

# MAT1030 – Diskret Matematikk

## Forelesning 22: Grafteori

Dag Normann

Matematisk Institutt, Universitetet i Oslo

14. april 2010

(Sist oppdatert: 2010-04-14 12:43)



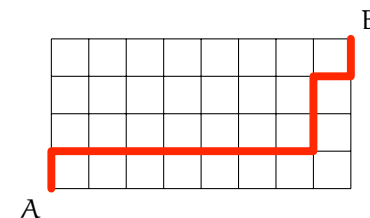
# Kombinatorikk

## Oppsummering av regneprinsipper

- Ordnet utvalg med repetisjon:  $n^r$
- Ordnet utvalg uten repetisjon:  ${}^n P_r$
- Permutasjoner:  $n!$
- Kombinasjoner:  $\binom{n}{k}$
- Vi skal se på noen flere eksempler.

## Eksempel

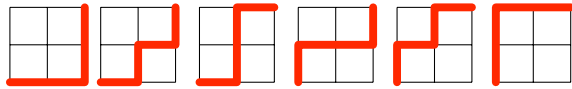
På hvor mange forskjellige måter kan vi gå fra A til B i dette 8 ganger 4-rutenettet? Vi må gå ett steg av gangen og kun oppover eller til høyre.



- Hvor mange slike stier er det?
- Denne stien kan representeres som  $\uparrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \uparrow \uparrow \uparrow$ .
- Dette er et ord på 12 tegn over alfabetet  $\{\uparrow, \rightarrow\}$  hvor fire av tegnene er  $\uparrow$  og åtte av tegnene er  $\rightarrow$ .
- Hvor mange slike ord er det? Det er  $\binom{12}{4} = \frac{12 \cdot 11 \cdot 10 \cdot 9}{4 \cdot 3 \cdot 2 \cdot 1} = 495$

## Eksempel

- Vi sjekker at det stemmer for 2 ganger 2-rutenettet:



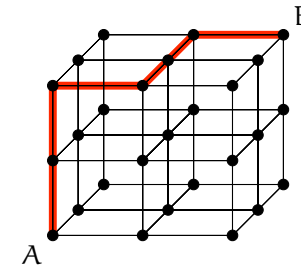
- Antall ord blir i dette tilfellet:

$$\binom{4}{2} = \frac{4 \cdot 3}{2 \cdot 1} = 6$$

som stemmer...

## Oppgave

På hvor mange forskjellige måter kan vi gå fra A til B i denne  $2 \times 2 \times 2$ -kuben? Vi må gå ett steg av gangen og kun oppover, til høyre eller innover?



## Eksempel

- (Dette ble også nevnt på forrige forelesning i et introduksjonseksempel om kuler og bokser.)
- På hvor mange ulike måter kan de 6 karakterene A til F gis til 10 studenter?
- Vi ikke er interessert i hvilken karakter en bestemt student får, men kun antallet av hver karakter i fordelingen.
- Det er 6 muligheter for hver av de 10 studentene.
- Kan vi bruke multiplikasjonsprinsippet og si at svaret er  $6^{10}$ ? **Nei**
- La oss lage 6 båser og putte studentene i hver sin bås avhengig av hvilken karakter de får.

## Eksempel

- 2 studenter per karakter, og ingen stryk kan representeres slik:



- Fordelingen 123211 kan representeres slik:



- At alle stryker kan representeres slik:



- Hvor mange slike kombinasjoner fins det?
- Av 15 tegn må 5 være røde streker.
- Antall muligheter må være

$$\binom{15}{5} = \frac{15 \cdot 14 \cdot 13 \cdot 12 \cdot 11}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 3003$$

## Eksempel

### Eksempel

- Mer generelt har vi

$$\binom{n+k-1}{k-1}$$

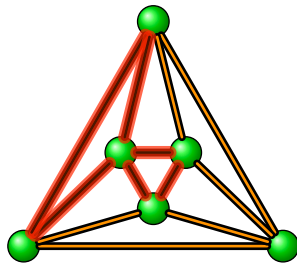
som gir hvor mange forskjellige måter vi kan fordele  $n$  identiske elementer i  $k$  forskjellige beholdere på.

