Grafteori-workshop

UNF København

Ungdommens Naturvidenskabelige Forening

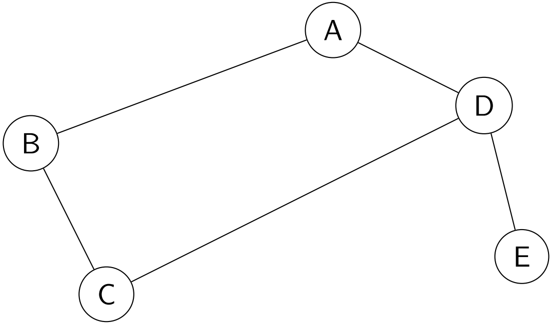
07-12-2021

Program

Program

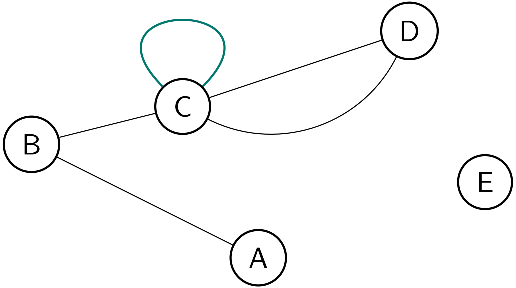
Hvad er en graf?

En graf er en samling af *knuder* forbundet af *kanter*.



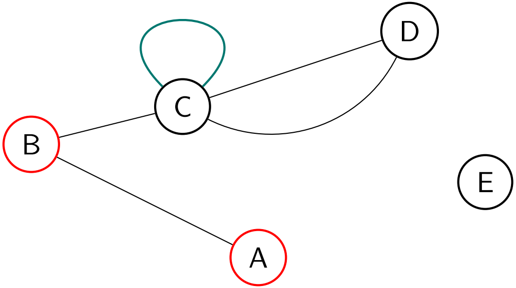
Vigtige begreber

- Løkke



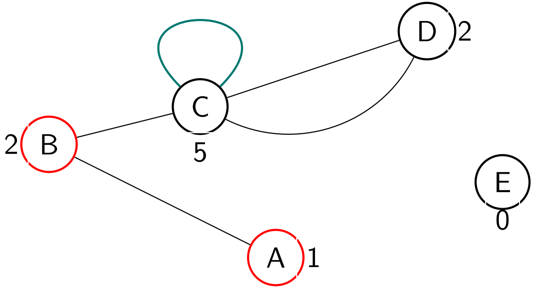
Vigtige begreber

- Løkke
- Naboknuder



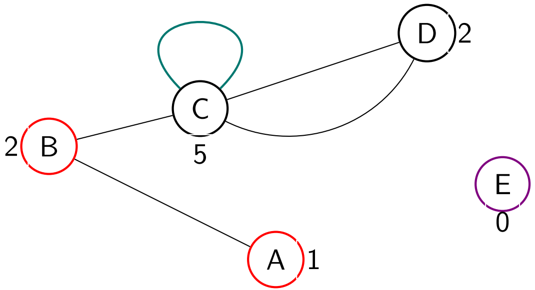
Vigtige begreber

- Løkke
- Naboknuder
- Valens



Vigtige begreber

- Løkke
- Naboknuder
- Valens
- Isoleret knude



Resultater om valens

Sætning 1.2

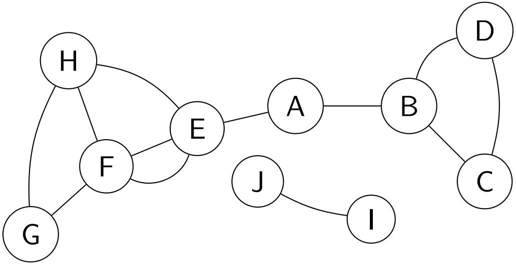
Den totale valens i en graf er antallet af kanter gange 2.

Korollar 1.3

Den totale valens for en graf er et lige tal.

