

# Wiskunde

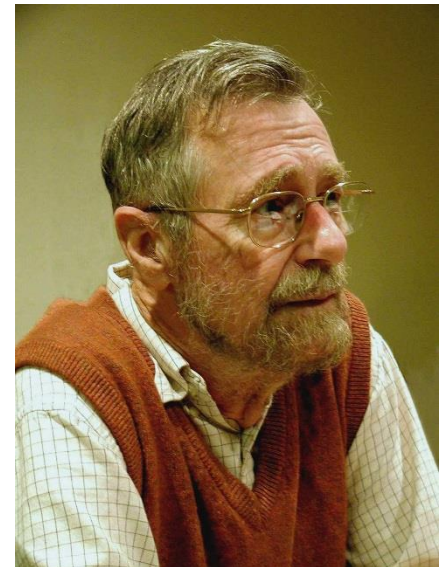
Gerwin van Dijken  
([gerwin.vandijken@inholland.nl](mailto:gerwin.vandijken@inholland.nl))

# Wiskunde programma

- Blok 3:
  - Recursie (1)
  - Recursie (2)
  - Markov keten
- Blok 4:
  - Grafen
  - **Shortest path algorithms (Dijkstra, A\*)**
  - Minimum Spanning Tree algorithms (Prim, Kruskal)
  - Proeftentamen / herhaling

# Shortest Path

- Kortste pad bepalen (tussen source en destination)
- 'Dijkstra'-algoritme, Edsger Dijkstra (1959)
- Wat is het kortste pad tussen X en Y?
- Praktijktoepassingen:
  - Routebepaling (TomTom)
  - Telecommunicatie (OSPF)
  - Gaming
  - ...



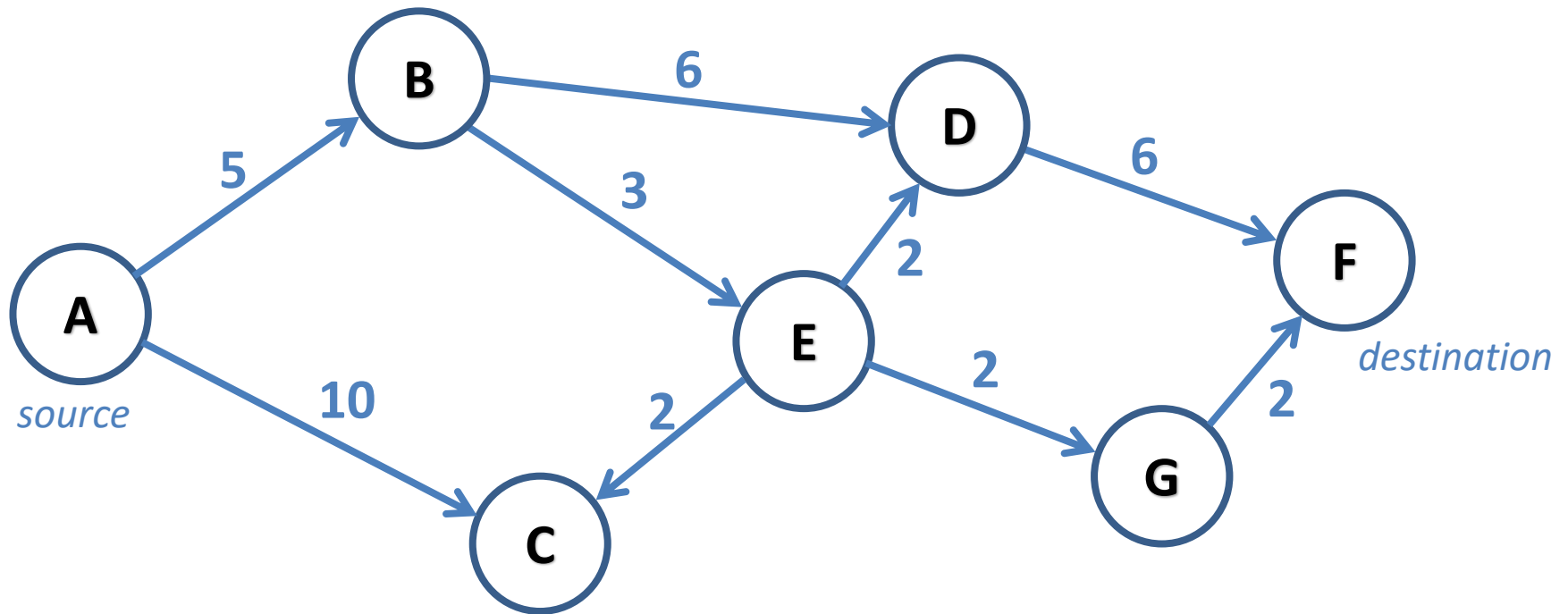
# Dijkstra algoritme

- Werkt op een gerichte graaf met 'gewichten' tussen de nodes ( $\geq 0$ )
- De gewichten kunnen afstanden zijn, maar ook bv kosten of tijden of ...
- Single-source algoritme → bepaalt de minimale afstand tussen source-node en alle andere nodes
- Als kortste afstand tot destination-node bepaald is, zou je eerder kunnen stoppen...

# Dijkstra algoritme

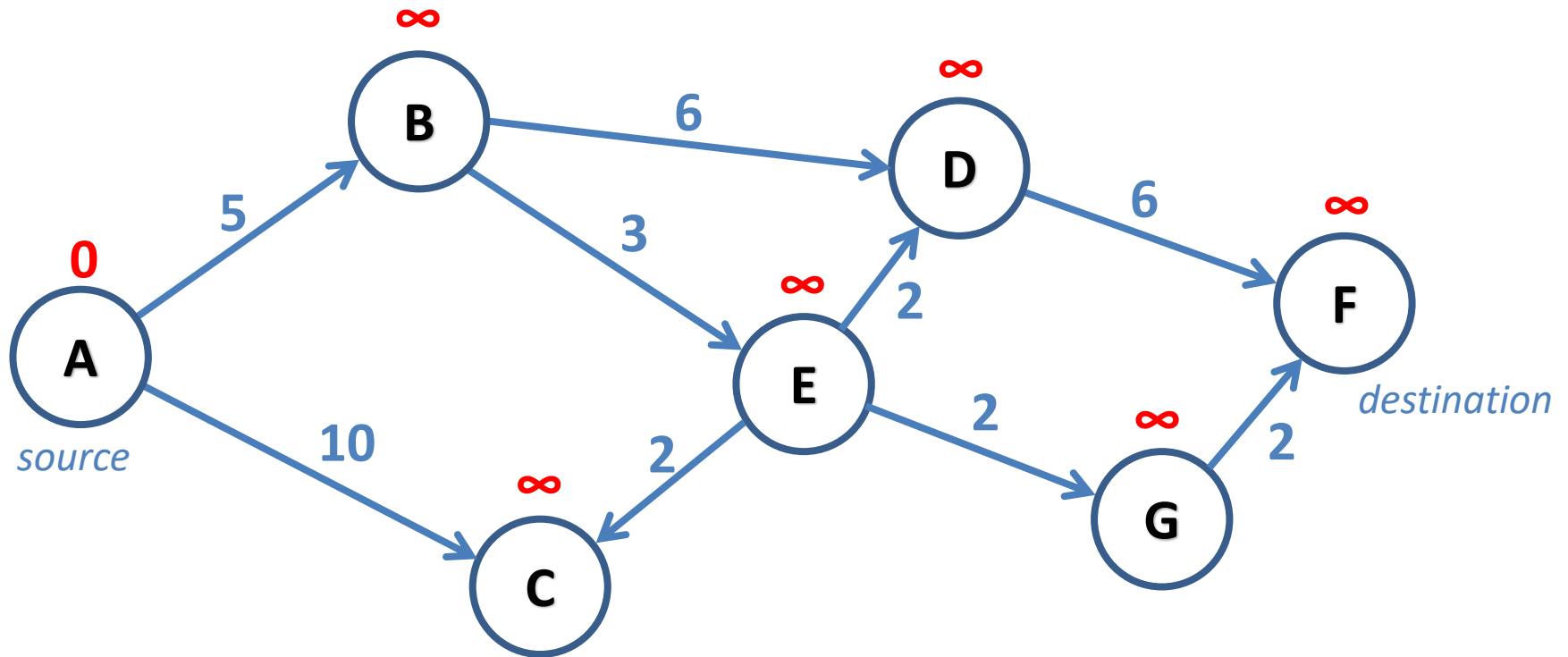
1. Zet afstandswaarde van startnode op 0 en van alle andere nodes op 'oneindig'
2. Plaats alle nodes in lijst '*unvisited*' ( $Q$ )
3. Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$  (pad is bekend!)  
*(bij meerdere dezelfde laagste afstandswaarden: kies willekeurig)*
4. Bepaal van elke (unvisited) neighbour van node  $u$  (uit stap 3) de afstand; als deze kleiner is dan de huidige: update afstandswaarde + set parent (naar  $u$ )  
*(negeer nodes waarvan kortste pad al bekend is)*
5. Ga naar stap 3 (zolang er nog nodes in  $Q$  zijn)

# Een voorbeeld...



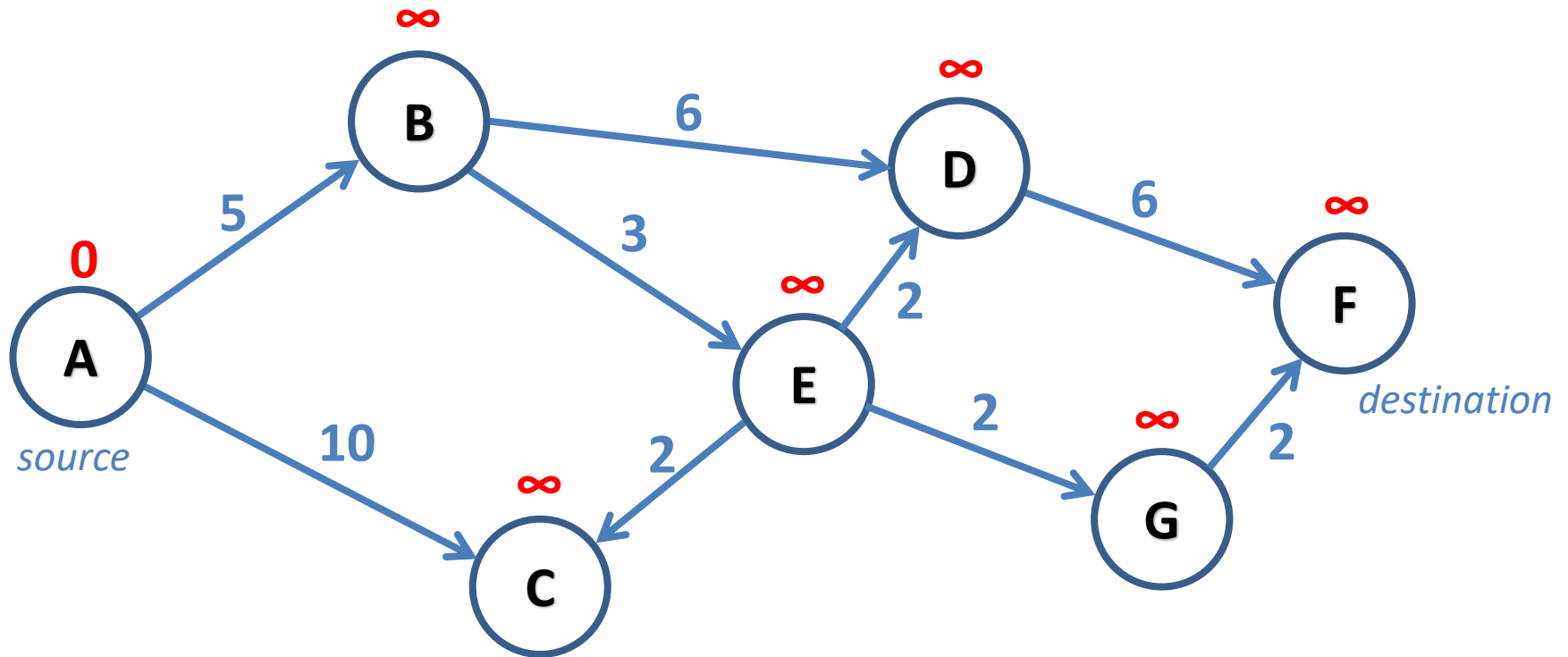
Zet afstandswaarde van startnode op 0 en van alle andere nodes op 'oneindig'