- Dette kalles også for uordnet utvalg *med repetisjon*.

## Grafteori

## Grafteori

- I går ga vi en smaksprøve på *grafteori*.
- Vi sa at en *graf* består av *noder* og *kanter*:



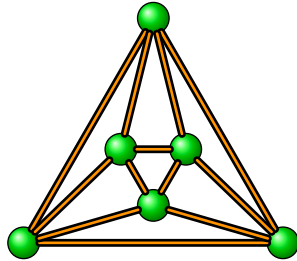
- Vi ga følgende oppgave: Klarer dere å tegne denne på et ark uten å løfte blyanten og uten å gå over en kant to ganger?

## Grafteori

- Grafer fins overalt rundt oss!
- Grafteori er en viktig del av både *anvendt* og *teoretisk* matematikk.
- Grafteori brukes til å modellere problemer innenfor mange områder: informatikk, biologi, samfunnsvitenskap, lingvistikk, fysikk ...
- Mange problemer kan *representeres* ved å bruke grafer.
- Vi har møtt idéen om *representasjon* flere ganger allerede.
- Vi representerer f.eks. reelle tall ved hjelp av binære tall.
- Vi kan representere et matematisk problem som et annet som er enklere å løse.
- En representasjon gjør at vi kan se bort fra det som er irrelevant. Vi fanger inn *essensen*.
- Det er akkurat det som skjer i grafteori.

## En graf

- En *graf* består av *nod*er (●) og *kanter* (—).
- Vi har til nå sett et eksempel på en graf:



- Klarte dere å tegne denne på et ark uten å løfte blyanten og uten å gå over en kant to ganger?
- Etter disse forelesningene i grafteori skal alle klare å besvare dette spørsmålet umiddelbart.
- Vi skal se at oppgaven er ekvivalent med å finne en såkalt *Eulersti*.

## Søkealgoritmer for grafer

- Søkealgoritmer for grafer er et viktig tema.
- Kjente søkealgoritmer for grafer er f.eks.:
  - *Prims algoritme* for å finne et *minimalt spenn*tre i en vektet graf. (Kruskals algoritme for samme problem.)
  - *Dijkstras algoritme* for å finne *minste avstand*, eller *korteste sti*, i en vektet graf.
- Vi kan søke bredde først eller dybde først
- For veldig mange grafproblemer har man ikke funnet *effektive* algoritmer.

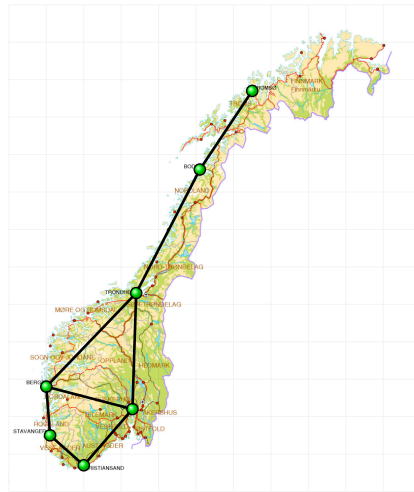
## Grafteori - noen eksempler

## Eksempler på grafer

Grafer kan representere mange forskjellige ting.

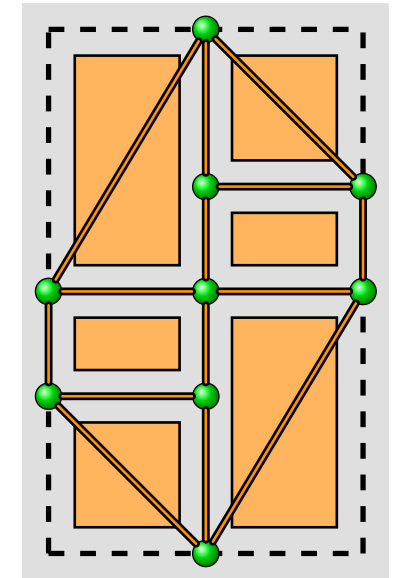
- Nodene kan være studentene på Blindern, og kantene kan representere at to kjenner hverandre.
- Nodene kan være tilstandene som et dataprogram er i, og kantene kan representere overganger mellom tilstandene.
- En graf kan representere lenkestrukturen på et nettsted, hvor nodene er nettsider og kantene er lenker.
- Listen fortsetter: elektroniske kretser, molekyler i kjemi, datanettverk, analyse av nettverkstrafikk. . .
- Et *tre* er en spesiell type graf. Vi kommer til trær i kapittel 11.
- Vi skal se på flere eksempler på grafer før vi begynner med teorien.

