# Grafen
## Algoritme van Floyd

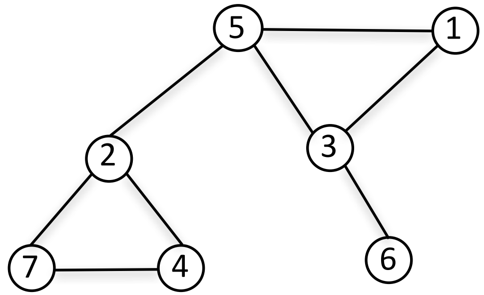
Lesweek 9

Samenvatting van (een gedeelte van) de cursusnota's

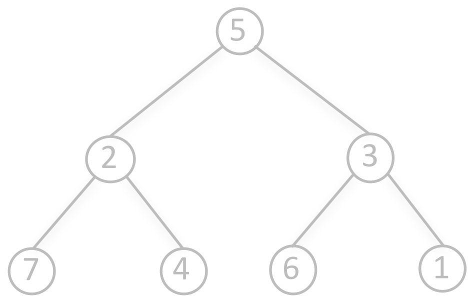
# Datastructuren

- **Lineair**: de elementen vormen een rij.

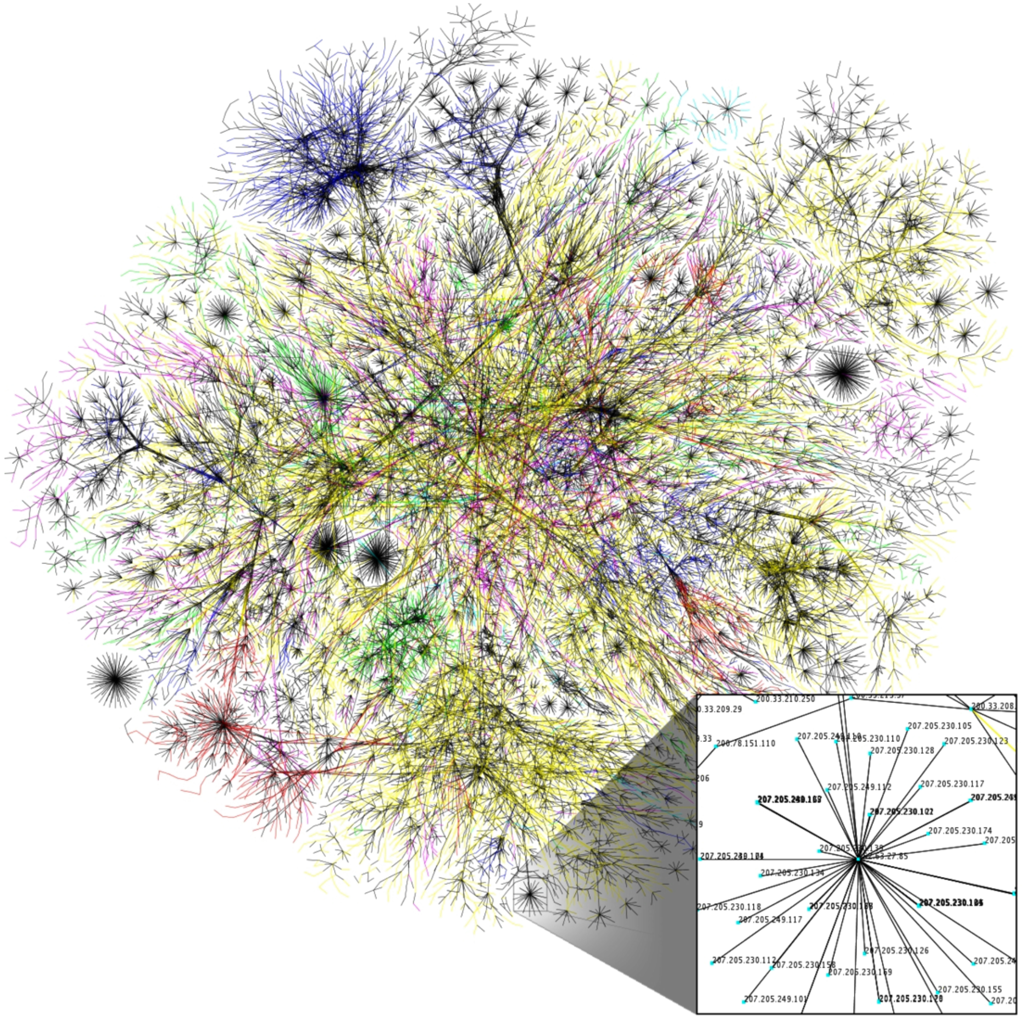