# Stap 1



Plaats alle nodes in lijst 'unvisited' (Q)

## Stap 2

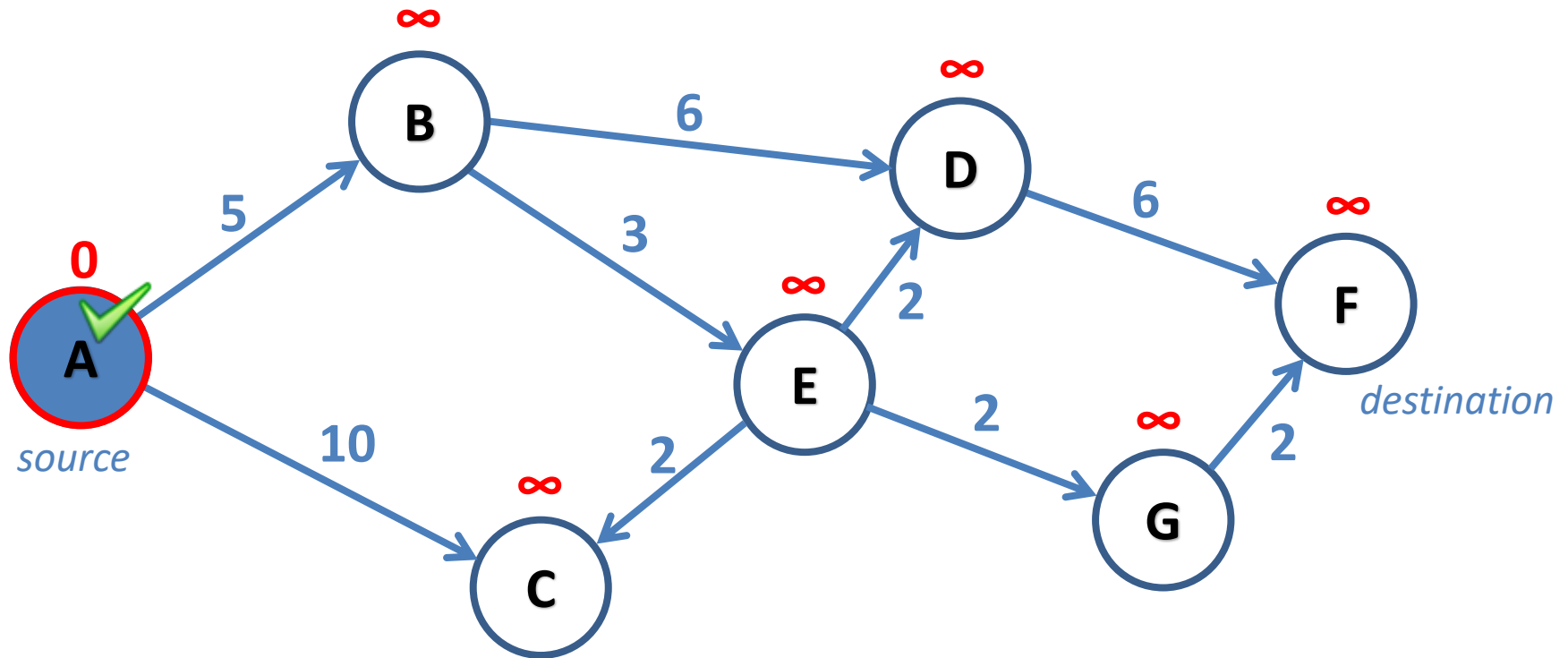


Unvisited list (Q): A,B,C,D,E,F,G



Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$

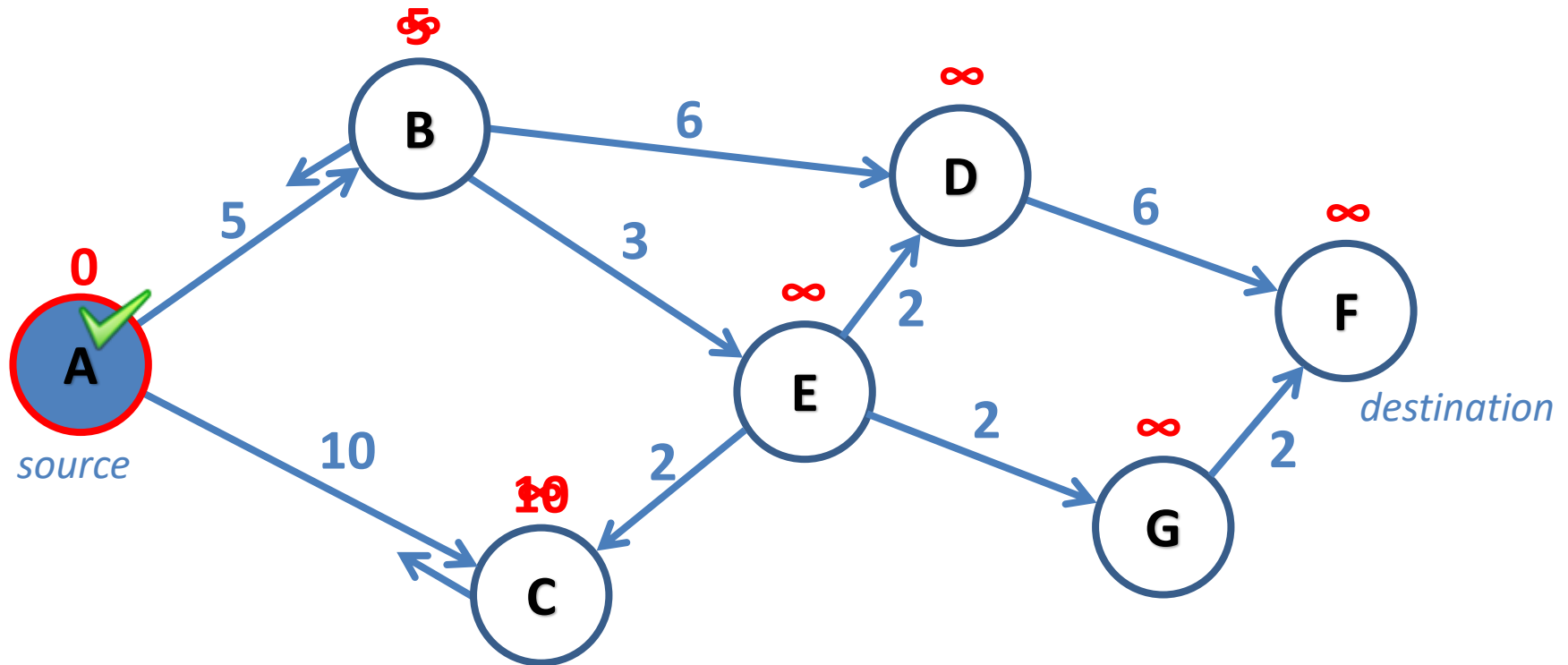
## Stap 3



Unvisited list ( $Q$ ): ~~A~~, ~~B~~, ~~D~~, ~~E~~, ~~G~~, G

# Stap 4

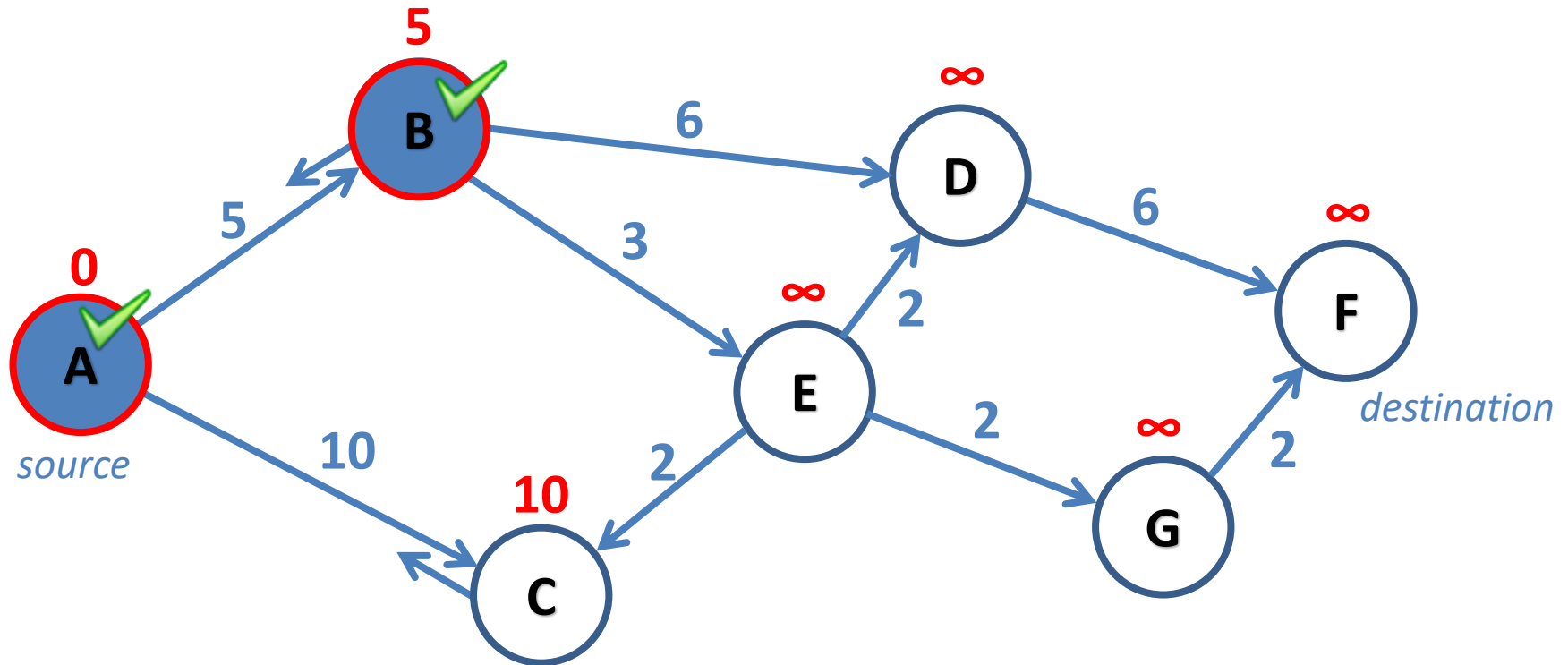
Bepaal van elke (unvisited) neighbour van node  $u$  (uit stap 3) de afstand; als deze kleiner is dan de huidige: update afstandswaarde + set parent (naar  $u$ )