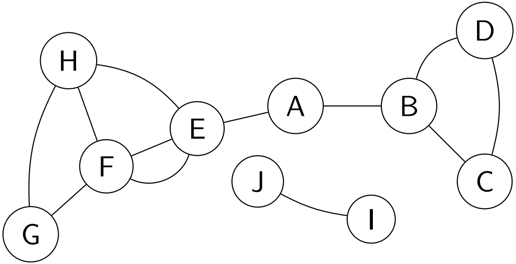
Flere definitioner

- Rute



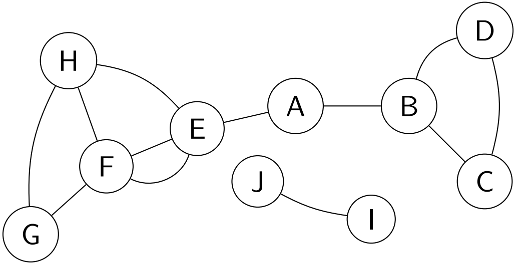
Flere definitioner

- Rute
- Tur



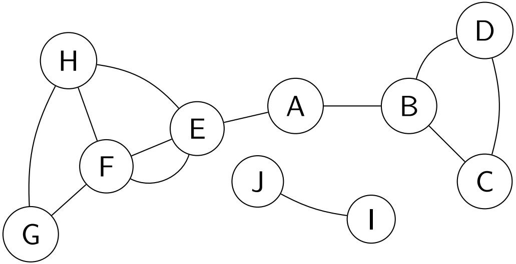
Flere definitioner

- Rute
- Tur
- Vej



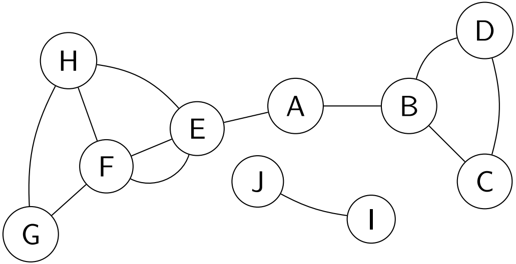
Flere definitioner

- Rute
- Tur
- Vej
- Lukkede ruter og ture

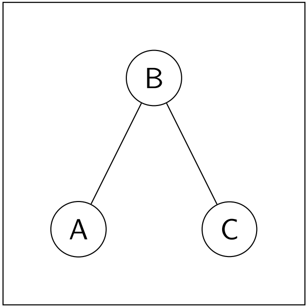


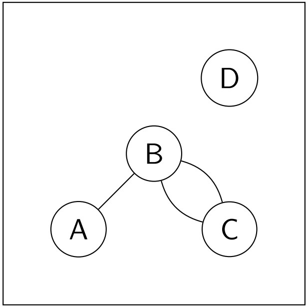
Flere definitioner

- Rute
- Tur
- Vej
- Lukkede ruter og ture
- Kreds

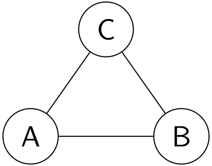


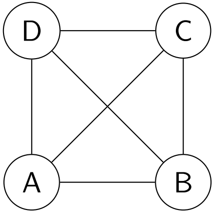
Simple grafer og sammenhængende grafer



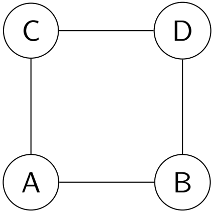


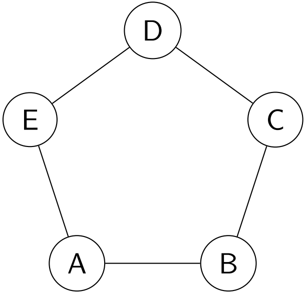
Komplette grafer, K_n



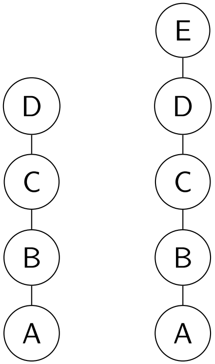


Kredsgrafer, C_n





Vejgrafer, P_n



Program

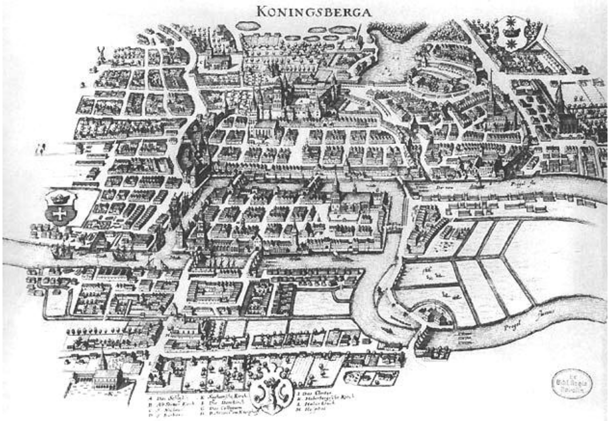
Opgaver

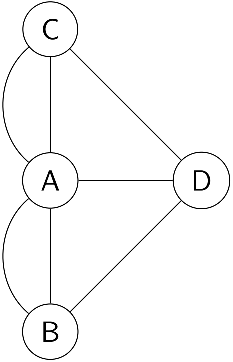
- Opgaverne er 1.1 til 1.14 i det udleverede materiale.

Program

Program

Euler og Königsberg (historisk)





Åbne og lukkede Eulerture

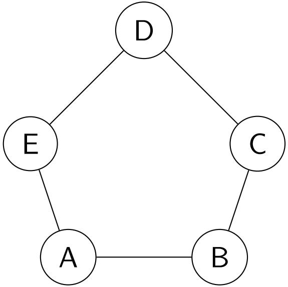
Definition

En *Eulertur* er en tur som indeholder alle grafens kanter.

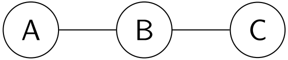