- Et kart kan gi utgangspunkt for flere forskjellige grafer.
- En mulighet er at
  - nodene representerer *byer*
  - kantene representerer *veier*
- En annen mulighet er at
  - nodene representerer *områder*, f.eks. fylker
  - kantene representerer *grenser*
- Når vi har representasjonen, så kan vi egentlig glemme det opprinnelige kartet.



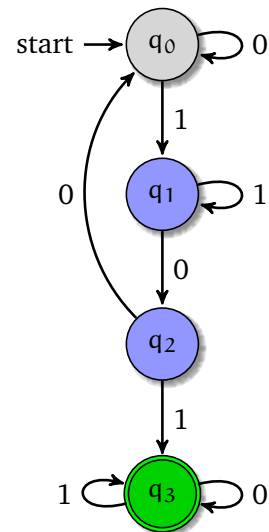
## Veinett som grafer

- Et veinett kan representeres som en graf.
- Vi kan la hvert *kryss* svare til en node.
- Vi kan la *veiene* som forbinder kryssene svare til kantene.
- Når vi har tegnet opp grafen, så kan vi resonnerer om den i stedet for om selve veinettet.



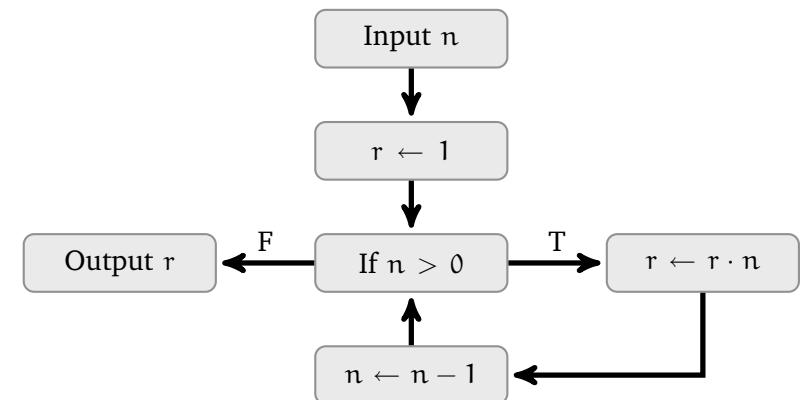
## Endelige tilstandsmaskiner som grafer

- Vi kan tenke på *endelige tilstandsmaskiner* som grafer.
- Her kalles  $q_0$ ,  $q_1$ ,  $q_2$  og  $q_3$  *tilstander* og overgangene kalles *transisjoner*.
- Vi har en starttilstand,  $q_0$  og en såkalt *aksepterende* tilstand,  $q_3$ .
- Hvis vi begynner med tallet 1101 og følger kantene etter hvert som vi leser siffer, så ender vi opp i  $q_3$ . Siden det er en aksepterende tilstand, er 1101 *akseptert*.
- Tallet 100 aksepteres ikke, siden vi ender opp i tilstand  $q_0$ , som ikke er aksepterende.
- Ser du hvilke tall som aksepteres?