Unvisited list (Q): B,C,D,E,F,G

Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$

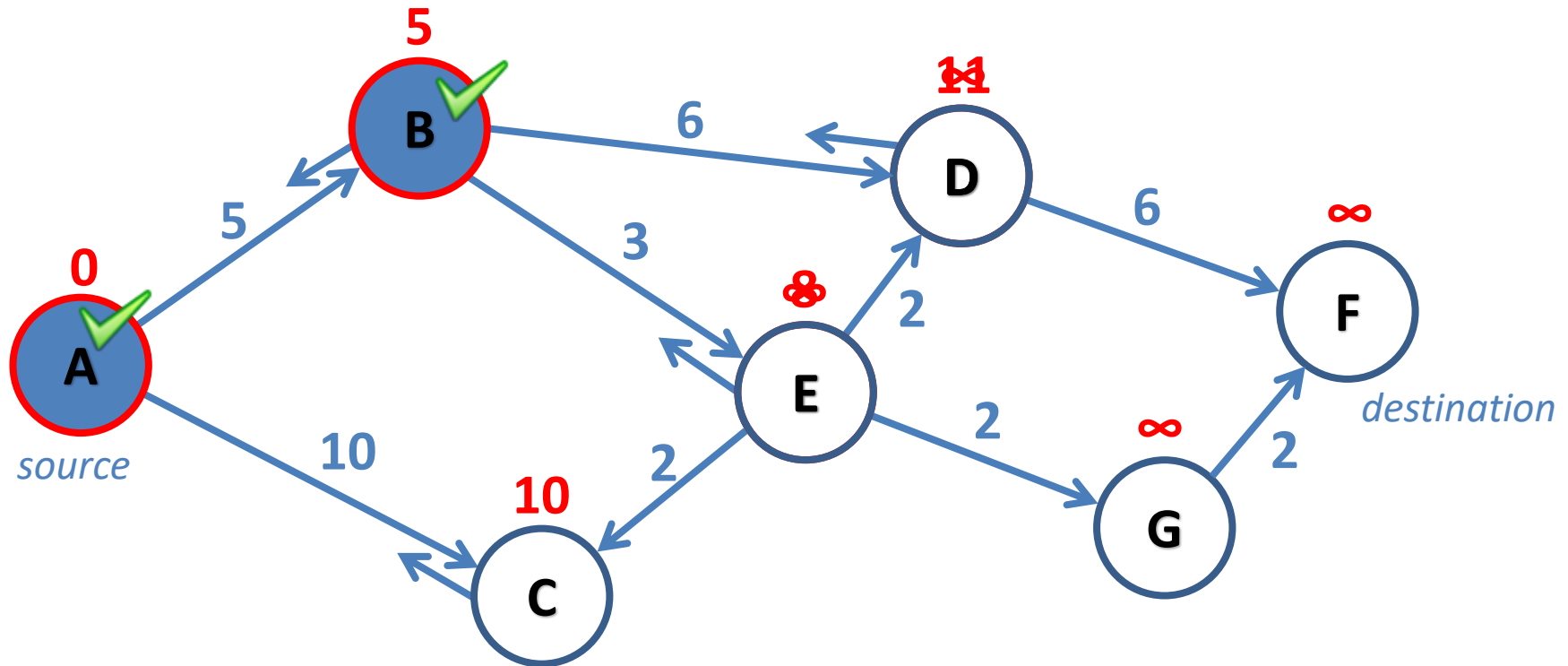
## Stap 3



Unvisited list ( $Q$ ): B,D,E,E,G,G

# Stap 4

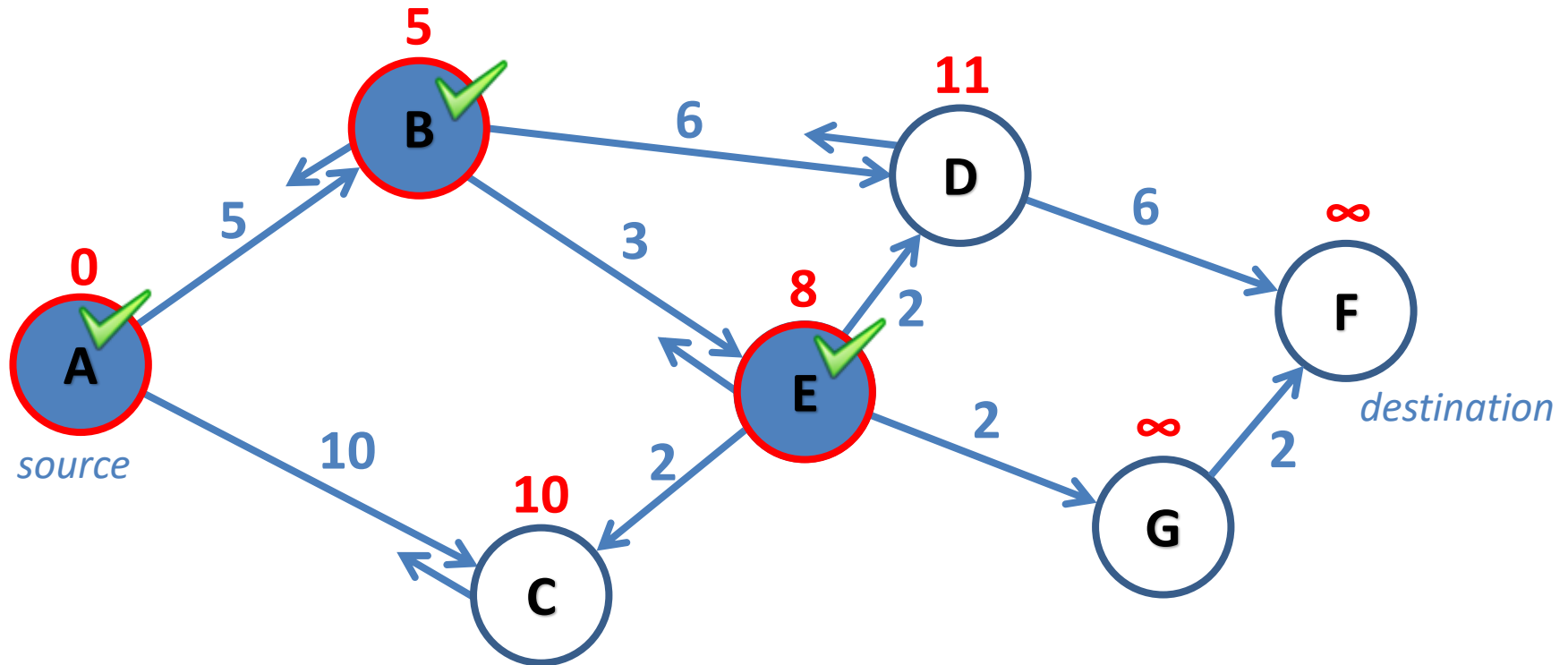
Bepaal van elke (unvisited) neighbour van node  $u$  (uit stap 3) de afstand; als deze kleiner is dan de huidige: update afstandswaarde + set parent (naar  $u$ )



Unvisited list (Q): C,D,E,F,G

Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$

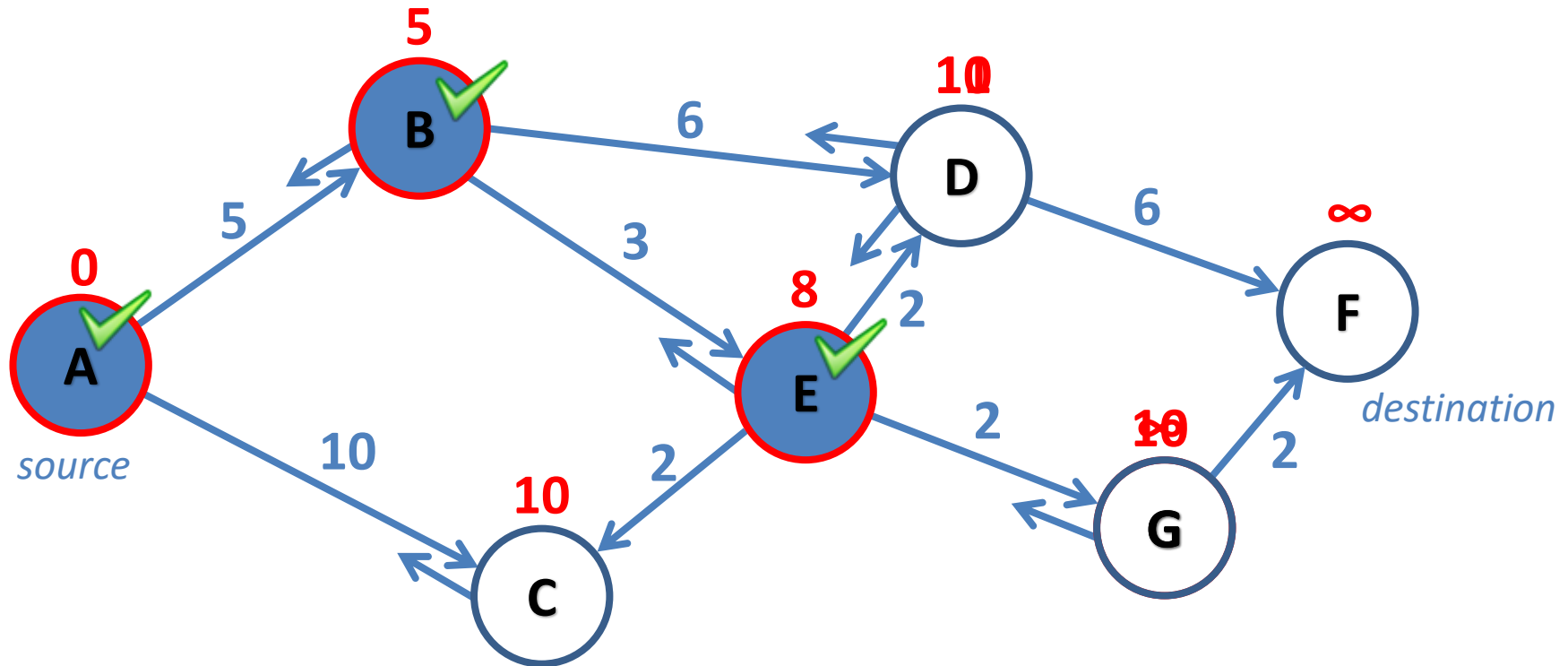
## Stap 3



Unvisited list ( $Q$ ): C,D,E,G

# Stap 4

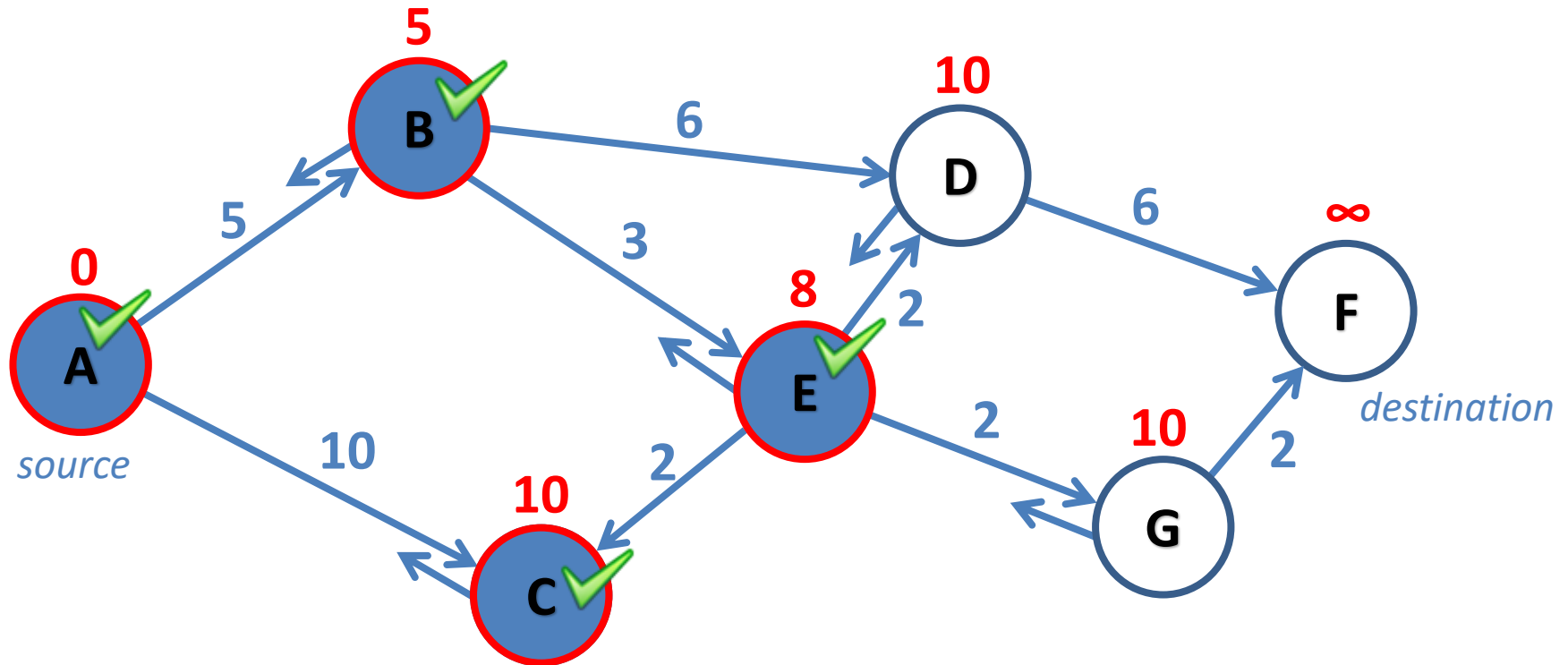
Bepaal van elke (unvisited) neighbour van node  $u$  (uit stap 3) de afstand; als deze kleiner is dan de huidige: update afstandswaarde + set parent (naar  $u$ )