- **Niet-lineair**: de elementen vormen geen rij.
  - **Graaf**: lussen zijn toegelaten.




  - **Boom**: lussen zijn niet toegelaten.



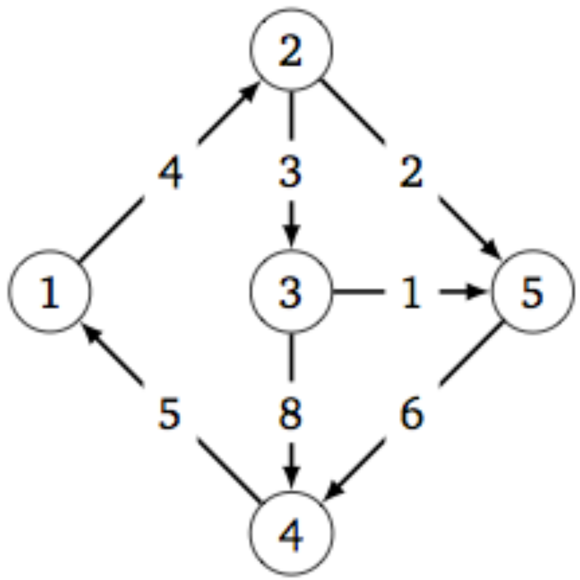
# Definitie



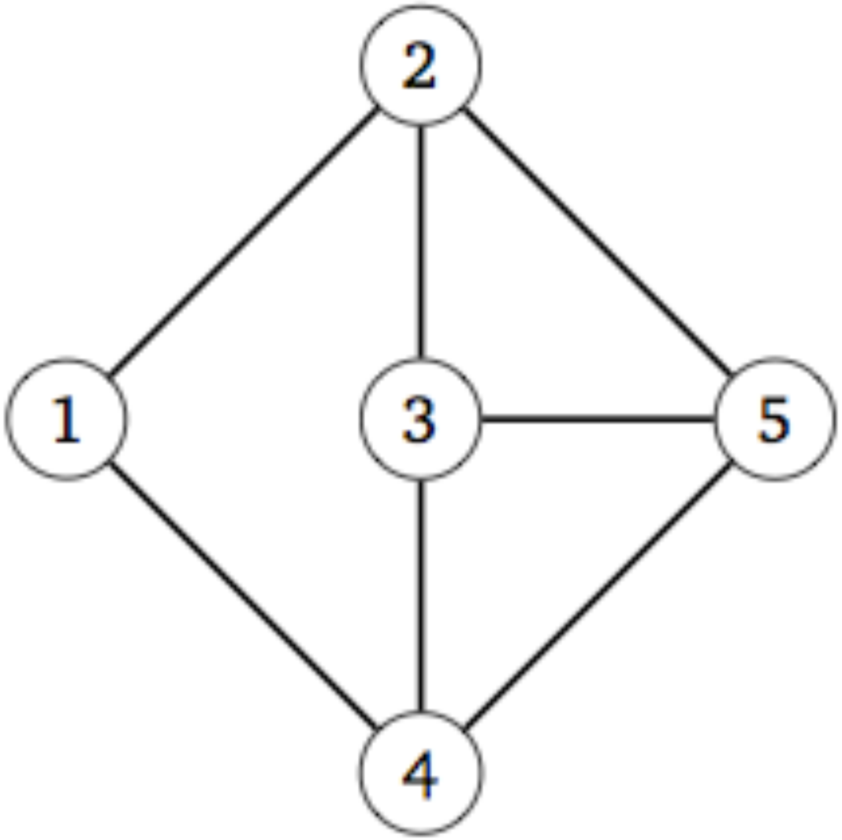
Graaf = verzameling knooppunten die met mekaar verbonden zijn

