Grafer & Algoritmer

Peter Levinsky IT Roskilde

08.04.2024

Noget om grafer

To kategorier

- Træer
- Grafer / Net

Og de kan være **med** eller **uden** Vægte

Træer

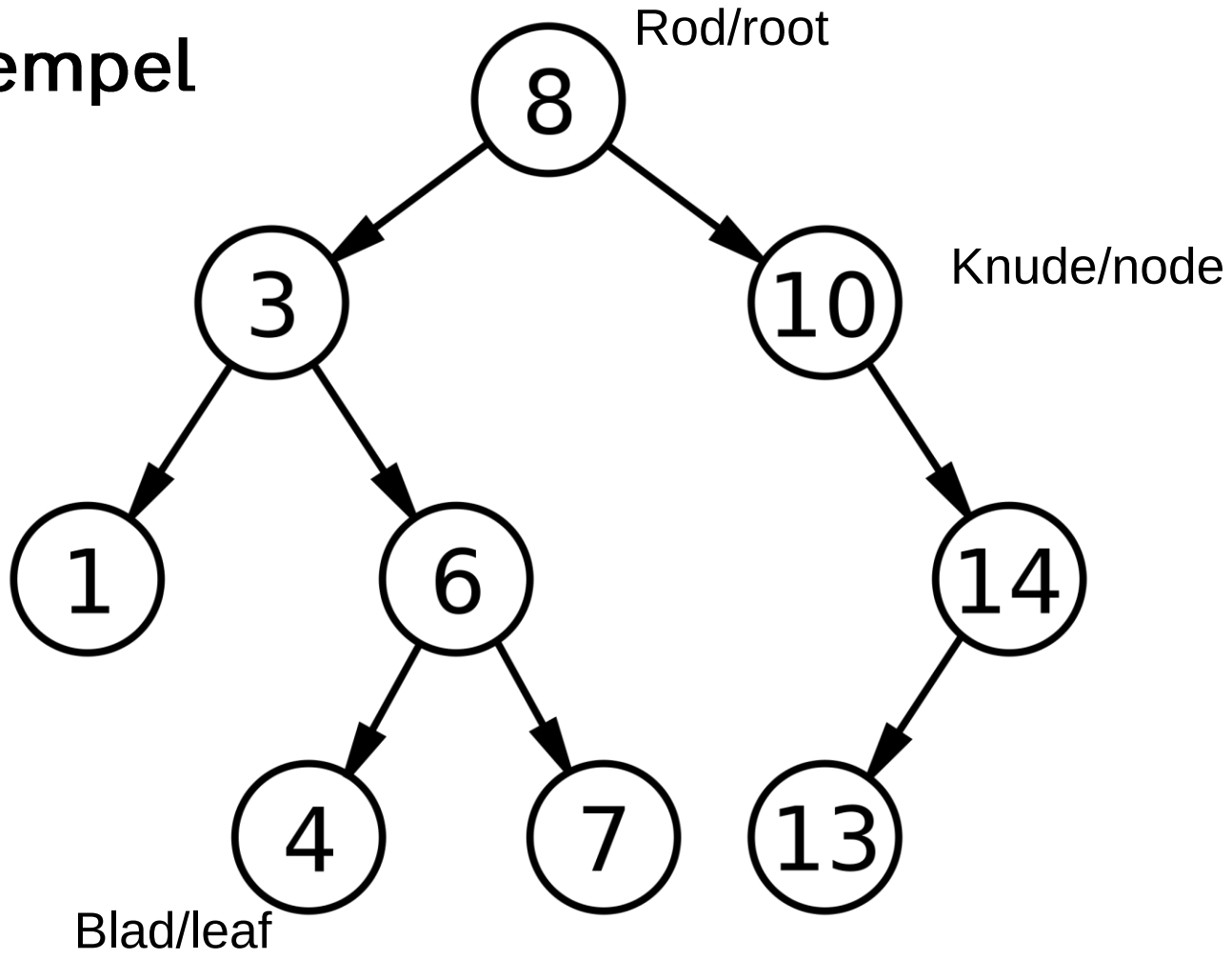
Kendetegn ved træer (egentlig en special udgave af en graf)