Unvisited list (Q): C,D,F,G

Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$

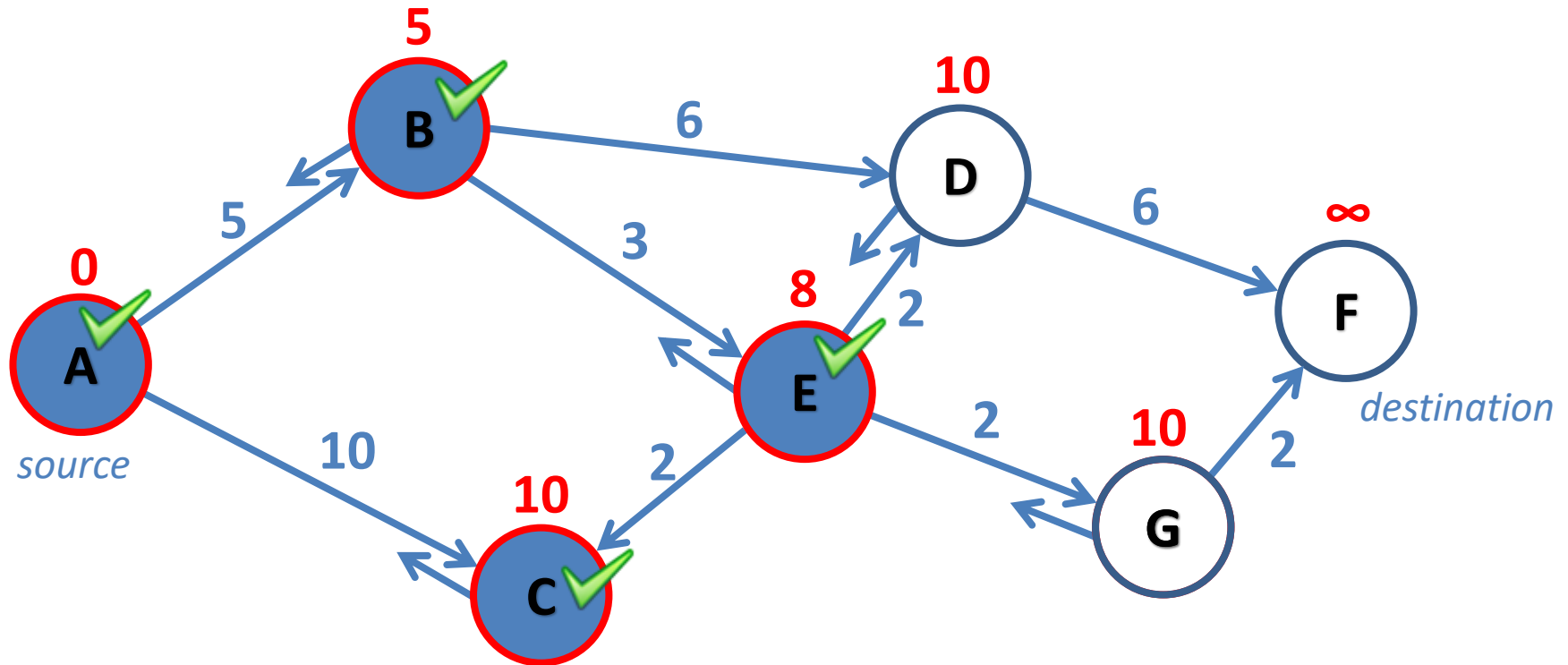
## Stap 3



Unvisited list ( $Q$ ): D, F, G

# Stap 4

Bepaal van elke (unvisited) neighbour van node  $u$  (uit stap 3) de afstand; als deze kleiner is dan de huidige: update afstandswaarde + set parent (naar  $u$ )