# Definities

netwerk (gewogen) graaf

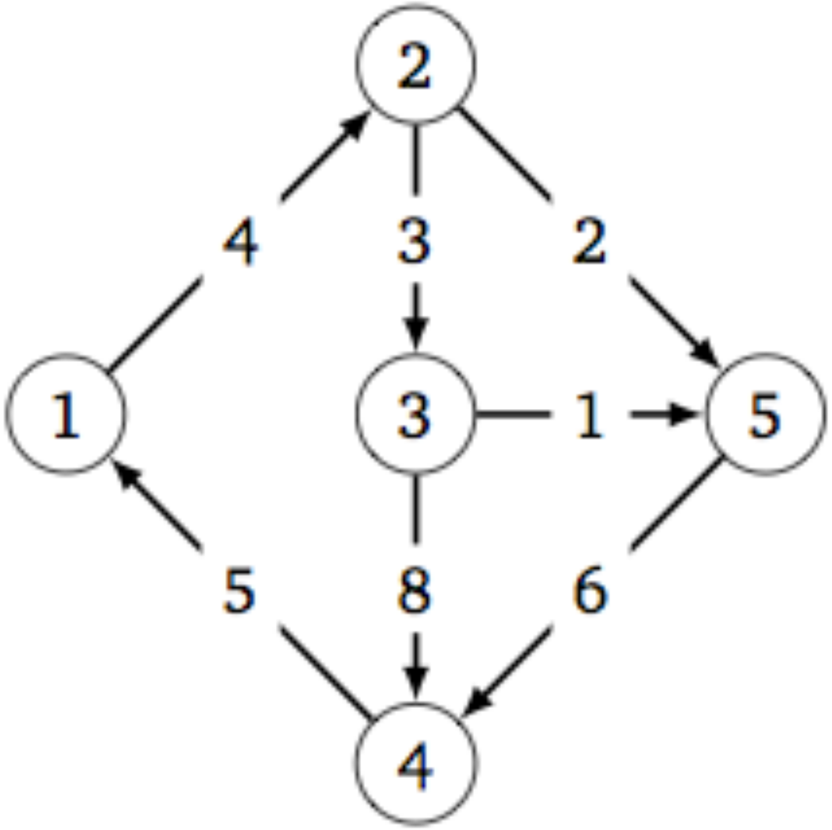


# Verbindingsmatrix




$$
\begin{array}{cc}
 & \text{naar} \\
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\text{van} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \left[\begin{array}{ccccc} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{array}\right]
\end{array}
$$

# Gewichtenmatrix




$$
\begin{array}{c}
 \\ \\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\text{naar} \\
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\left[\begin{array}{ccccc}
0 & 4 & \infty & \infty & \infty \\
\infty & 0 & 3 & \infty & 2 \\
\infty & \infty & 0 & 8 & 1 \\
5 & \infty & \infty & 0 & \infty \\
\infty & \infty & \infty & 6 & 0
\end{array}\right]
\end{array}
$$