- Én rod
- Unik vej mellem alle kunder (noder)

Type af træer (med eller uden vægte mellem hver knude)

- Usorteret træ
- Binær sorteret træ
  - Ubalanceret
  - Balanceret
- 2-3 træer
- Hob (heap)

Træer – Et eksempel



Grafer

Kendetegn ved grafer

- Et netværk af knuder (noder)
  - Der kan være et start knude og evt. slut knude
  - Mellem to knuder er der en kant (dge),
    som kan være med eller uden vægt)

- Flere veje mellem kunder (noder)

Grafer – Et eksempel

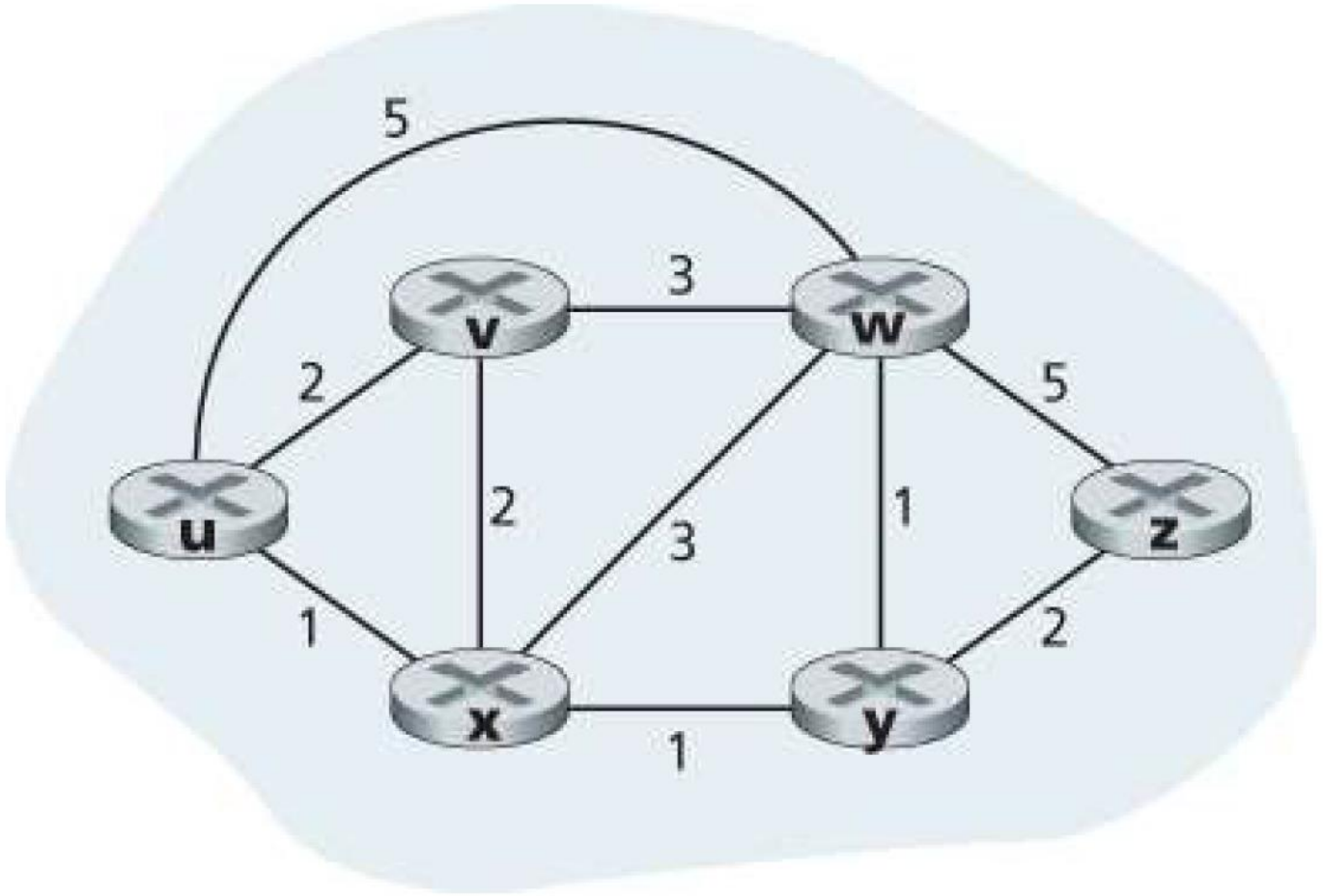


Figure 5.3 Abstract graph model of a computer network

Grafer & algoritmer

Forskellige algoritmer

- Dijkstra's algorithm (korteste vej til alle knuder)