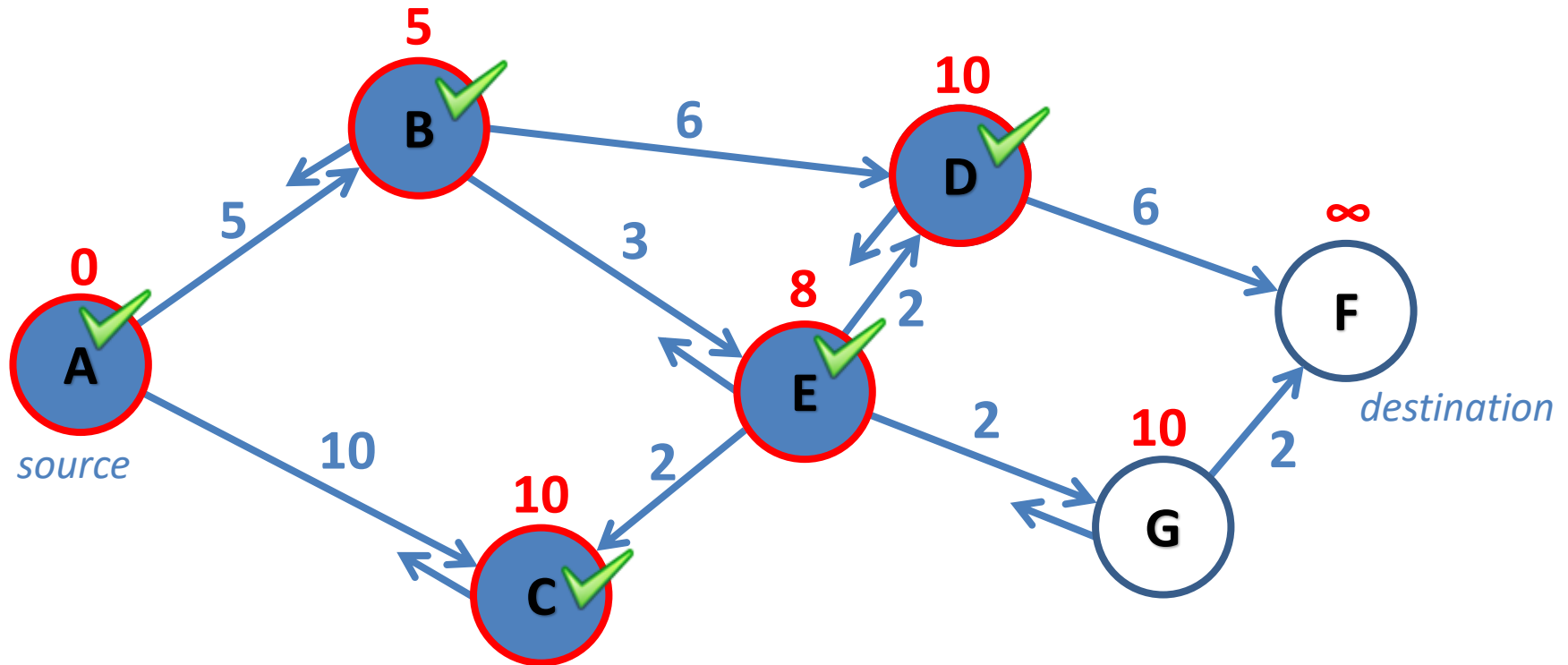


Unvisited list (Q): D,F,G



Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$

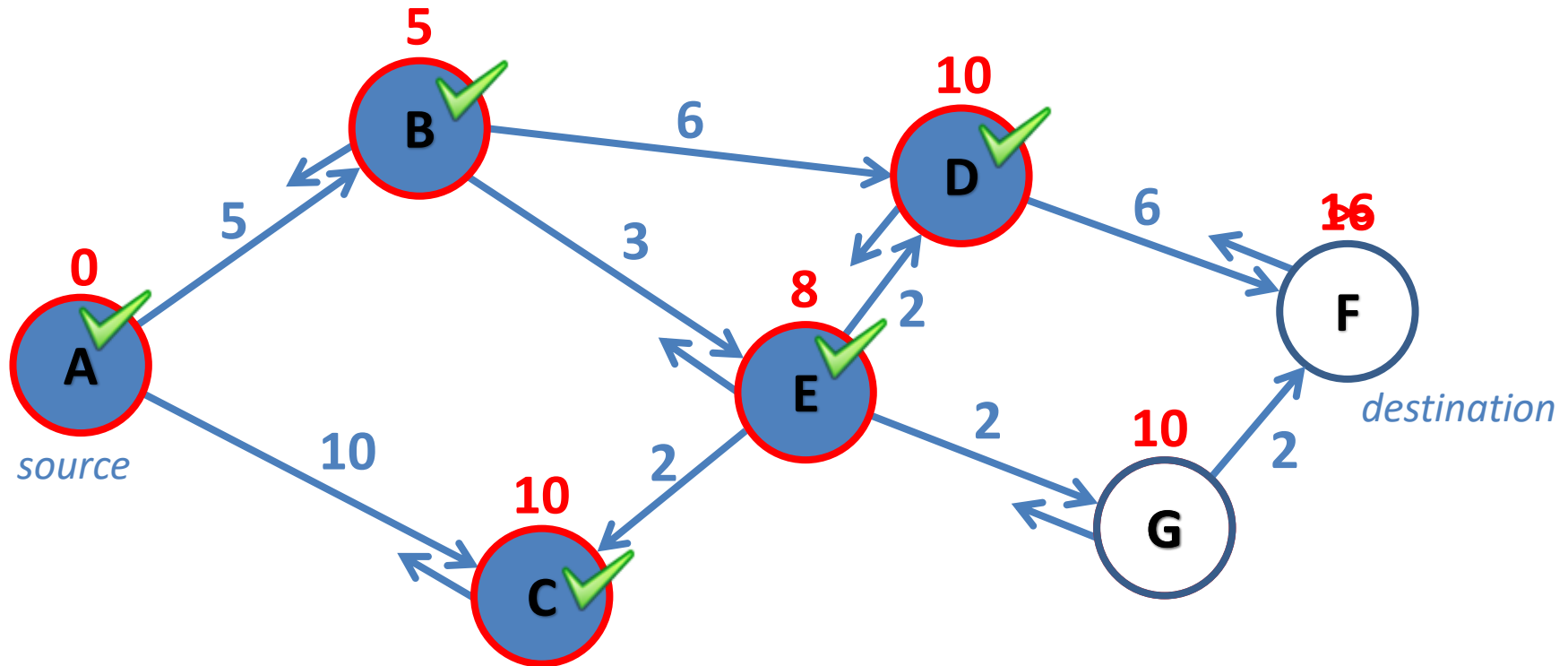
## Stap 3



Unvisited list ( $Q$ ): ~~D~~, ~~E~~, G

# Stap 4

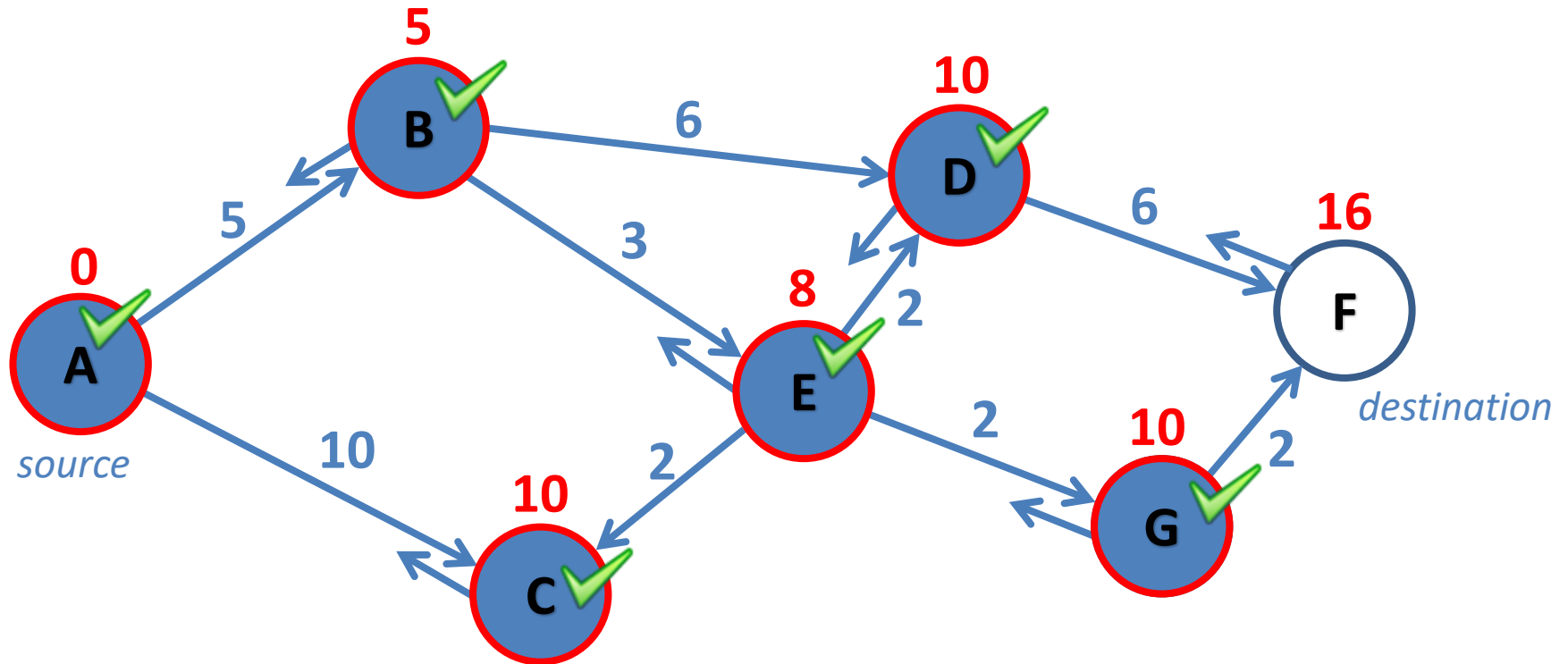
Bepaal van elke (unvisited) neighbour van node  $u$  (uit stap 3) de afstand; als deze kleiner is dan de huidige: update afstandswaarde + set parent (naar  $u$ )



Unvisited list (Q): F,G

Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$

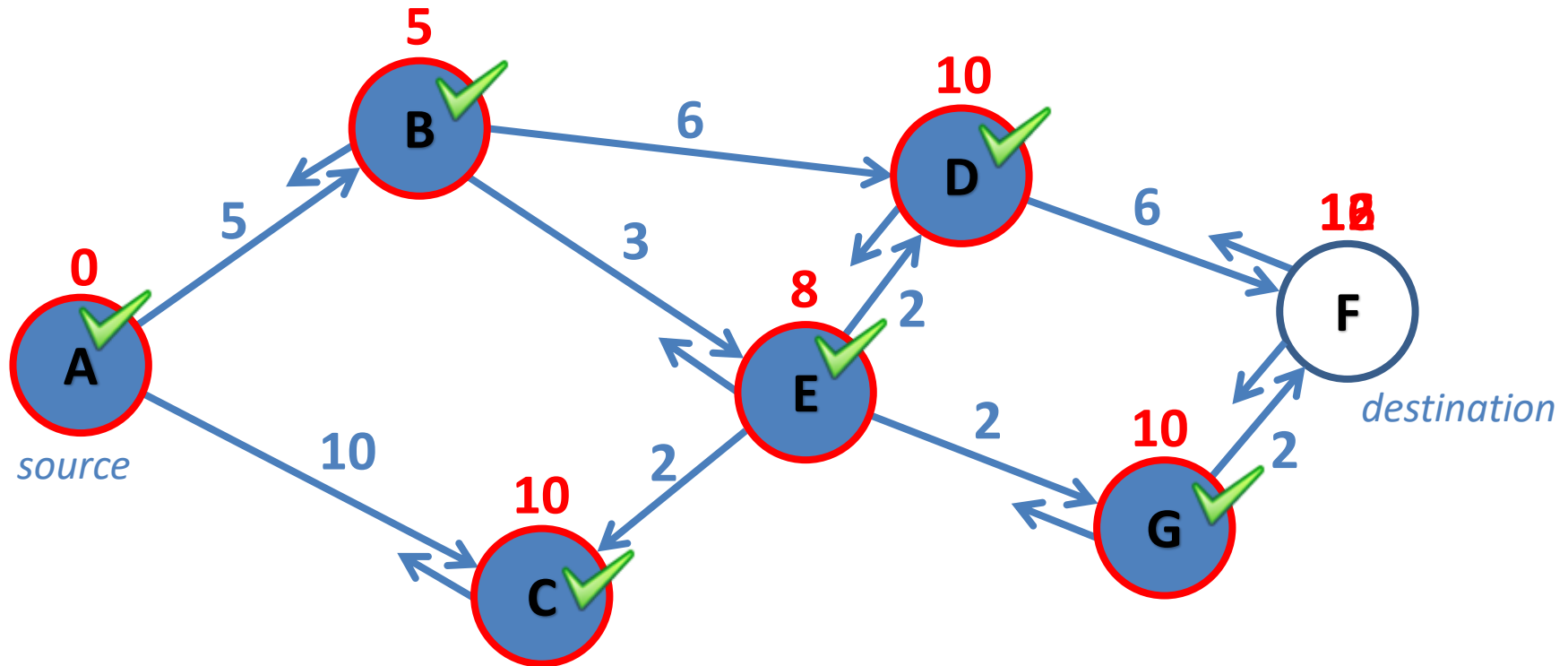
## Stap 3



Unvisited list (Q): F,G

# Stap 4

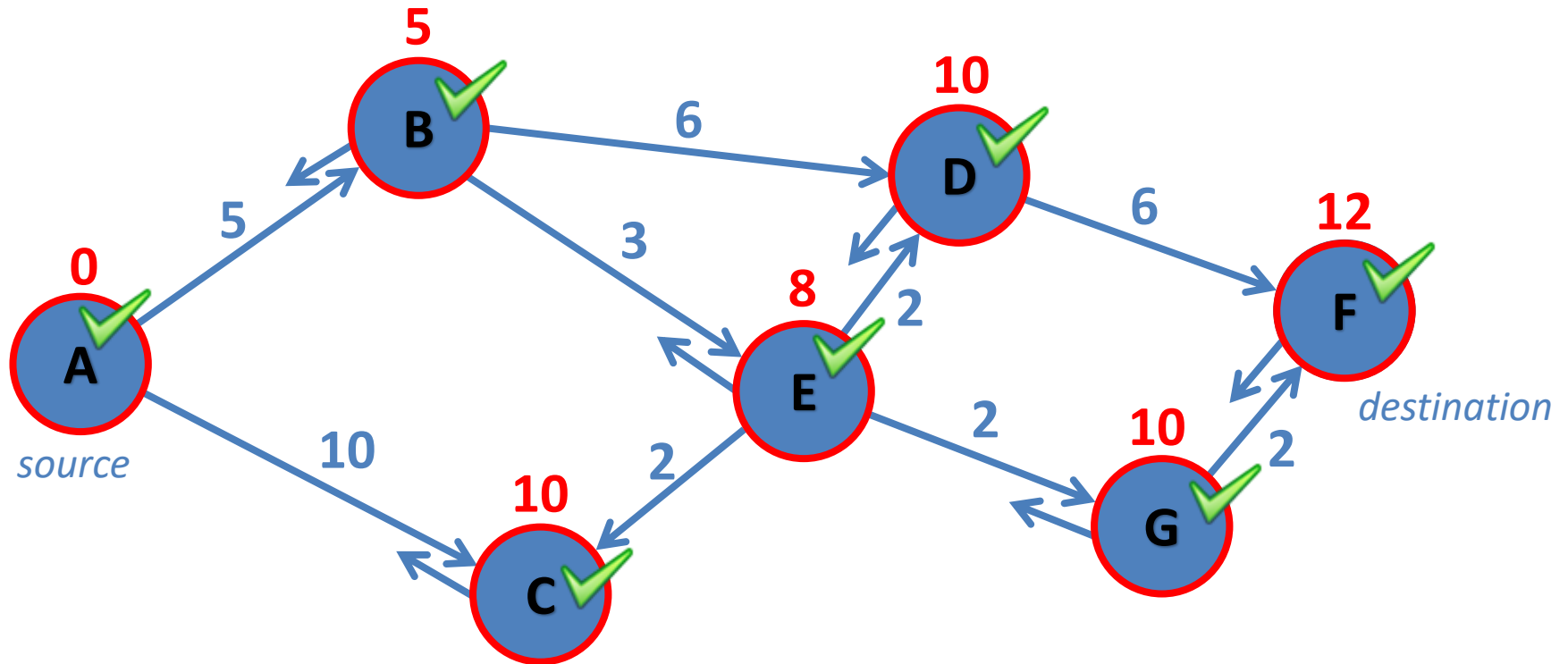
Bepaal van elke (unvisited) neighbour van node  $u$  (uit stap 3) de afstand; als deze kleiner is dan de huidige: update afstandswaarde + set parent (naar  $u$ )



Unvisited list (Q): F

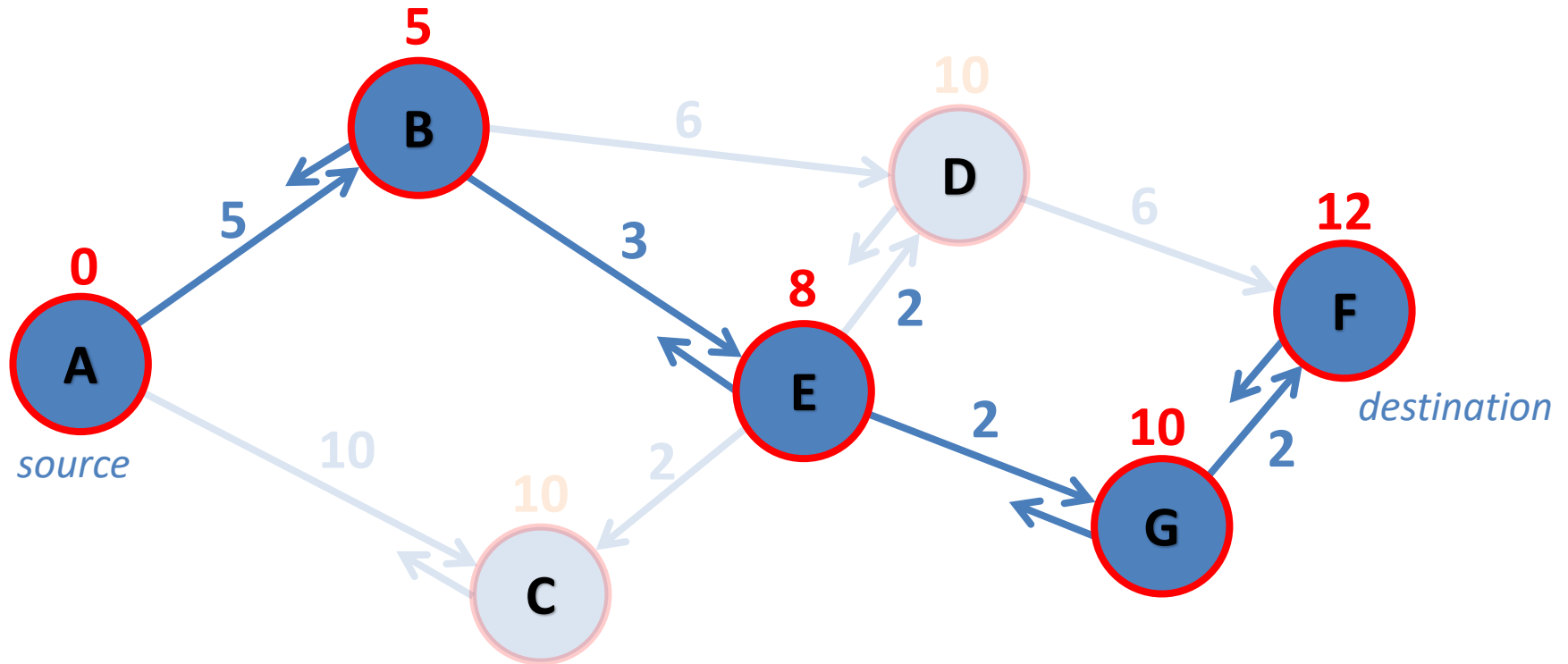
Neem node  $u$  uit  $Q$  met de laagste afstandswaarde (kortste afstand vanaf source) en haal deze uit  $Q$