## Flyttdiagrammer som grafer

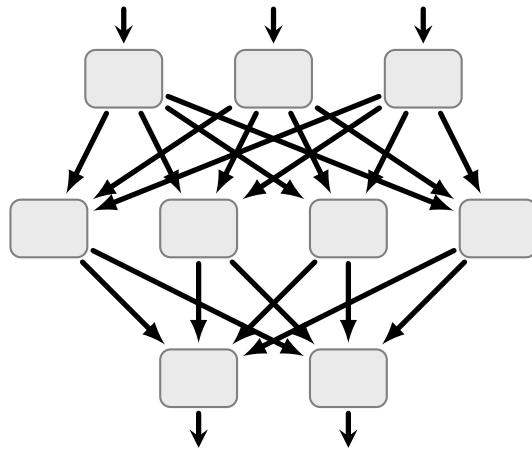
- Vi kan tenke på *flyttdiagrammer* som grafer.
- I følgende eksempel representerer nodene programinstruksjoner.



- Hvilket program er dette?

## Flytdiagrammer og multiplikasjonsprinsippet

- Multiplikasjonsprinsippet igjen.





## Grafteori - definisjoner og begreper

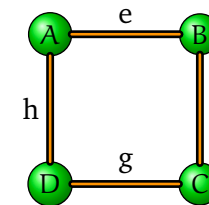
## Grafteori - definisjoner og begreper

### Definisjon (Graf)

En *graf*  $G$  består av en ikke-tom mengde *noder*  $V$  og en mengde *kanter*  $E$ , slik at enhver kant forbinder nøyaktig to noder med hverandre eller en node med seg selv.

- Dette er med vilje litt upresist.
- Vi presiser heller etter hvert når vi trenger det.
- På engelsk brukes begrepene
  - *vertex/vertices* om noder, og
  - *edges* om kanter.
- Vi tegner noder slik: 
- og kanter slik: 
- Det er ikke viktig akkurat *hvordan* vi tegner grafer; det er *strukturen* i graf som er viktig, hvilke noder som er forbundet med hvilke via en kant.

## Grafteori - definisjoner og begreper



Her er A, B, C og D noder, mens e, f, g og h er kanter.

### Definisjon (Inntil/naboer)

En kant ligger *inntil* (engelsk: *incident*) nodene som forbindes av den. To noder er *naboer* (engelsk: *adjacent*) hvis de forbindes av en kant.

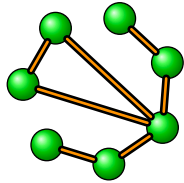
- Kanten e ligger inntil nodene A og B.
- Nodene B og C er naboer, siden de forbindes av kanten f.

## Sammenhengende grafer

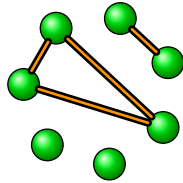
En graf trenger ikke å være *sammenhengende*.

### Definisjon (Sammenhengende)

En graf er *sammenhengende* (engelsk: *connected*) hvis det er mulig å komme fra enhver node til enhver annen node ved å følge kantene.



En *sammenhengende* graf.



En *usammenhengende* graf.

## Tomme grafer og løkker

En graf trenger ikke å ha noen kanter, men den må ha minst én node. Grafer uten kanter kalles *nullgrafer* eller *tomme grafer* (engelsk: *null graph*).



En tom graf.

En graf kan ha *løkker* (engelsk: *loop*), en kant som går fra en node til den samme noden.



En graf med en løkke.

## Parallele kanter og enkle grafer

En graf kan ha *parallelle* kanter, to eller flere kanter som forbinder de samme to nodene.



En graf med parallelle kanter.

### Definisjon (Enkel)

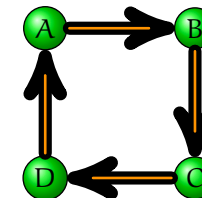
En graf er *enkel* (engelsk: *simple*) hvis den ikke har løkker eller parallelle kanter.

- Det er ganske vanlig å definere grafer slik at løkker og parallelle kanter ikke forekommer.

## Rettede grafer

### Definisjon

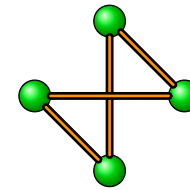
En *rettet graf* (engelsk: *directed*) er en graf hvor hver kant har en retning.