  Animation:
  https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#/media/File:Dijkstras_progress_animation.gif

- A* algorithm (korteste vej mellem to knuder)

  Animation:
  https://en.wikipedia.org/wiki/A*_search_algorithm#/media/File:Astar_progress_animation.gif

- Spanning tree (omdanner en graf til et træ (min eller max))

- Maximum Flow algorithm
  (hvor meget kan der passere mellem to knuder)

Dijkstra’s algorithm - Grafer & algoritmer

Bliver også benyttet som Open Shortest Path First (OSPF) eller Link State

1. Indsamler information om alle knuder og kanter (nodes/edges)

2. Vælger en start knude (første del af ‘løsningsnet’)

3. Vælger den ‘mindste’ kant til næste knude
    1. Knude indføjes i ‘løsningsnet’
    2. Nye kanter udregnes
    3. Dette fortsættes til alle knuder er i løsningsnettet

Dijkstra's algorithm - Grafer & algoritmer

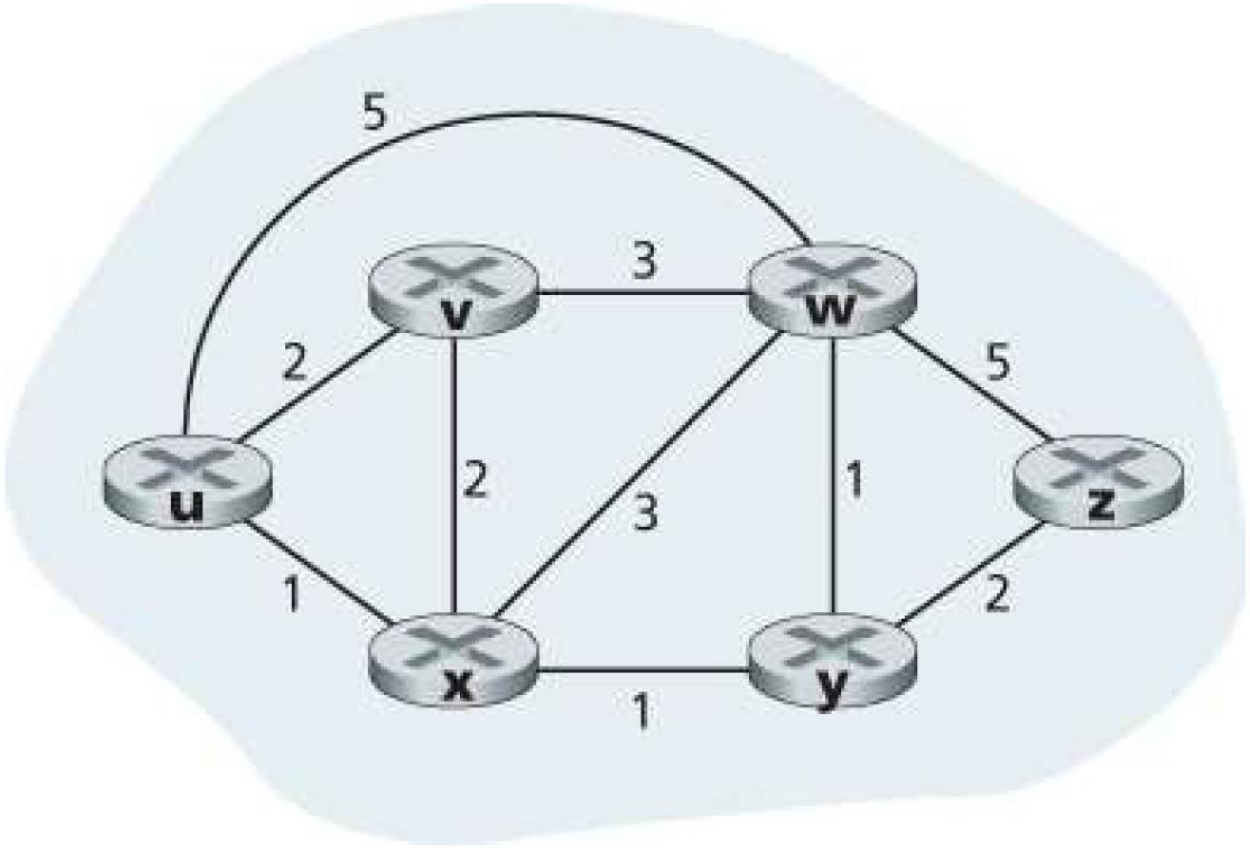
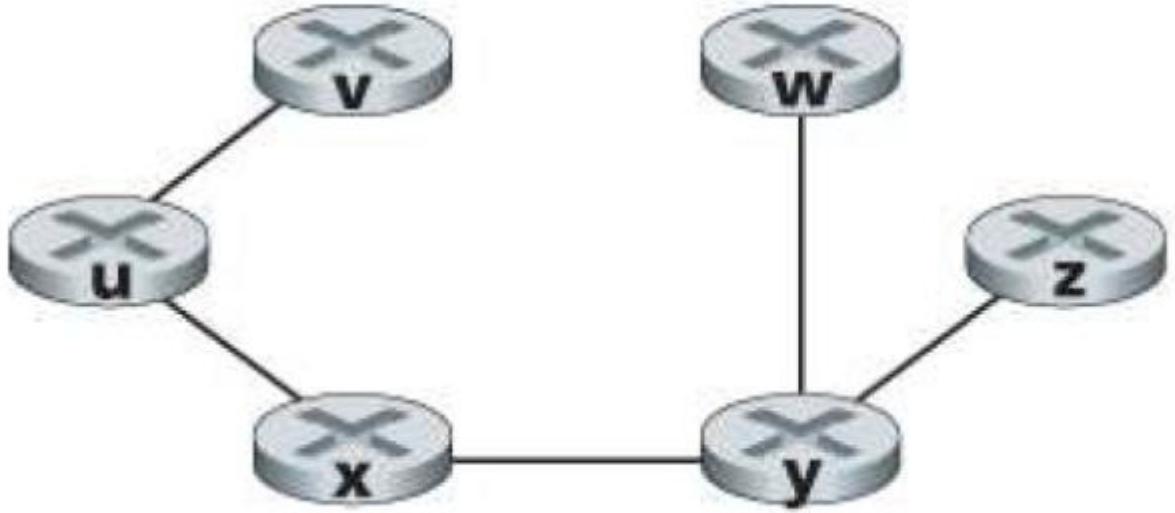


Figure 5.3 Abstract graph model of a computer network

Dijkstra's algorithm - Grafer & algoritmer



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Figure 5.4 Least cost path and forwarding table for node u

A* algorithm - **Grafer & algoritmer**

Bliver benyttet i spil til at finde vej mellem to givne punkter

1. Indsamler løbende information om alle knuder og kanter (nodes/edges)

2. Vælger en start knude og en slut knude

3. Vælger den ‘mindste’ afstand + ‘estimeret’ (heuristik) vej til slut
    1. Knude indføjes i ‘løsningsnet’
    2. Fra løsningsnet vælges mindste estimeret afstand til slut knude
        1. Tilføjes ‘løsningsnet’
        2. Husker hvor du kom fra

A* algorithm - Heuristik

For at A* algoritmen skal være effektiv skal heuristikken være god

- Godt i x,y koordinater spil (eller x,y,z),
  - Heuristik afstand mellem to punkter fx:
    - Absolut
    - Pythagyras
    - …

- Mindre godt i computer net,
  - Der er ingen umiddelbar afstand mellem to routere
    => svært at definere en god heuristik
    eller afstanden er altid en (hvorved den så minder om Dijkstra's algoritme)