- Lukket: starter og slutter i samme knude.
- Åben: starter og slutter i forskellige knuder.

Eksempler

Lukket:



Åben:



Eulerture

Sætning 2.4

Betragt en graf G uden isolerede knuder.

1. G har en lukket Euler-tur, hvis og kun hvis den er sammenhængende og alle knuder har lige valens.
2. G har en åben Euler-tur, hvis og kun hvis den er sammenhængende og har præcist to knuder med ulige valens.

Opgaver

Opgaverne er 2.1 til 2.8 i det udleverede materiale.

Fleurys algoritme

Definition: Bro

En kant i en graf kaldes en *bro*, hvis grafen bliver usammenhængende, når kanten fjernes.

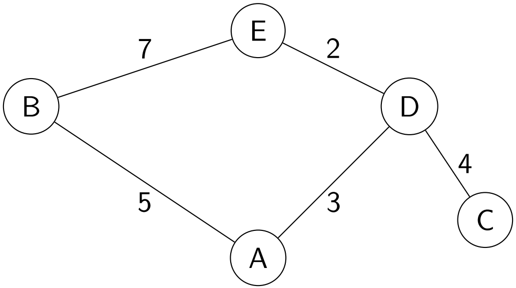
Fleurys algoritme (lukket Eulertur)

- Vælg en vilkårlig knude
- Gå langs en kant til en anden knude, "fjern" denne kant
- Hvis der kun er én kant videre, fortsæt da langs denne. Hvis der er flere kanter, vælg da en som ikke er en bro.
- Fortsæt denne proces, indtil Eulerturen er slut.

For en åben Eulertur: skal starte og slutte i en knude med ulige valens.

Program

Program


E
7
2
B
D
4
5
3
C
A

Djikstra's algoritme

1. Vælg en startknude s, og sæt afstanden fra s til alle andre knuder til uendelig, ∞. Markér s som besøgt (en besøgt knude bliver aldrig besøgt igen).

