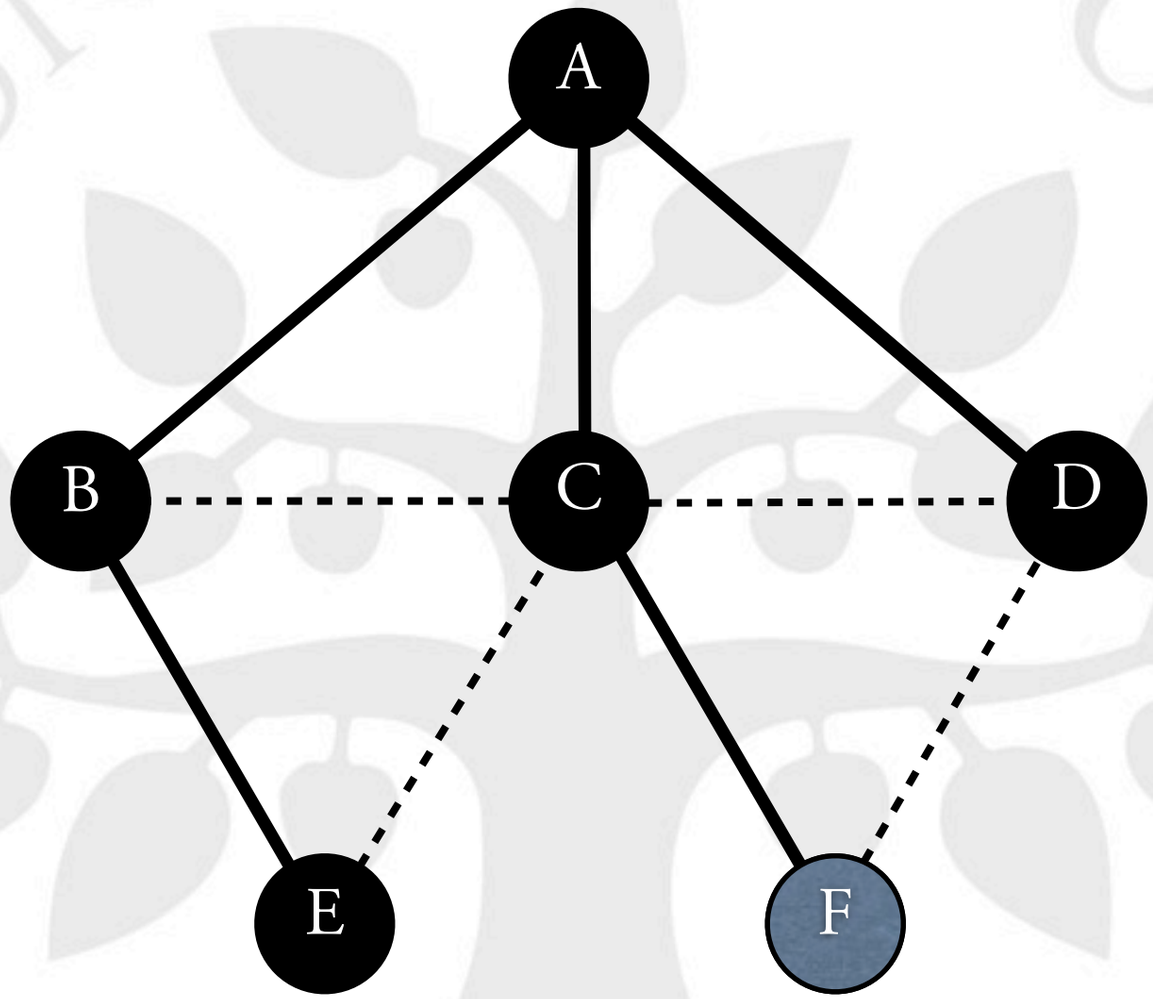
Eksempel



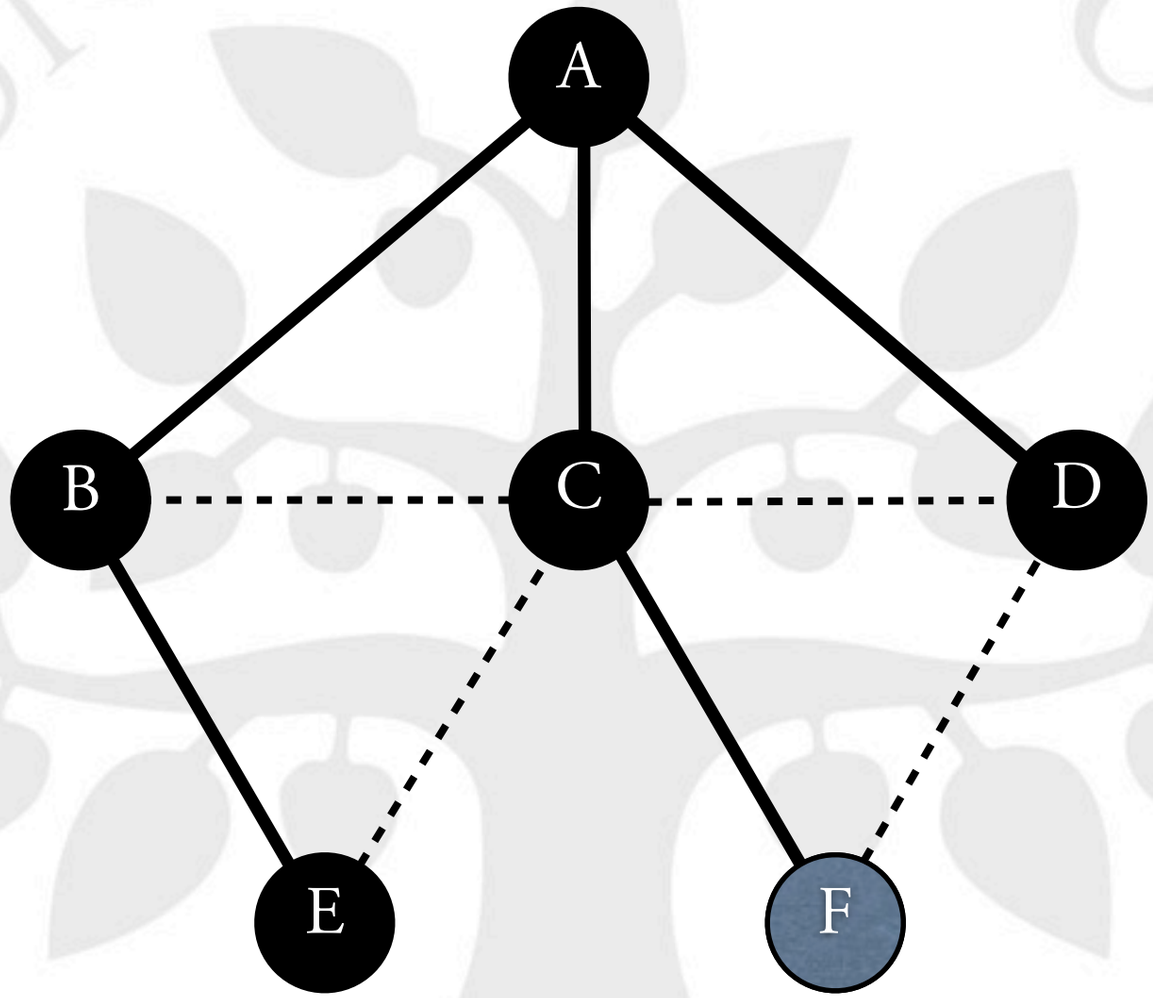
Eksempel



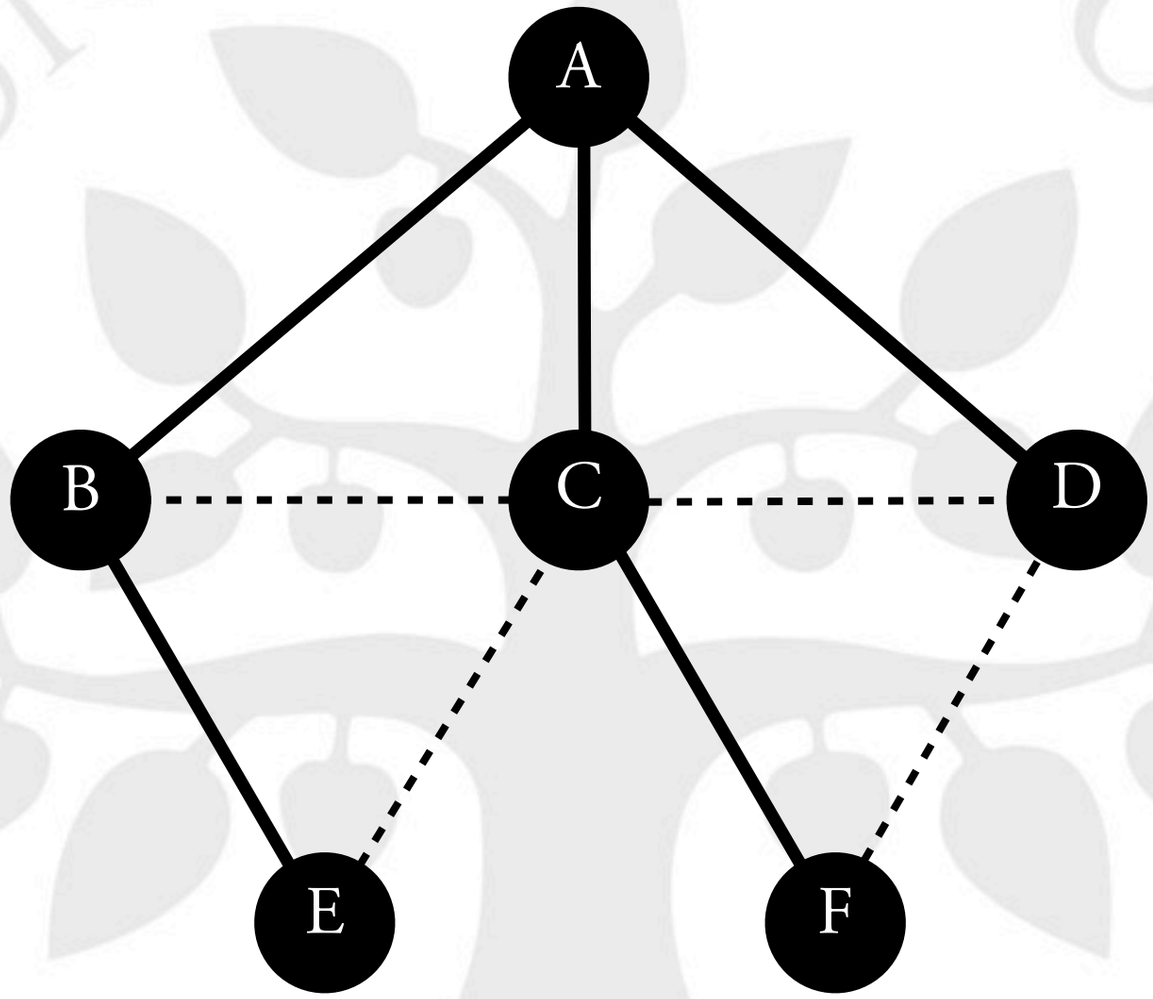
Eksempel



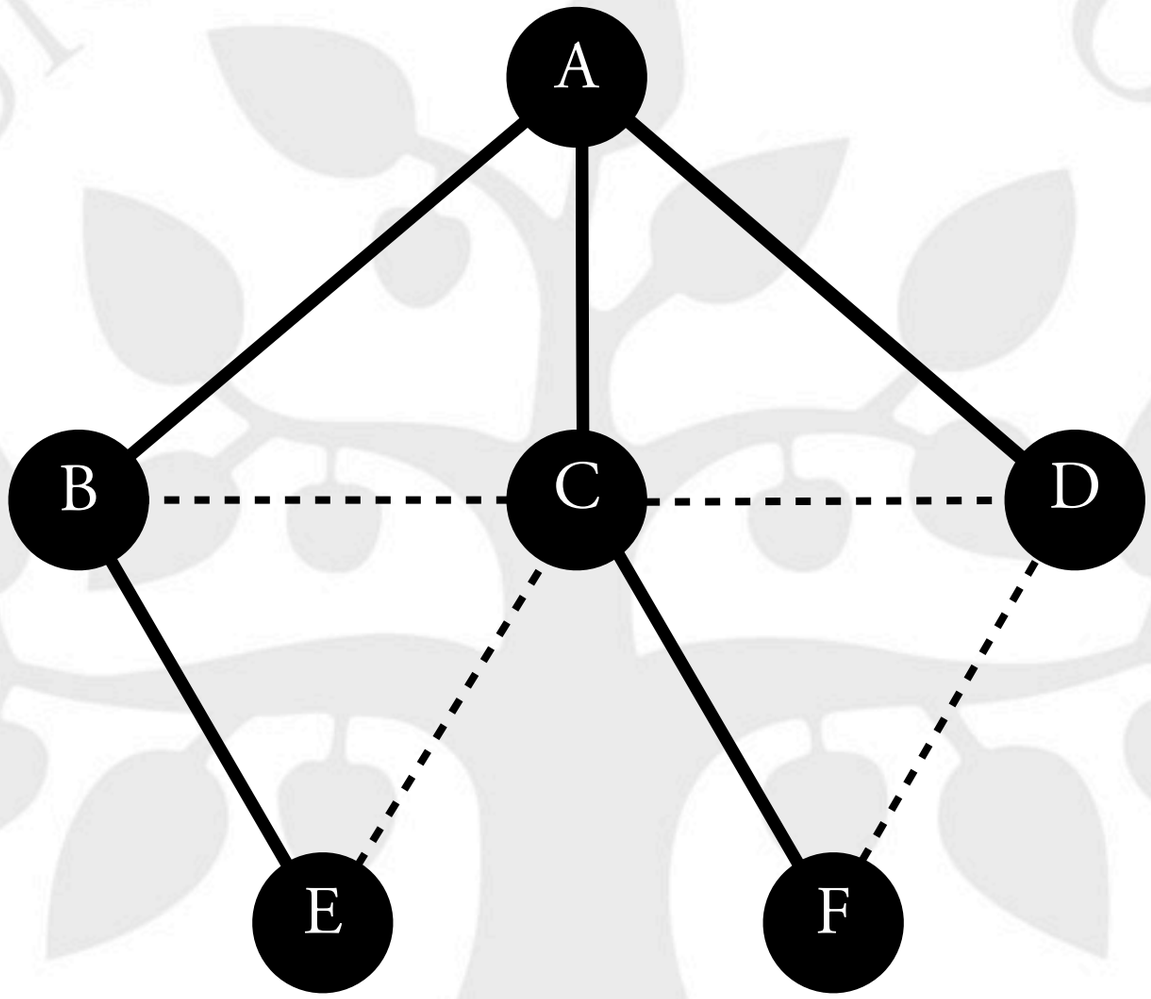
Eksempel



Eksempel

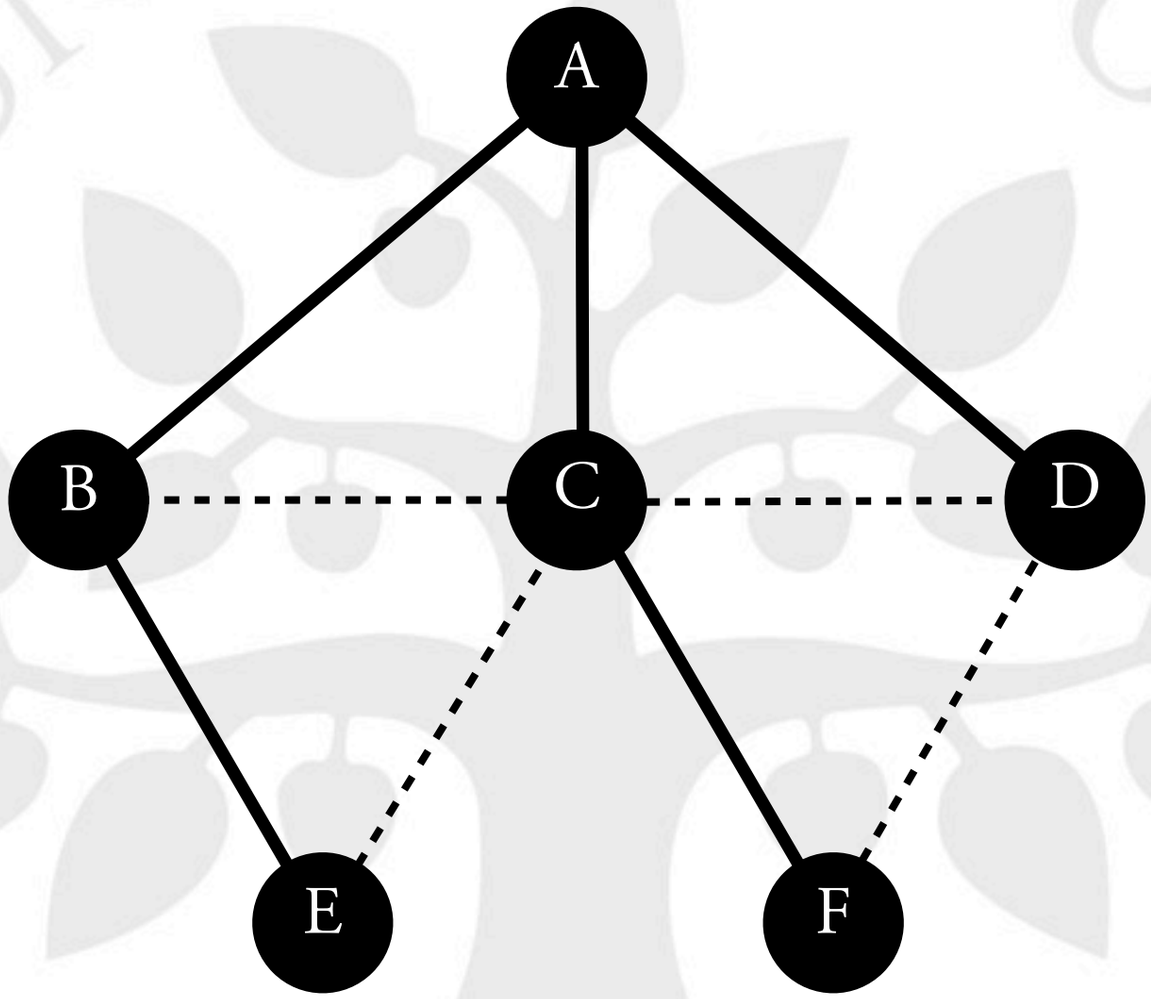