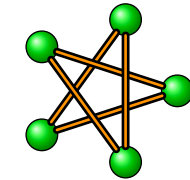
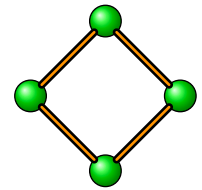
- En *relasjon* kan ses på som en rettet graf.
- Denne grafen svarer til relasjonen  $\{\langle A, B \rangle, \langle B, C \rangle, \langle C, D \rangle, \langle D, A \rangle\}$ .
- Hvis vi har en *symmetrisk* relasjon, riktignok, så kan vi tenke på denne som en vanlig (urettet) graf.
- Foreløpig skal vi ikke snakke om rettede grafer.

## Måter å tegne opp grafer på

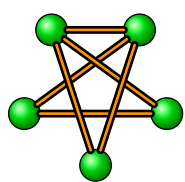
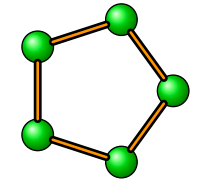
- Det fins ikke noen unik måte å tegne opp en graf på.
- Vi bryr oss for eksempel ikke om hvor lange kantene er, om de er bøyd, etc.
- Det eneste som spiller noen rolle er om to noder er forbundet med en kant.
- La oss se på noen eksempler. Følgende par av grafer er identiske, men tegnet opp på forskjellige måter.
- Vi forestiller oss at kantene er elastiske og at vi flytter om på nodene.
- Vi skal etter hvert presisere dette gjennom begrepet *isomorfi*. Følgende par av grafer kalles *isomorfe*.



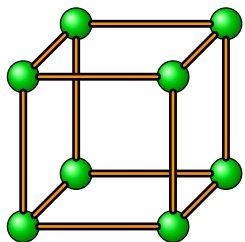
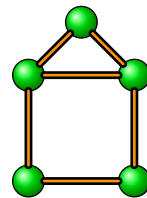
er lik



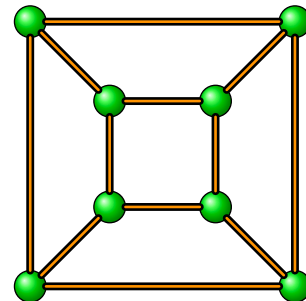
er lik



er lik



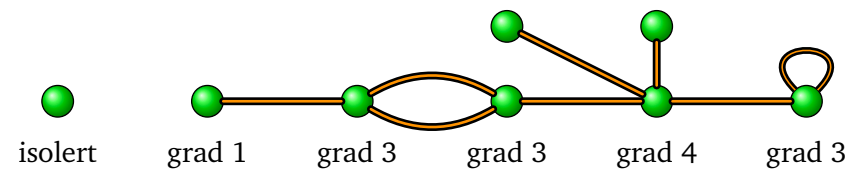
er lik



## Graden til noder

### Definisjon (Grad)

*Graden* (engelsk: *degree*) til en node  $v$  er antall kanter som ligger inntil  $v$ . En løkke teller som to kanter. Med  $\text{deg}(v)$  mener vi graden til  $v$ . En node med grad 0 kalles *isolert*.





## Graden til noder

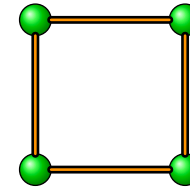
### Teorem

Summen av gradene til alle nodene i en graf er lik 2 ganger antallet kanter. Hvis  $V$  er mengden av noder og  $E$  er mengden av kanter, så har vi

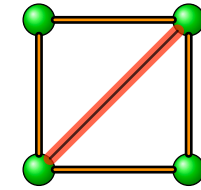
$$\sum_{v \in V} \deg(v) = 2|E|.$$

- Hver *kant* som legges til i en graf vil øke summen av gradene med to.
- La oss se på et eksempel.

## Graden til noder



Antall kanter er 4.  
Summen av gradene er 8.