Djikstra's algoritme

1. Vælg en startknude s, og sæt afstanden fra s til alle andre knuder til uendelig, ∞. Markér s som besøgt (en besøgt knude bliver aldrig besøgt igen).
2. Udregn afstanden fra s til den nuværende knudes naboer. Hvis afstanden er kortere end den tidligere noterede afstand, erstat den gamle afstand med den nye.

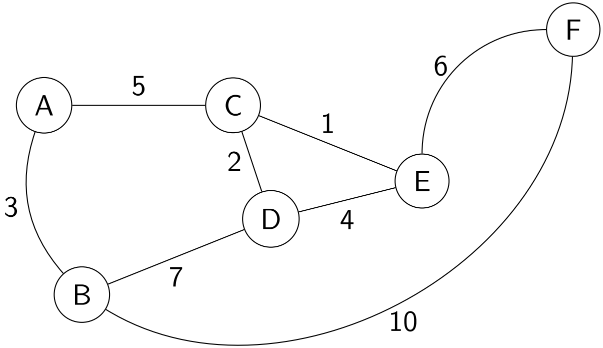
Djikstra's algoritme

1. Vælg en startknude s, og sæt afstanden fra s til alle andre knuder til uendelig, ∞. Markér s som besøgt (en besøgt knude bliver aldrig besøgt igen).
2. Udregn afstanden fra s til den nuværende knudes naboer. Hvis afstanden er kortere end den tidligere noterede afstand, erstat den gamle afstand med den nye.
3. Udvælg knuden (der ikke nødvendigvis er en nabo til den nuværende knude) med kortest afstand til s og markér denne som besøgt. Denne knude bliver også den nye nuværende knude.

Djikstra's algoritme

1. Vælg en startknude s, og sæt afstanden fra s til alle andre knuder til uendelig, ∞. Markér s som besøgt (en besøgt knude bliver aldrig besøgt igen).
2. Udregn afstanden fra s til den nuværende knudes naboer. Hvis afstanden er kortere end den tidligere noterede afstand, erstat den gamle afstand med den nye.
3. Udvælg knuden (der ikke nødvendigvis er en nabo til den nuværende knude) med kortest afstand til s og markér denne som besøgt. Denne knude bliver også den nye nuværende knude.
4. Gentag trin 2 og 3 til alle knuder er besøgt.

Eksempel



Eksempel (fortsat)

Nuværende knude	B	C	D	E	F
$A \rightarrow$	3	5	∞	∞	∞
$B \rightarrow$	$\checkmark$	5	$3+7$	∞	$3+10$
$C \rightarrow$		$\checkmark$	$5+2$	$5+1$	13
$E \rightarrow$			7	$\checkmark$	$6+6$
$D \rightarrow$			$\checkmark$		12
$F \rightarrow$					$\checkmark$

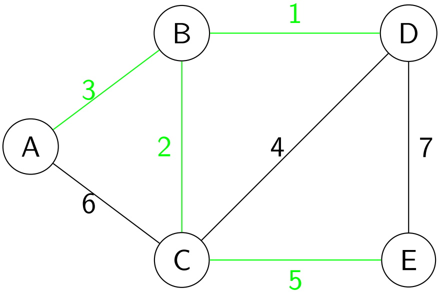
Afrunding af Djikstra's algoritme

Sætning

Djikstra's algoritme giver den korteste vej.

Vi udelader beviset, men det kan let findes online eller i lærebøger såsom "Introduction to Algorithms"

Mindste udspændende træer



Kruskals algoritme

1. Sortér kanterne i grafen efter vægt fra mindst til størst. Har nogle af kanterne ens vægt, er rækkefølgen ligegyldig. Lad A betegne mængden (som i starten er tom) af de kanter, der skal være med i vores træ.