A* algorithm - eksempel

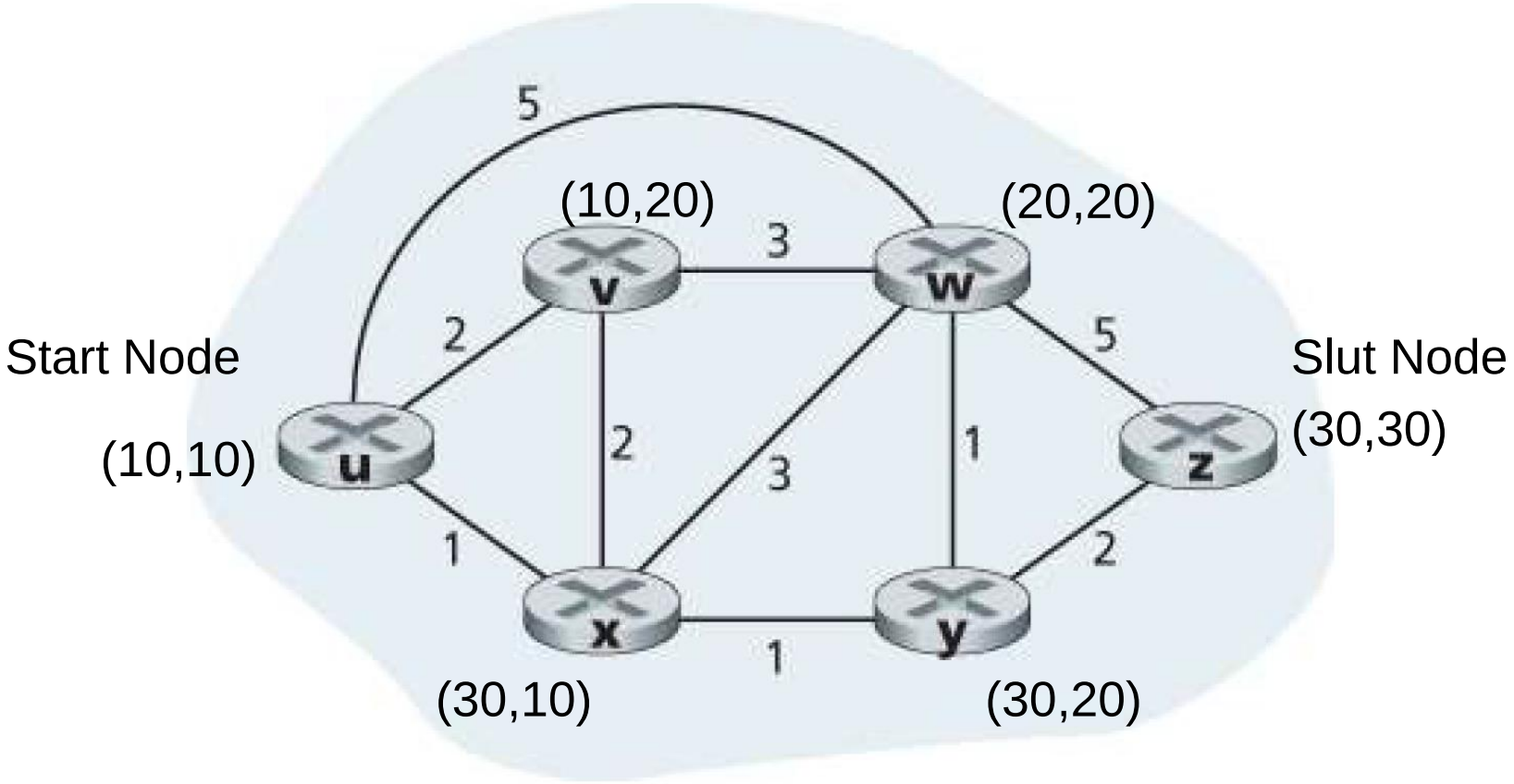


Figure 5.3 Abstract graph model of a computer network

That’s it

- **Training: xxxx**

- **And the Mandatory Assignment** ☺