## Stap 3



Unvisited list (Q): F

# Resultaat

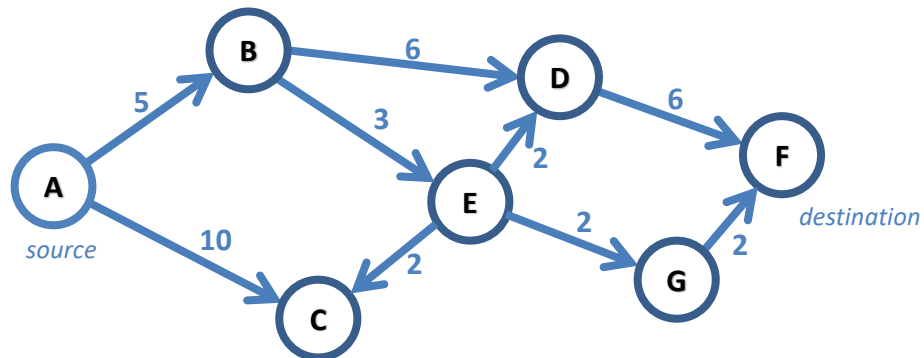


Kortste pad: A-B-E-G-F



# Dijkstra - tabel

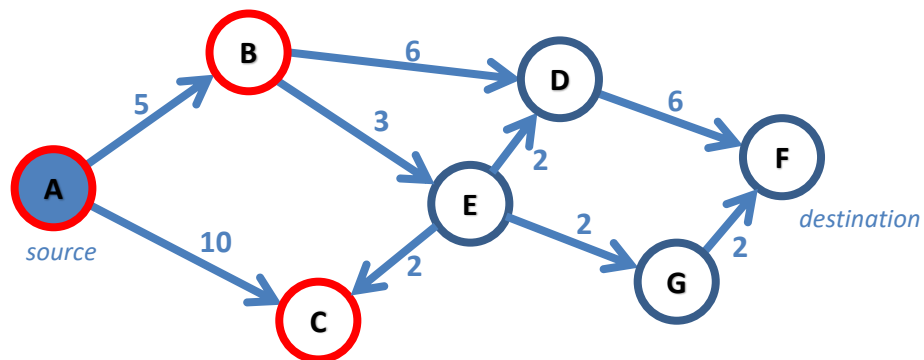
Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1							
2							
3							
4							
5							
6							





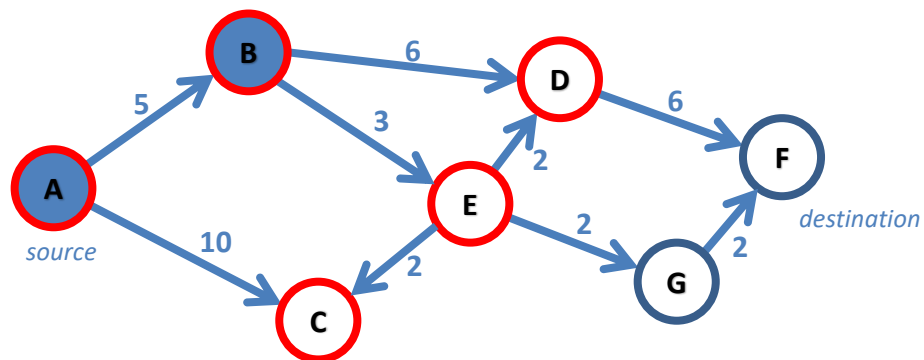
# Dijkstra - tabel

Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	5	10	$\infty$	$\infty$	$\infty$	$\infty$
2							
3							
4							
5							
6							



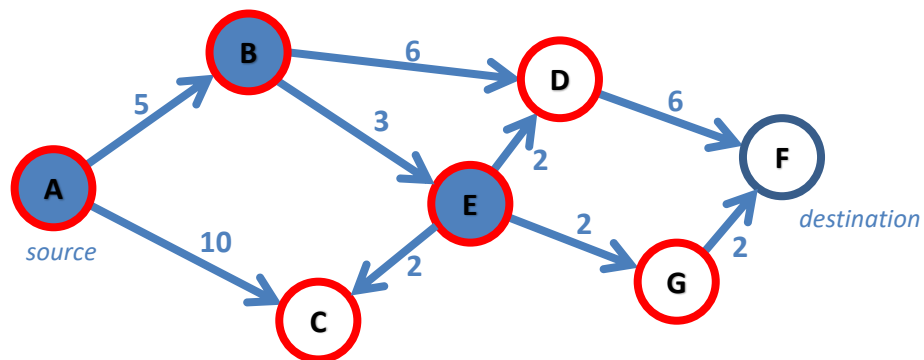
# Dijkstra - tabel

Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	5	10	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	10	11	8	$\infty$	$\infty$
3							
4							
5							
6							



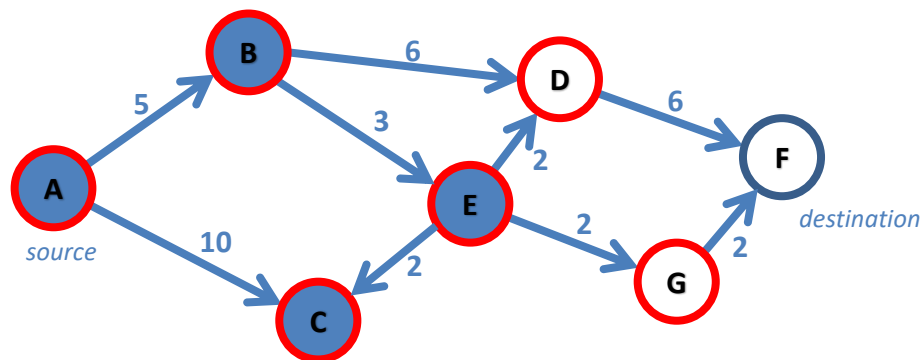
# Dijkstra - tabel

Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	5	10	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	10	11	8	$\infty$	$\infty$
3			10	10	✓	$\infty$	10
4							
5							
6							



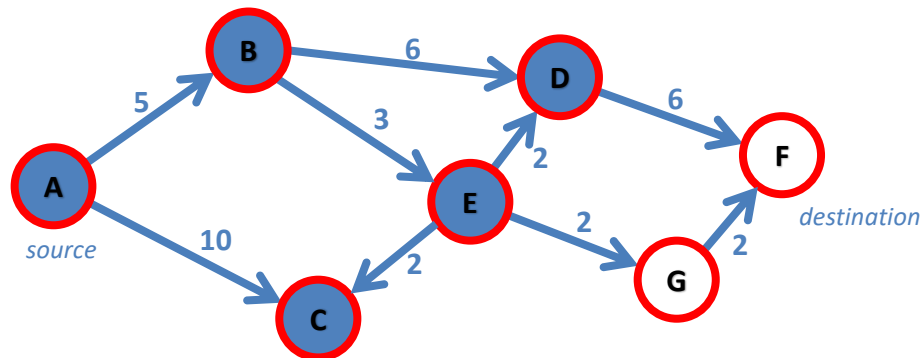
# Dijkstra - tabel

Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	5	10	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	10	11	8	$\infty$	$\infty$
3			10	10	✓	$\infty$	10
4			✓	10		$\infty$	10
5							
6							



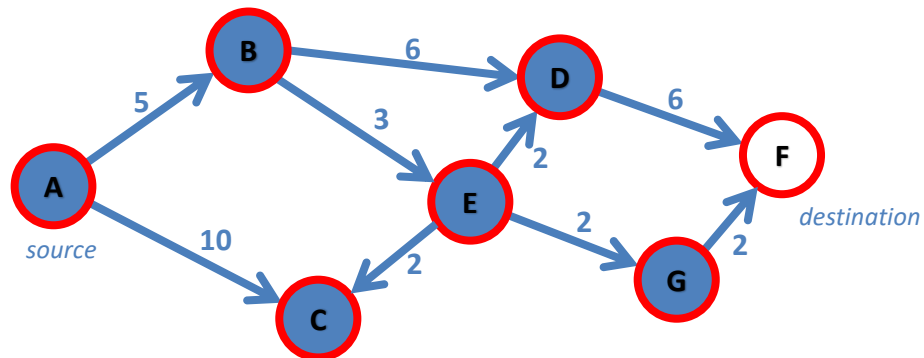
# Dijkstra - tabel

Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	5	10	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	10	11	8	$\infty$	$\infty$
3			10	10	✓	$\infty$	10
4			✓	10		$\infty$	10
5				✓		16	10
6							



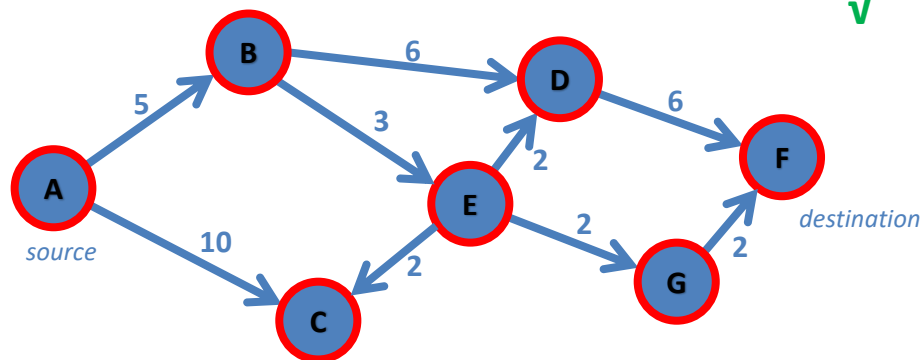
# Dijkstra - tabel

Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	5	10	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	10	11	8	$\infty$	$\infty$
3			10	10	✓	$\infty$	10
4			✓	10		$\infty$	10
5				✓		16	10
6						12	✓

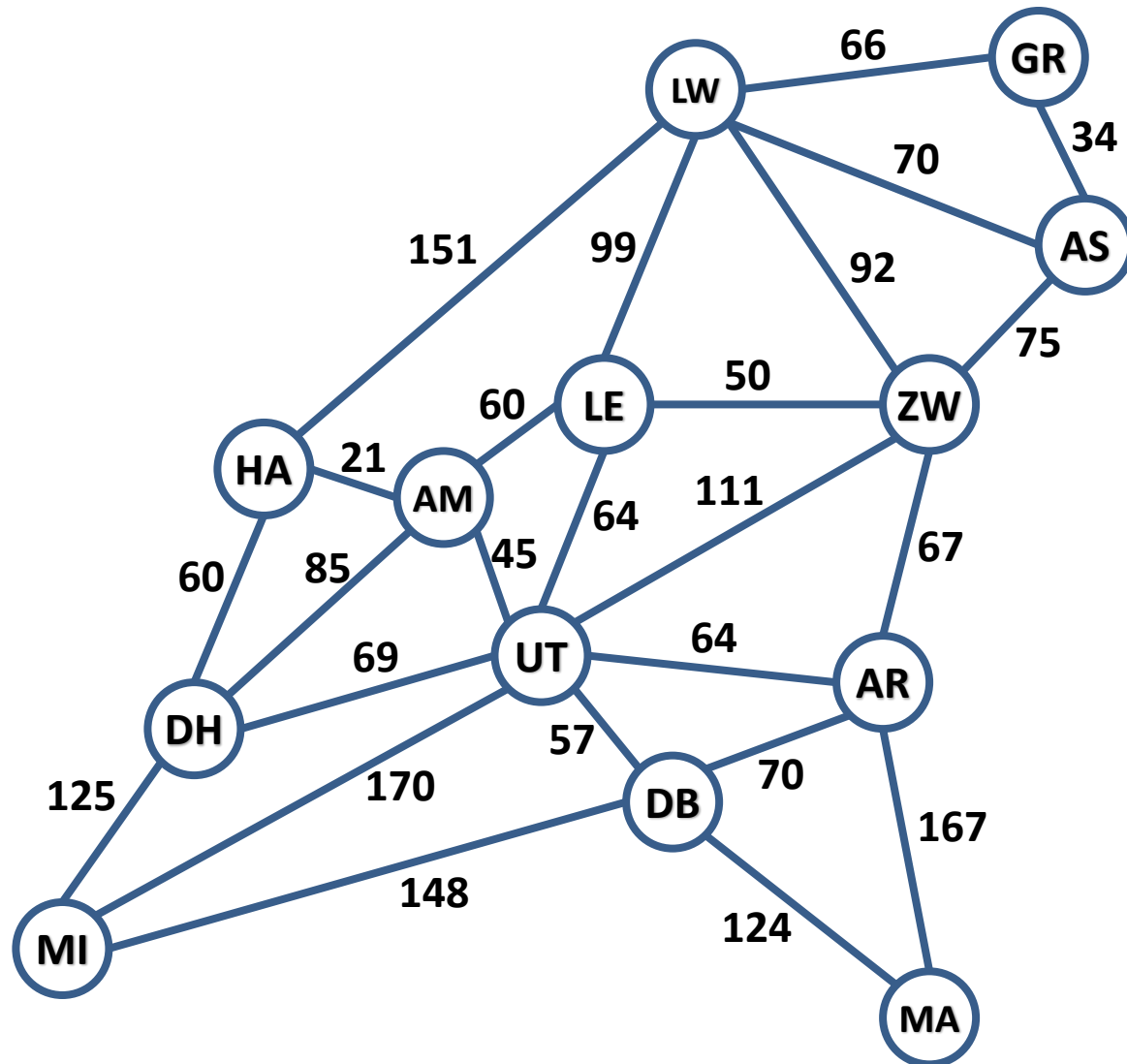


# Dijkstra - tabel

Stap	A	B	C	D	E	F	G
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	5	10	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	10	11	8	$\infty$	$\infty$
3			10	10	✓	$\infty$	10
4			✓	10		$\infty$	10
5				✓		16	10
6						12	✓