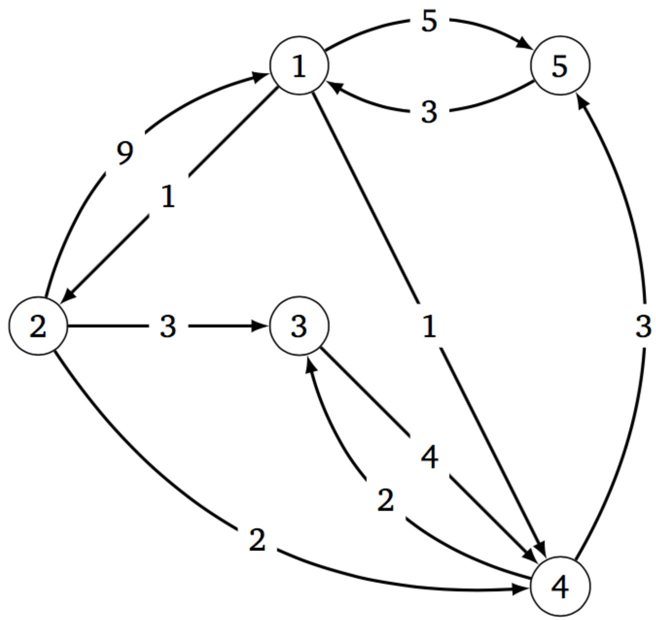
# Algoritme van Floyd

- Netwerk (gewogen graaf)
- Zoekt paden met **kleinste gewicht**
- Geeft lijst met “kortste” paden tussen eender welke twee knooppunten van graaf

BFS: “kortste” pad is pad met minst aantal knooppunten

Gewichtenmatrix: toont afstanden tussen knooppunten zonder “tussenstation”

$$
\begin{array}{ccc}
 & & \text{naar} \\
 & & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\text{van} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix}
0 & 1 & \infty & 1 & 5 \\
9 & 0 & 3 & 2 & \infty \\
\infty & \infty & 0 & 4 & \infty \\
\infty & \infty & 2 & 0 & 3 \\
3 & \infty & \infty & \infty & 0
\end{bmatrix}
\end{array}
$$



# Algoritme van Floyd

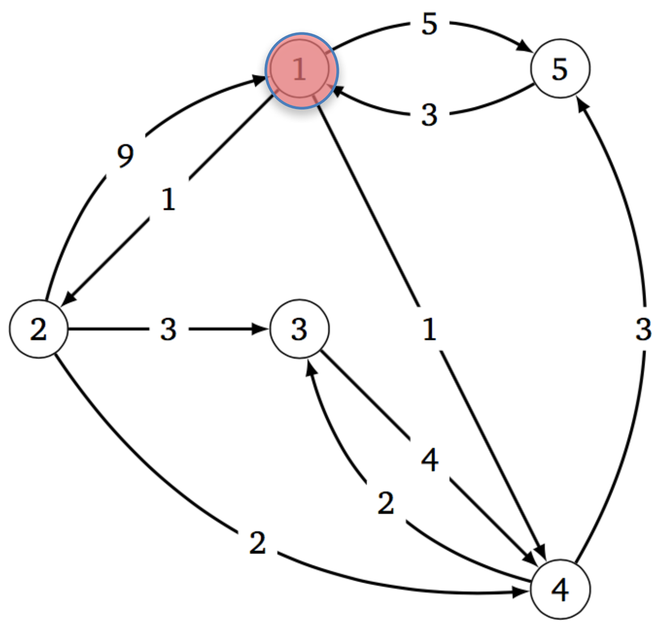
Systematisch één knooppunt meer toelaten als 'tussenstation' om afstanden te optimaliseren

- Eerst knooppunt 1, dan knooppunt 2 enz (i.e. systematisch)
- Telkens je een knooppunt als tussenstation toelaat, afstanden optimaliseren
- Gebruikte tussenstations noteren in Pointermatrix

**Eindresultaat**:

- Pointermatrix met gebruikte tussenstations

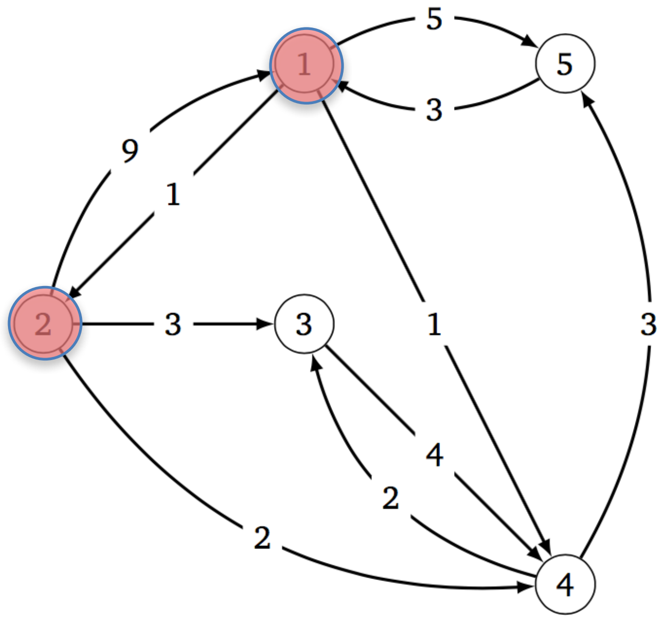
Stap 1: knoop 1 als tussenstap toelaten indien afstand korter wordt