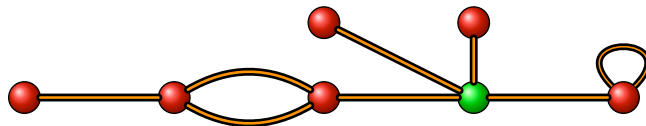
Antall kanter er 5.  
Summen av gradene er 10.

### Bevis

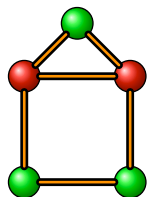
Hvis vi legger sammen gradene til alle nodene, så vil hver kant telle to ganger, siden hver kant ligger inntil to noder.

### Lemma (håndhilselemmaet)

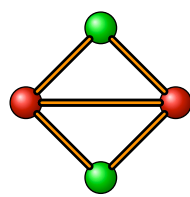
Det er alltid et partall antall noder av odde grad i en graf.



Her er det 6 noder (markert med rødt) med odde grad.



Her er det 2 noder av odde grad.



Her er det 2 noder av odde grad.

## Håndhilselemmaet

- Hvis vi forestiller oss mange mennesker samlet i et rom og at man håndhilser på hverandre, så må antallet av de som håndhilser på et odde antall personer være et partall.
- Vi kan representere denne situasjonen ved å representere menneskene som noder. En kant vil da representere at to personer håndhilser på hverandre.
- Det kalles et lemma fordi det ikke er så interessant i seg selv, men er nyttig for å bevise andre lemmaer og teoremer.
- Vi skal nå bevise håndhilselemmaet.

## Bevis (håndhilslemmaet)

La  $G$  være en graf. Vi deler mengden  $V$  av noder inn to:  $V_o$  er de som har odde grad (de som var røde) og  $V_p$  er de som har lik grad (de som var grønne). Vi har vist et teorem som sier at summene av gradene til *alle* nodene er to ganger antall kanter.

$$\sum_{v \in V} \deg(v) = 2|E|$$

Siden vi har delt opp mengden av noder i to, kan vi skrive dette slik:

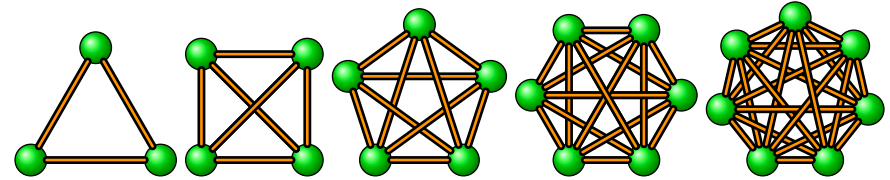
$$\sum_{v \in V_o} \deg(v) + \sum_{v \in V_p} \deg(v) = 2|E|$$

Siden  $2|E|$  er et partall og summen av gradene til nodene i  $V_p$  er et partall, så må summen av gradene til nodene i  $V_o$  også være et partall. Siden hver node i  $V_o$  har odde grad, så må det være et partall antall av dem.

## Komplette grafer

### Definisjon (Komplett graf)

En enkel graf er *komplett* hvis hver node er nabo med enhver annen node.



Komplette grafer ( $K_3, K_4, K_5, K_6, K_7$ ).

- Hvor mange kanter er det i en komplett graf?
- $K_3$  har 3 kanter.  $K_4$  har 6 kanter.  $K_5$  har 10 kanter.  $K_6$  har 15 kanter.  $K_7$  har 21 kanter. Er det noen som ser et mønster?

## Komplette grafer

### Teorem

Det er  $\binom{n}{2}$  kanter i en komplett graf med  $n$  noder.

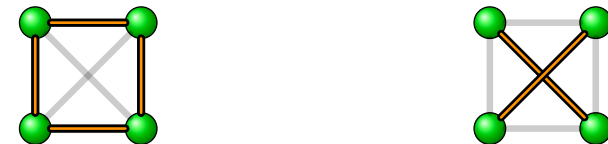
- Det er fordi vi har en kant for hver mengde av noder med kardinalitet 2.
- $K_n$  representerer kampoppsettet i en enkel serie som omfatter  $n$  lag.
- For det vanlige oppsettet med hjemme- og bortekamper vil kampoppsettet representeres av en komplett **rettet** graf.