Bepaal kortste pad van MI  $\rightarrow$  ZW





# Uitwerking (MI $\rightarrow$ zw)

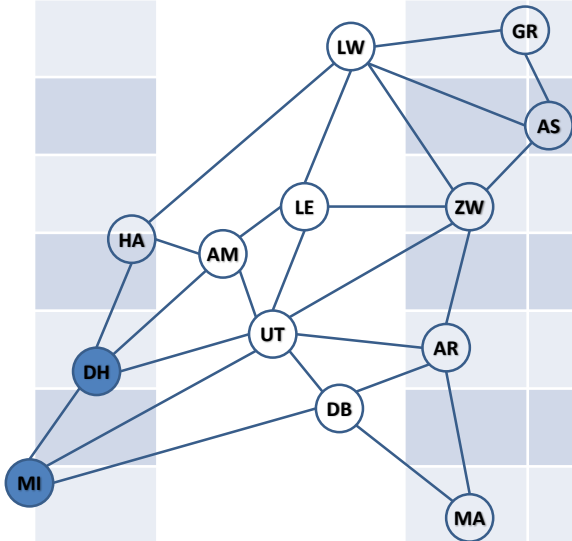
[illegible]

# Uitwerking (MI $\rightarrow$ zw)

[illegible]

# Uitwerking (MI $\rightarrow$ ZW)

Stp	MI	DH	DB	MA	HA	AM	UT	AR	LE	ZW	AS	LW	GR
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	125	148	$\infty$	$\infty$	$\infty$	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	148	$\infty$	185	210	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

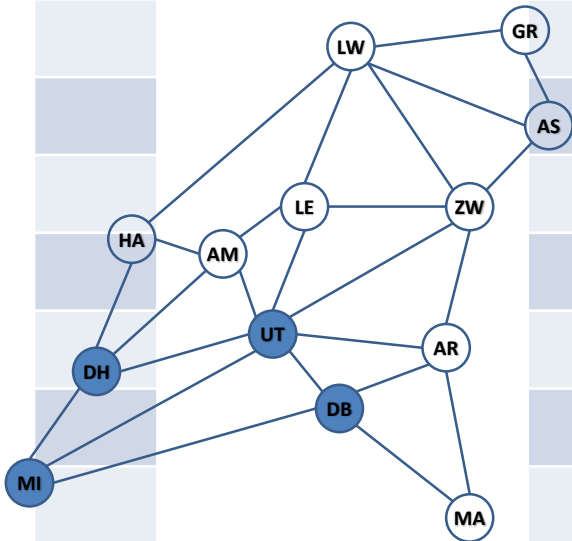


# Uitwerking (MI $\rightarrow$ zw)

[illegible]

# Uitwerking (MI $\rightarrow$ ZW)

Stp	MI	DH	DB	MA	HA	AM	UT	AR	LE	ZW	AS	LW	GR
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	125	148	$\infty$	$\infty$	$\infty$	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	148	$\infty$	185	210	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
3			✓	272	185	210	170	218	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
4				272	185	210	✓	218	234	281	$\infty$	$\infty$	$\infty$



# Uitwerking (MI $\rightarrow$ zw)

[illegible]

# Uitwerking (MI $\rightarrow$ zw)

[illegible]

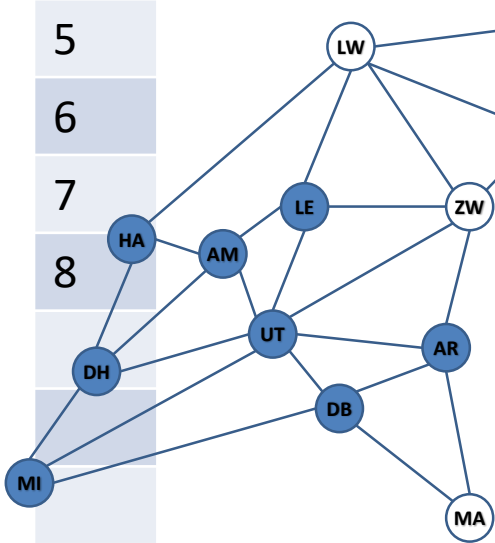
# Uitwerking (MI $\rightarrow$ zw)

[illegible]



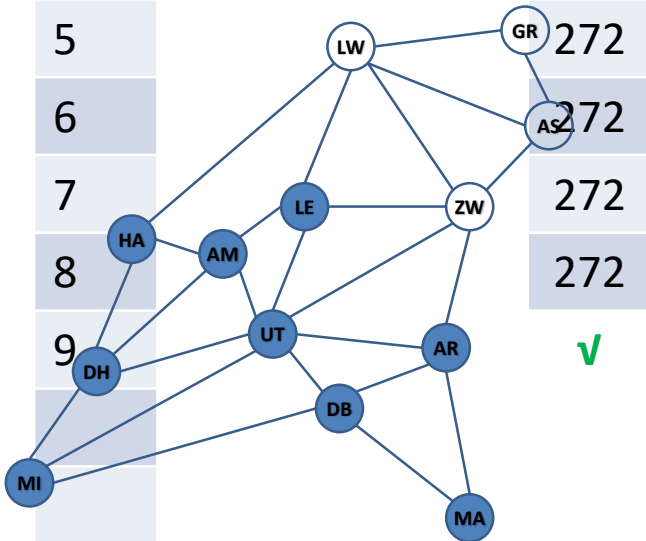
# Uitwerking (MI $\rightarrow$ ZW)

Stp	MI	DH	DB	MA	HA	AM	UT	AR	LE	ZW	AS	LW	GR
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	125	148	$\infty$	$\infty$	$\infty$	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	148	$\infty$	185	210	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
3			✓	272	185	210	170	218	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
4				272	185	210	✓	218	234	281	$\infty$	$\infty$	$\infty$
5				272	✓	206		218	234	281	$\infty$	336	$\infty$
6				272		✓		218	234	281	$\infty$	336	$\infty$
7				272				✓	234	281	$\infty$	336	$\infty$
8				272					✓	281	$\infty$	333	$\infty$



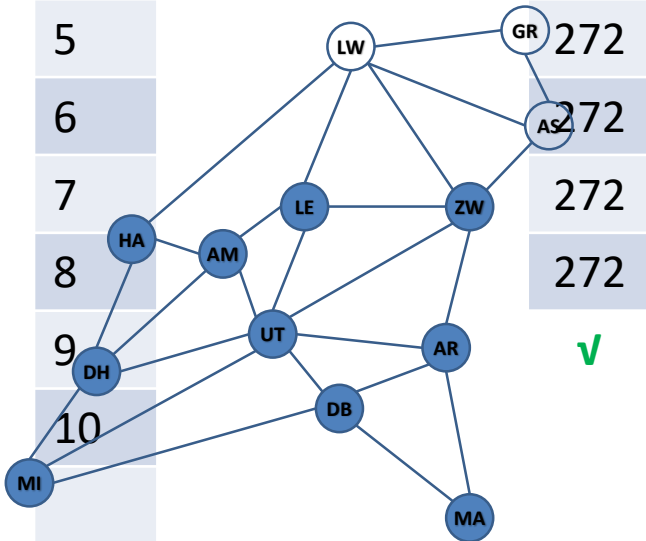
# Uitwerking (MI $\rightarrow$ ZW)

Stp	MI	DH	DB	MA	HA	AM	UT	AR	LE	ZW	AS	LW	GR
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	125	148	$\infty$	$\infty$	$\infty$	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	148	$\infty$	185	210	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
3			✓	272	185	210	170	218	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
4				272	185	210	✓	218	234	281	$\infty$	$\infty$	$\infty$
5				272	✓	206		218	234	281	$\infty$	336	$\infty$
6				272		✓		218	234	281	$\infty$	336	$\infty$
7				272				✓	234	281	$\infty$	336	$\infty$
8				272					✓	281	$\infty$	333	$\infty$
9				✓						281	$\infty$	333	$\infty$



# Uitwerking (MI $\rightarrow$ ZW)

Stp	MI	DH	DB	MA	HA	AM	UT	AR	LE	ZW	AS	LW	GR
0	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1	✓	125	148	$\infty$	$\infty$	$\infty$	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
2		✓	148	$\infty$	185	210	170	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
3			✓	272	185	210	170	218	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
4				272	185	210	✓	218	234	281	$\infty$	$\infty$	$\infty$
5				272	✓	206		218	234	281	$\infty$	336	$\infty$
6				272		✓		218	234	281	$\infty$	336	$\infty$
7								✓	234	281	$\infty$	336	$\infty$
8									✓	281	$\infty$	333	$\infty$
9										281	$\infty$	333	$\infty$
10										✓			



# Dijkstra's algoritme

-	MI	DH	DB	MA	HA	AM	UT	AR	LE	ZW	AS	LW	GR
MI	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
MI	0	125	148?	∞	∞	∞	170?	∞	∞	∞	∞	∞	∞
MI	0	125	148	∞	185?	210?	170?	∞	∞	∞	∞	∞	∞
MI	0	125	148	272?	185?	210?	170	218?	∞	∞	∞	∞	∞
MI	0	125	148	272?	185	210?	170	218?	234?	281?	∞	∞	∞
MI	0	125	148	272?	185	206	170	218?	234?	281?	∞	336?	∞
MI	0	125	148	272?	185	206	170	218	234?	281?	∞	336?	∞
MI	0	125	148	272?	185	206	170	218	234	281?	∞	336?	∞
MI	0	125	148	272	185	206	170	218	234	281?	∞	333?	∞
MI	0	125	148	272	185	206	170	218	234	281	∞	333?	∞
MI	0	125	148	272	185	206	170	218	234	281	356?	333	∞
MI	0	125	148	272	185	206	170	218	234	281	356	333	399?
MI	0	125	148	272	185	206	170	218	234	281	356	333	390