$$
\begin{array}{cc}
 & \text{naar} \\
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\text{van} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix}
0 & 1 & \infty & 1 & 5 \\
9 & 0 & 3 & 2 & \infty \\
\infty & \infty & 0 & 4 & \infty \\
\infty & \infty & 2 & 0 & 3 \\
3 & \infty & \infty & \infty & 0
\end{bmatrix}
\end{array}
$$

$$
D^{(1)} = \begin{bmatrix}
0 & 1 & \infty & 1 & 5 \\
9 & 0 & 3 & 2 & 14 \\
\infty & \infty & 0 & 4 & \infty \\
\infty & \infty & 2 & 0 & 3 \\
3 & 4 & \infty & 4 & 0
\end{bmatrix}
\quad
P = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix}
$$

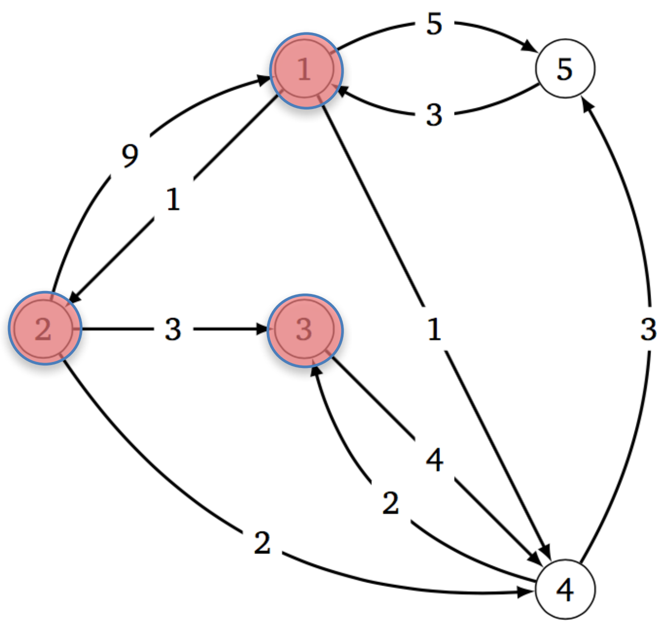
Stap 2:
niet alleen knoop 1 als tussenstap toelaten,
maar ook knoop 2 indien afstand korter wordt




$$D^{(1)} = \begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 4 & \infty & 4 & 0 \end{bmatrix}$$

$$D^{(2)} = \begin{bmatrix} 0 & 1 & 4 & 1 & 5 \\ 9 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 4 & 7 & 4 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

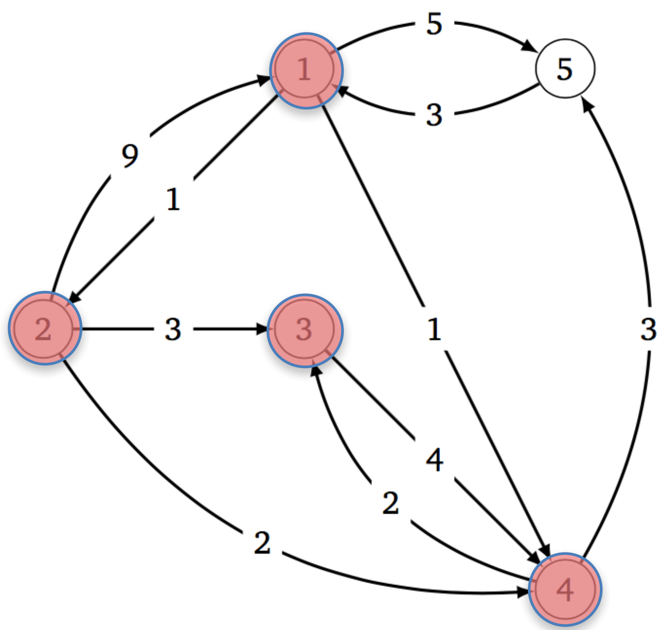
Stap 3:
niet alleen knoop 1 en 2 als tussenstap toelaten,
maar ook knoop 3 indien afstand korter wordt