Eksempel



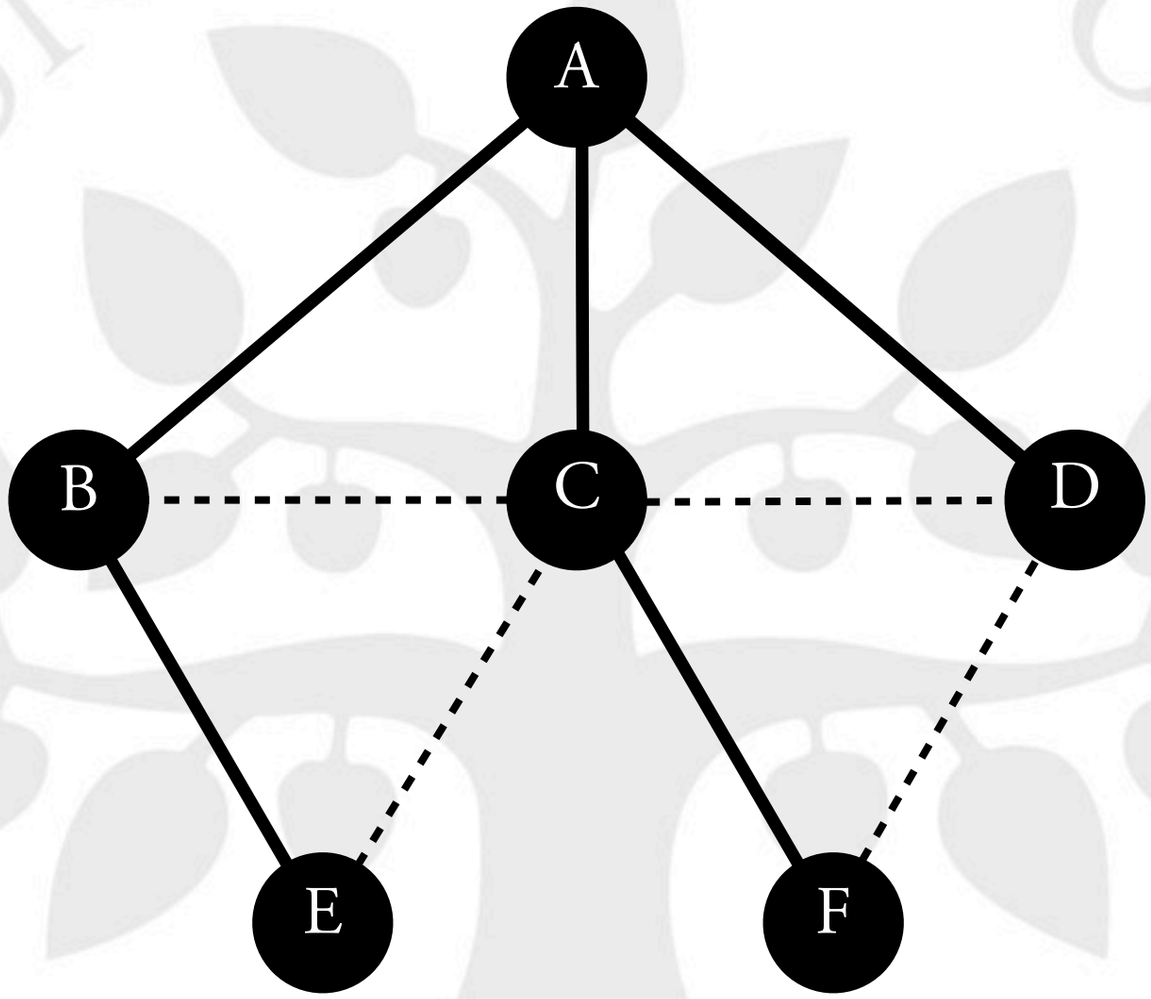
- Hvordan kan BFS udvides til at

Eksempel



- Hvordan kan BFS udvides til at
  - finde en kreds i grafen?

Eksempel



- Hvordan kan BFS udvides til at
  - finde en kreds i grafen?
  - beregne den korteste sti mellem to givne knuder?

BFS's egenskaber

BFS's egenskaber

- Egenskab 1

BFS's egenskaber

- Egenskab 1
  - BFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af

BFS's egenskaber

- Egenskab 1
  - BFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af
- Egenskab 2

BFS's egenskaber

- Egenskab 1
  - BFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af
- Egenskab 2
  - De kanter BFS(G, v) følger danner et udspændende træ T_v af den sammenhængskomponent v er en del af

BFS's egenskaber

- Egenskab 1
  - BFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af
- Egenskab 2
  - De kanter BFS(G, v) følger danner et udspændende træ T_v af den sammenhængskomponent v er en del af
- Egenskab 3

BFS's egenskaber

- Egenskab 1
  - BFS(G, v) besøger alle knuder og kanter i den sammenhængskomponent v er en del af
- Egenskab 2
  - De kanter BFS(G, v) følger danner et udspændende træ T_v af den sammenhængskomponent v er en del af
- Egenskab 3
  - For enhver knude w i G gælder at enhver sti fra v til w i G er mindst lige så lang som stien fra v til w i T_v.