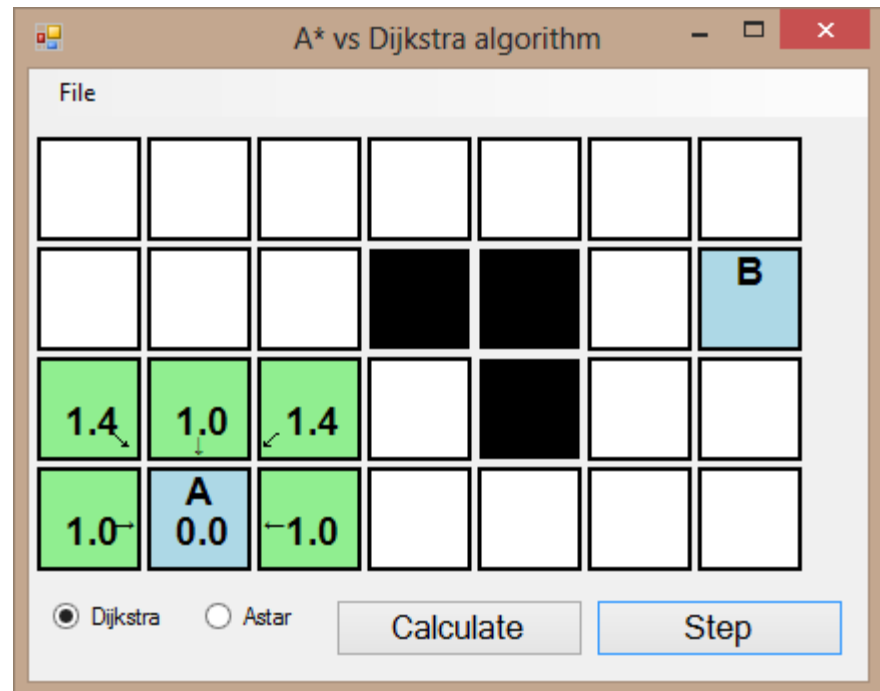
Calculate



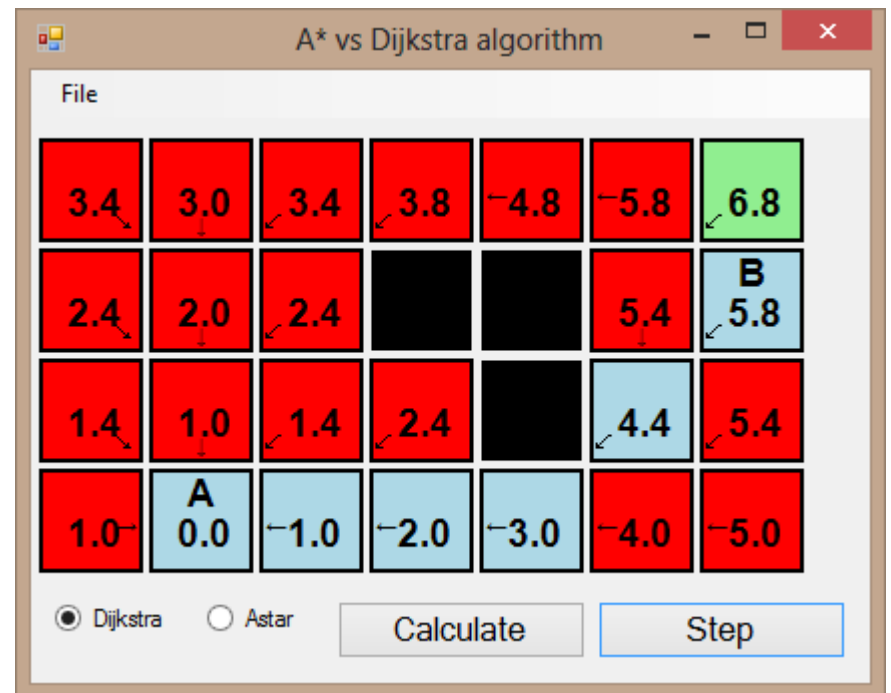
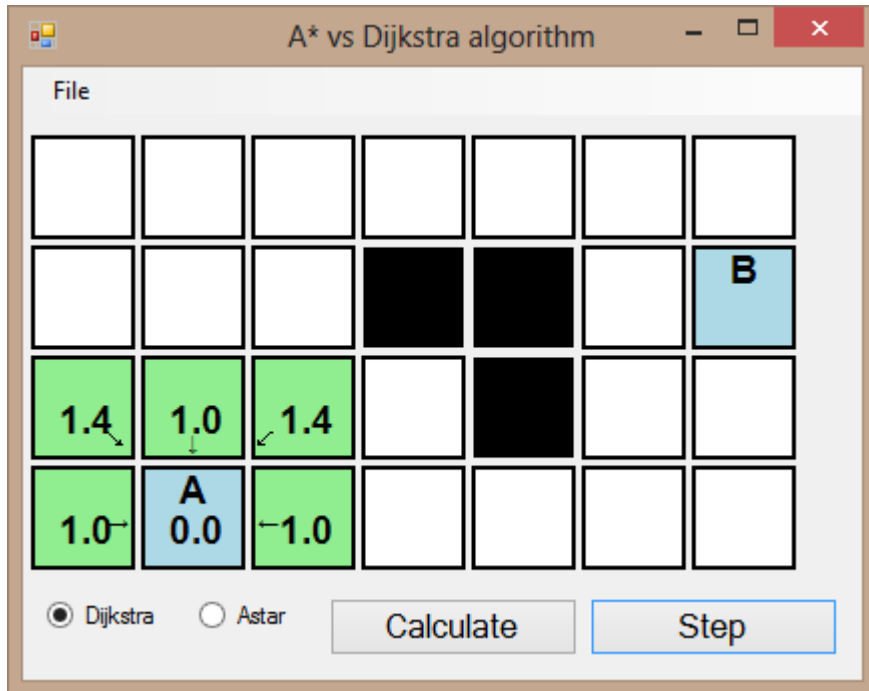
# Grid ipv graph

- Dijkstra algoritme ook toepasbaar op een grid
- Wat zijn de knopen? Wat de paden?
- Wat zijn de afstanden tussen de knopen?

- Afstanden:
  - recht pad: 1
  - schuin pad:  $\sqrt{2}$



# Demo



# Wat is hier het nadeel?





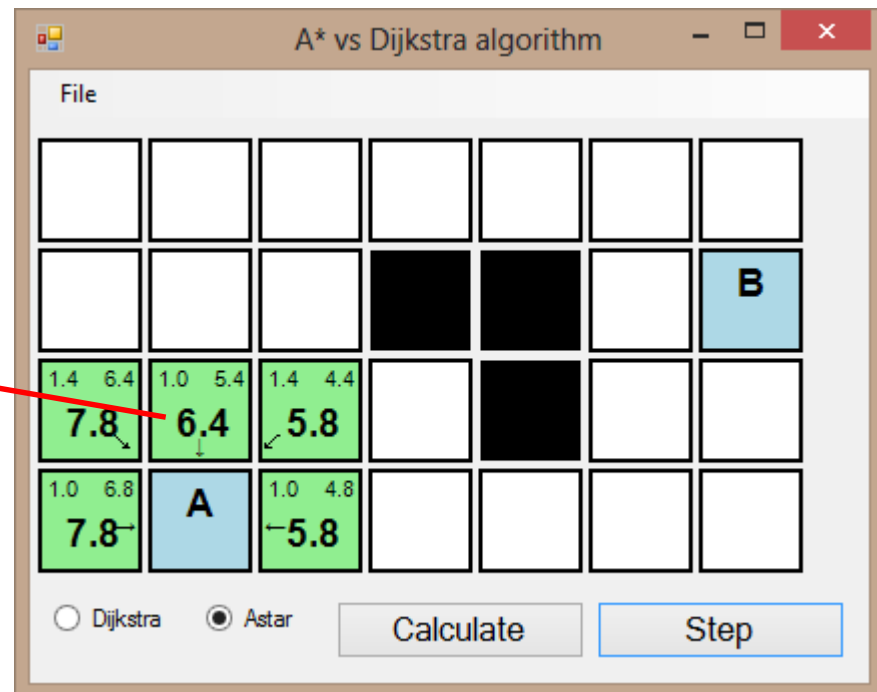
# A\* (search) algoritme

- Ipv alleen maar rekening te houden met de afgelegde afstand, kan er uiteraard ook rekening worden gehouden met de afstand naar het doel (mits je dat weet uiteraard)
- Bij een grid is dit makkelijker te bepalen dan bij een graaf...
- G cost = distance from starting node
- H cost = distance from end node
- F cost = G cost + H cost
- Kies steeds de volgende cel met de laagste F cost en update de neighbours van deze cel

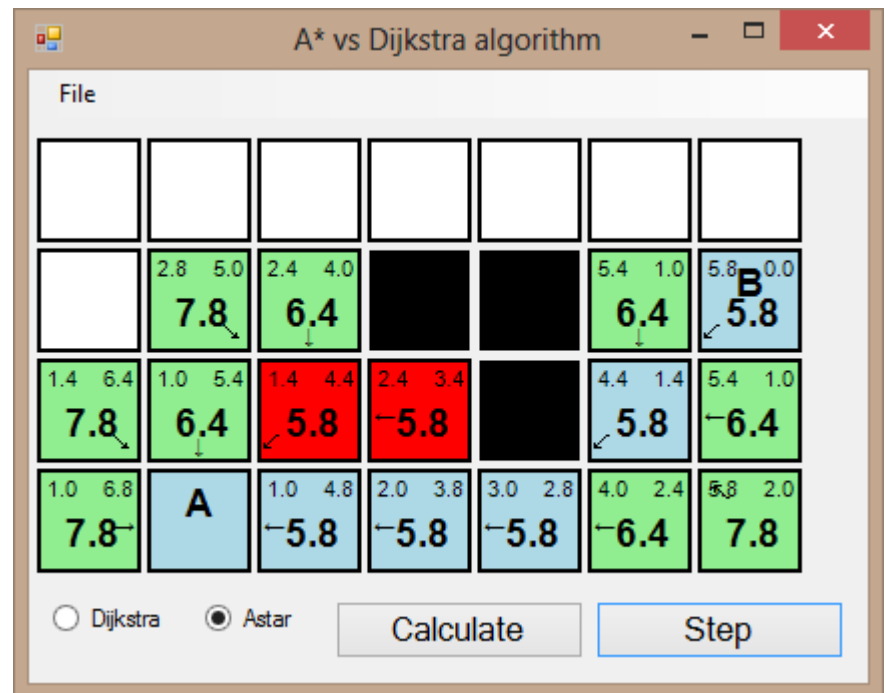
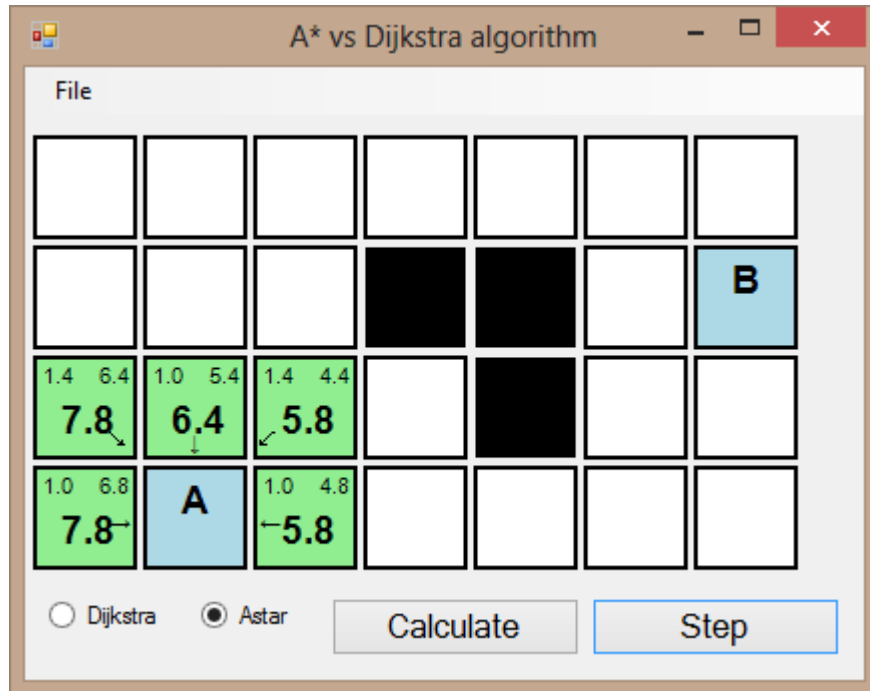
# Voorbeeld

- G cost = distance from starting node
- H cost = distance from end node
- F cost = G cost + H cost

- G cost = 1
- H cost = 5.4
- F cost = 6.4



# Demo



# Beter resultaat (dan Dijkstra)