$$D^{(2)} = \begin{bmatrix} 0 & 1 & 4 & 1 & 5 \\ 9 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 4 & 7 & 4 & 0 \end{bmatrix}$$

$$D^{(3)} = \begin{bmatrix} 0 & 1 & 4 & 1 & 5 \\ 9 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 4 & 7 & 4 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

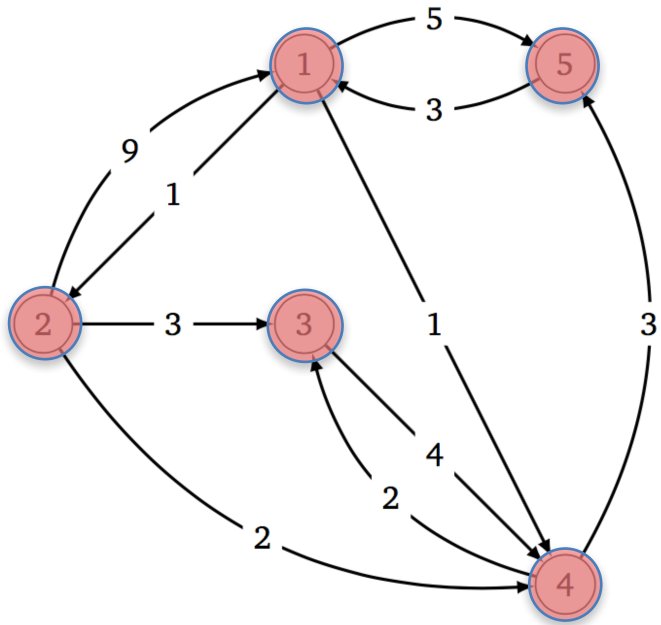
Stap 4:
niet alleen knoop 1, 2 en 3 als tussenstation toelaten, maar ook knoop 4 indien afstand korter wordt




$$D^{(3)} = \begin{bmatrix} 0 & 1 & 4 & 1 & 5 \\ 9 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 4 & 7 & 4 & 0 \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & 1 & 3 & 1 & 4 \\ 9 & 0 & 3 & 2 & 5 \\ \infty & \infty & 0 & 4 & 7 \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 4 & 6 & 4 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

Stap 5:
ook knoop 5 indien afstand korter wordt




$$D^{(4)} = \begin{bmatrix} 0 & 1 & 3 & 1 & 4 \\ 9 & 0 & 3 & 2 & 5 \\ \infty & \infty & 0 & 4 & 7 \\ \infty & \infty & 2 & 0 & 3 \\ 3 & 4 & 6 & 4 & 0 \end{bmatrix}$$

$$D^{(5)} = \begin{bmatrix} 0 & 1 & 3 & 1 & 4 \\ \mathbf{8} & 0 & 3 & 2 & 5 \\ \mathbf{10} & \mathbf{11} & 0 & 4 & 7 \\ \mathbf{6} & \mathbf{7} & 2 & 0 & 3 \\ 3 & 4 & 6 & 4 & 0 \end{bmatrix} P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

# Pad aflezen uit pointermatrix

Voorbeeld: kortste pad van knoop 3 naar knoop 1:

3 … 1

- Stap 1: rij 3 kolom 1 van P-matrix = 5 → voorganger van 1 in kortste pad van 3 naar 1 is knoop met nummer 5