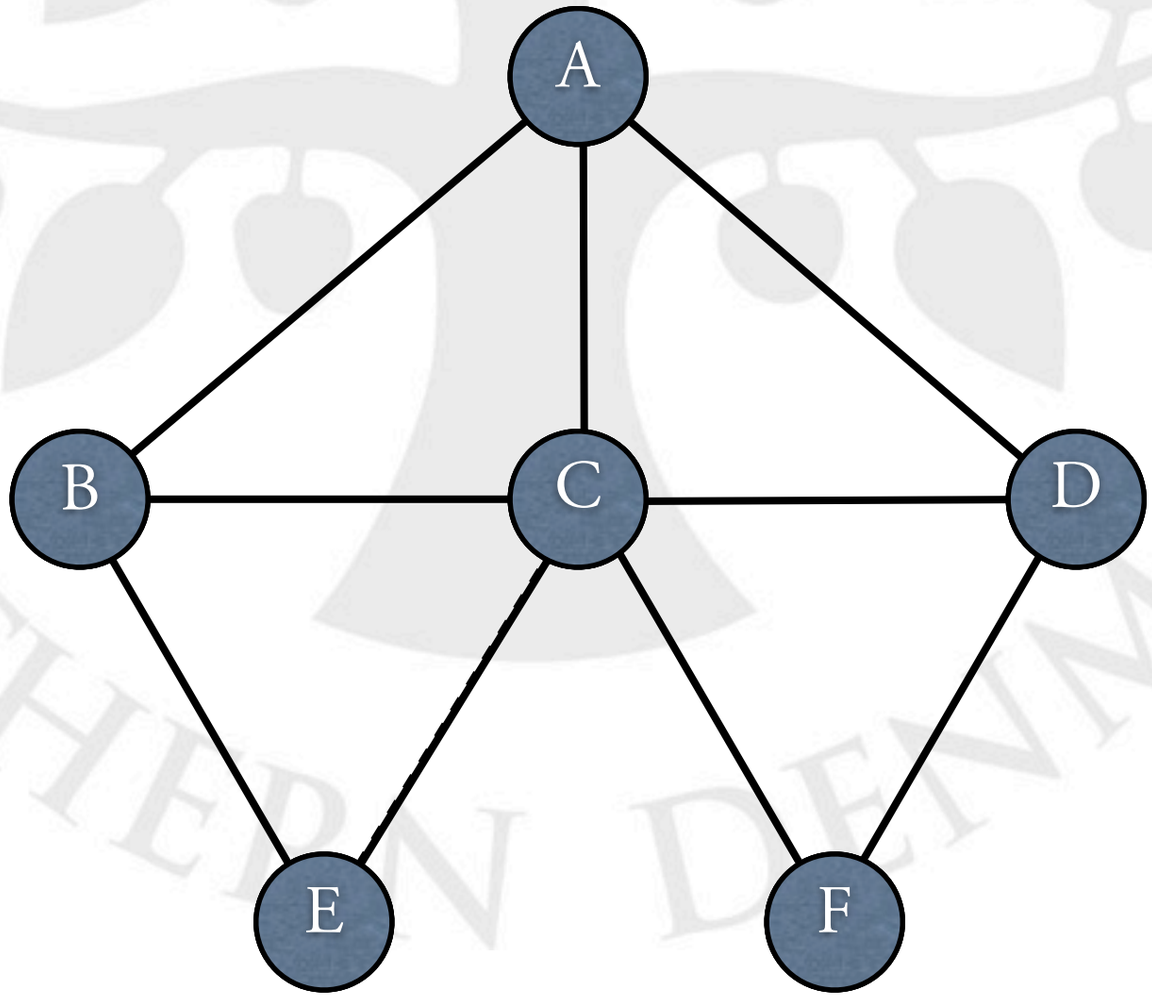
BFS.java

```
import java.util.*;

public class BFS {
    public static void main( String[] args ) {
        Node a = new Node( "A" );
        Node b = new Node( "B" );
        Node c = new Node( "C" );
        Node d = new Node( "D" );
        Node e = new Node( "E" );
        Node f = new Node( "F" );

        a.addNeighbour( b );
        a.addNeighbour( c );
        a.addNeighbour( d );

        b.addNeighbour( a );
        b.addNeighbour( c );
        b.addNeighbour( e );

        c.addNeighbour( a );
        c.addNeighbour( b );
        c.addNeighbour( d );
        c.addNeighbour( e );
        c.addNeighbour( f );

        d.addNeighbour( a );
        d.addNeighbour( c );
        d.addNeighbour( f );
```

BFS.java (cont.)

```
        e.addNeighbour( b );
        e.addNeighbour( c );

        f.addNeighbour( c );
        f.addNeighbour( d );

        BFS( a );
    }

    public static void BFS( Node v ) {
        ArrayDeque<Node> q = new ArrayDeque<Node>();
        v.setMark( true );
        q.add( v );
        while( !q.isEmpty() ) {
            Node u = q.remove();
            System.out.println( u + ": Visting" );
            for( Node w : u.getNeighbours() ) {
                if( !w.getMark() ) {
                    System.out.println( u + ": Adding "+w+" to the queue" );
                    w.setMark( true );
                    q.add( w );
                }
            }
        }
    }
}
```

Output



```
Terminal — bash — 66×14
MAC-SDU-00001:7 petersk$ javac BFS.java
MAC-SDU-00001:7 petersk$ java BFS
A: Visting
A: Adding B to the queue
A: Adding C to the queue
A: Adding D to the queue
B: Visting
B: Adding E to the queue
C: Visting
C: Adding F to the queue
D: Visting
E: Visting
F: Visting
MAC-SDU-00001:7 petersk$
```