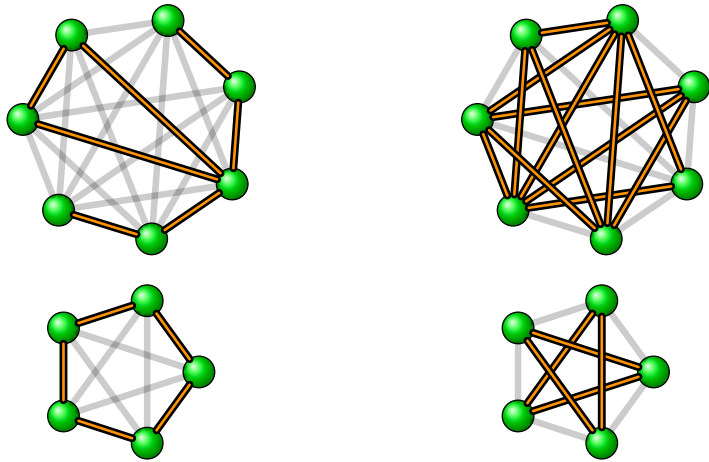
## Komplementet til en graf

### Definisjon (Komplement)

La  $G$  være en enkel graf. Da er *komplementet* til  $G$  grafen som har de samme nodene som  $G$ , men hvor to noder er naboer hvis og bare hvis nodene *ikke* er naboer i  $G$ . Vi skriver  $\bar{G}$  for komplementet til  $G$ .

Vi skal se på noen grafer og deres komplementer.

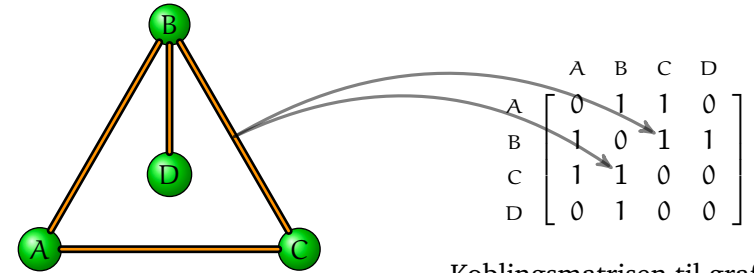




- I det siste tilfellet fikk vi ikke noen ny graf når vi tok komplementet.
- Slike grafer kalles *selv-komplementære*.

## Matriserepresentasjoner

På samme måte som med relasjoner, så har grafer en matriserepresentasjon. Vi kaller en slik matrise for en *koblingsmatrise* (engelsk: *adjacency matrix*).



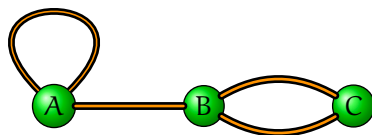
Koblingsmatrisen til grafen.

## Matriserepresentasjoner

### Definisjon (Koblingsmatrise)

Hvis  $G$  er en graf med  $n$  noder,  $v_1, \dots, v_n$ , så er koblingsmatrisen til  $G$  en  $n$ -ganger- $n$ -matrise hvor tallet i rad  $i$  og kolonne  $j$  er antall kanter mellom  $v_i$  og  $v_j$ .

	A	B	C
A	1	1	0
B	1	0	2
C	0	2	0



Koblingsmatrisen til grafen.

## Matriserepresentasjoner

- Legg merke til at vi kan speile en matrise om diagonalen.
- Det er fordi vi kun ser på *urettede* grafer.
- Hvis vi ser på rettede grafer, så kan vi ikke speile matrisen om diagonalen.
- Vi kunne også speile om diagonalen for *symmetriske relasjoner*.
- En forskjell mellom symmetriske relasjoner og grafer er at vi tillater *parallele* kanter i grafene.
- De kan vi ikke fange inn ved hjelp av en relasjon.
- Det fins flere matriser for samme graf, avhengig av rekkefølgen vi gir nodene i.