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ \boxed{5} & 5 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

Dus van 3 naar 1 : 3 … 5 … 1

# Pad aflezen uit pointermatrix

Voorbeeld: kortste pad van knoop 3 naar knoop 1

– Stap 2: kortste pad van 3 naar 5: rij 3 kolom 5 van P-matrix = 4 → voorganger van 5 in kortste pad van 1 naar 3 is knoop 4

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & \boxed{4} \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

Dus van 3 naar 1 : 3 … 4 … 5 … 1

# Pad aflezen uit pointermatrix

Voorbeeld: kortste pad van knoop 3 naar knoop 1

- Stap 3: kortste pad van 3 naar 4: rij 3 kolom 4 van P-matrix = 0 dus geen tussenliggende stap tussen 3 en 4 in kortste pad van 3 naar 1

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ 5 & 5 & 0 & \boxed{0} & 4 \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

Dus van 3 naar 1 : 3 → 4 … 5 … 1

# Pad aflezen uit pointermatrix

Voorbeeld: kortste pad van knoop 3 naar knoop 1

- Stap 4: kortste pad van 4 naar 5: rij 4 kolom 5 van P-matrix = 0 dus geen tussenliggende stap

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & \boxed{0} \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

Dus van 3 naar 1 : 3 → 4 → 5 … 1

# Pad aflezen uit pointermatrix

Voorbeeld: kortste pad van knoop 3 naar knoop 1

- Stap 5: kortste pad van 5 naar 1: rij 5 kolom 1 van P-matrix = 0 dus geen tussenliggende stap
- Alle tussenliggende stappen zijn gecontroleerd, dus pad is gevonden

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 0 \\ \boxed{0} & 1 & 4 & 1 & 0 \end{bmatrix}$$

Dus van 3 naar 1 : 3 → 4 → 5 → 1

# Pad aflezen uit pointermatrix

- Voorbeeld: kortste pad van knoop 3 naar knoop 1

3 ........................ 5 ...... 1

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ \boxed{5} & 5 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

# Pad aflezen uit pointermatrix

- Voorbeeld: kortste pad van knoop 3 naar knoop 1

3 ..... 4 ..... 5 ..... 1

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ \boxed{5} & 5 & 0 & 0 & \boxed{4} \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

# Pad aflezen uit pointermatrix

- Voorbeeld: kortste pad van knoop 3 naar knoop 1

$$3 \;\boxed{\rightarrow}\; 4 \;\ldots.\; 5 \;\ldots.\; 1$$

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ \boxed{5} & 5 & 0 & \boxed{0} & \boxed{4} \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

# Pad aflezen uit pointermatrix

- Voorbeeld: kortste pad van knoop 3 naar knoop 1

3 → 4 → 5 .... 1

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ \boxed{5} & 5 & 0 & \boxed{0} & \boxed{4} \\ 5 & 5 & 0 & 0 & \boxed{0} \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

# Pad aflezen uit pointermatrix

- Voorbeeld: kortste pad van knoop 3 naar knoop 1

3 → 4 → 5 → 1

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ \boxed{5} & 5 & 0 & \boxed{0} & \boxed{4} \\ 5 & 5 & 0 & 0 & \boxed{0} \\ \boxed{0} & 1 & 4 & 1 & 0 \end{bmatrix}$$

# Lengte van pad berekenen

- Afstanden tussen knooppunten berekenen uit gewichtenmatrix
- Voorbeeld: pad van knoop 3 naar knoop 1 is:

3 → 4 → 5 → 1 en

heeft gewicht

4 + 3 + 3 = 10

$$
\begin{array}{cc}
 & \text{naar} \\
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\text{van} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix}
0 & 1 & \infty & 1 & 5 \\
9 & 0 & 3 & 2 & \infty \\
\infty & \infty & 0 & 4 & \infty \\
\infty & \infty & 2 & 0 & 3 \\
3 & \infty & \infty & \infty & 0
\end{bmatrix}
\end{array}
$$

Als je D-matrices bewaart, kan je daar ook de lengte van de paden aflezen