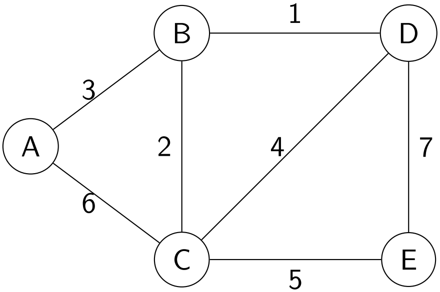
Kruskals algoritme

1. Sortér kanterne i grafen efter vægt fra mindst til størst. Har nogle af kanterne ens vægt, er rækkefølgen ligegyldig. Lad A betegne mængden (som i starten er tom) af de kanter, der skal være med i vores træ.
2. Kig på kanten med mindst vægt. Hvis tilføjelsen af kanten til A skaber en kreds i A, spring kanten over. Ellers tilføj kanten til A.

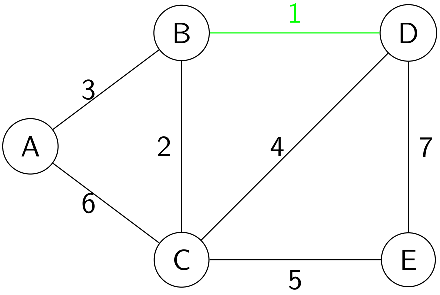
Kruskals algoritme

1. Sortér kanterne i grafen efter vægt fra mindst til størst. Har nogle af kanterne ens vægt, er rækkefølgen ligegyldig. Lad A betegne mængden (som i starten er tom) af de kanter, der skal være med i vores træ.
2. Kig på kanten med mindst vægt. Hvis tilføjelsen af kanten til A skaber en kreds i A, spring kanten over. Ellers tilføj kanten til A.
3. Gentag trin 2 for den næste kant i den sorterede liste af kanter, indtil alle kanter er blevet checket.

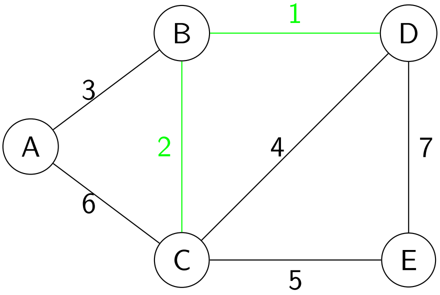
Eksempel



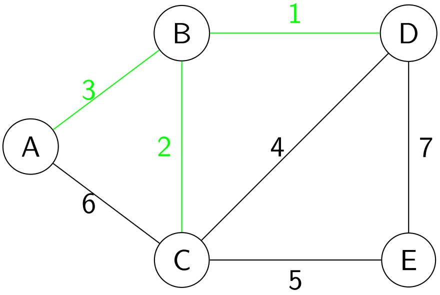
Eksempel



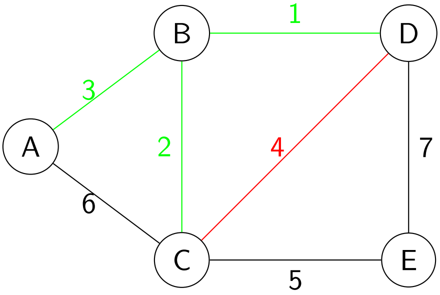
Eksempel



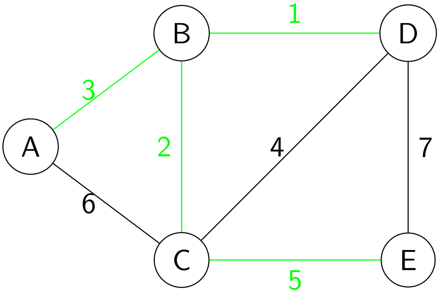
Eksempel



Eksempel



Eksempel



Program

Opgaver

- Opgaverne er 3.1 til 3.9, hvis man vil arbejde med Djikstras algoritme. Vil man arbejde med Kruskals algoritme, se opgave 3.10 til 3.16.

Program

Tak for denne gang

Andre arrangementer (foredrag, workshops og andet) i UNF København kan ses her:
https://unf.dk/aktiviteter/?department=kbh

Information om vores sommer-sciencecamps kan ses her: https://unf.dk/sciencecamps/