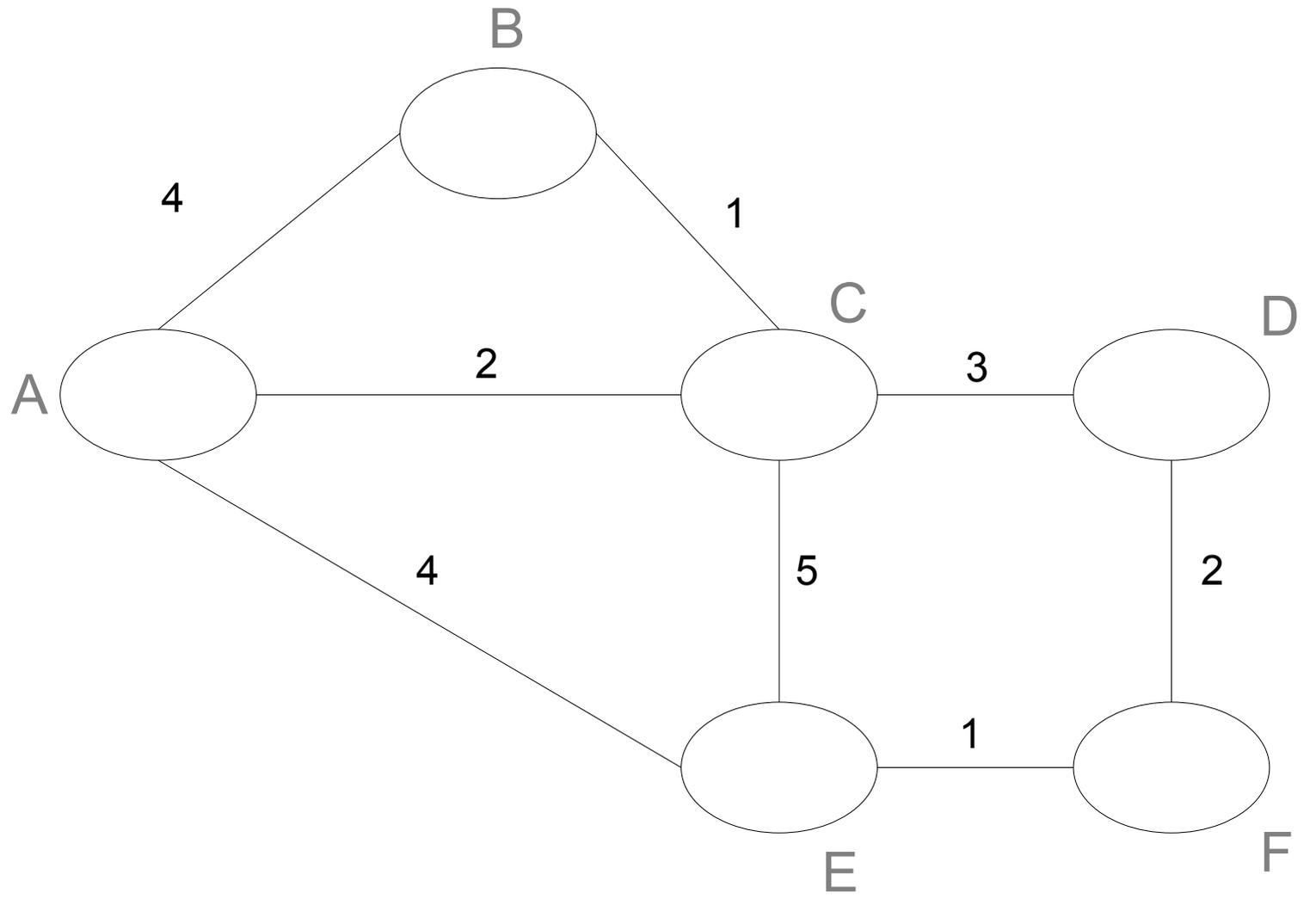


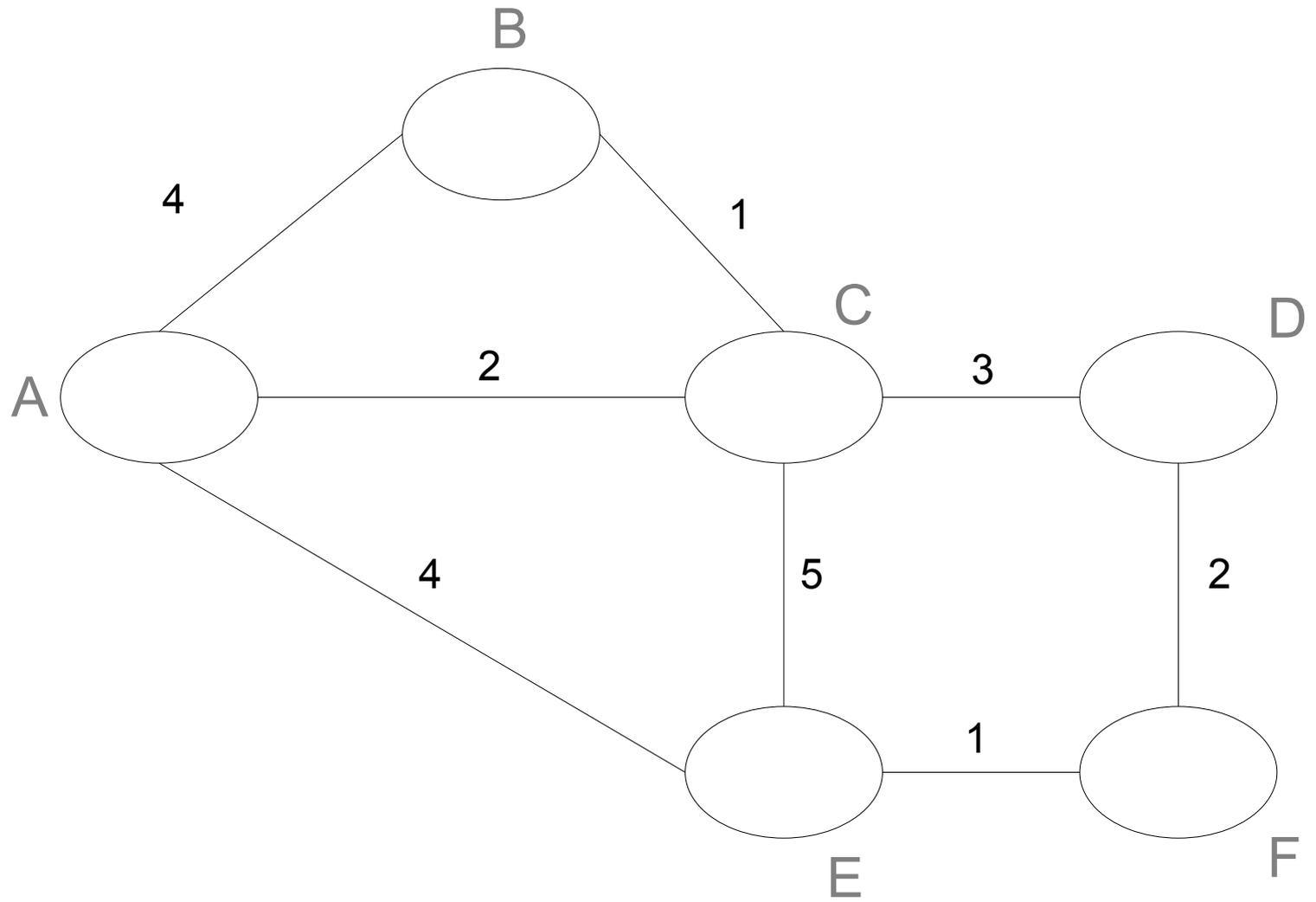
# Der Dijkstra-Algorithmus

Von einem gegebenen Startknoten aus:  
die kürzesten Wege zu allen anderen Knoten

A		
B		
C		
D		
E		
F		



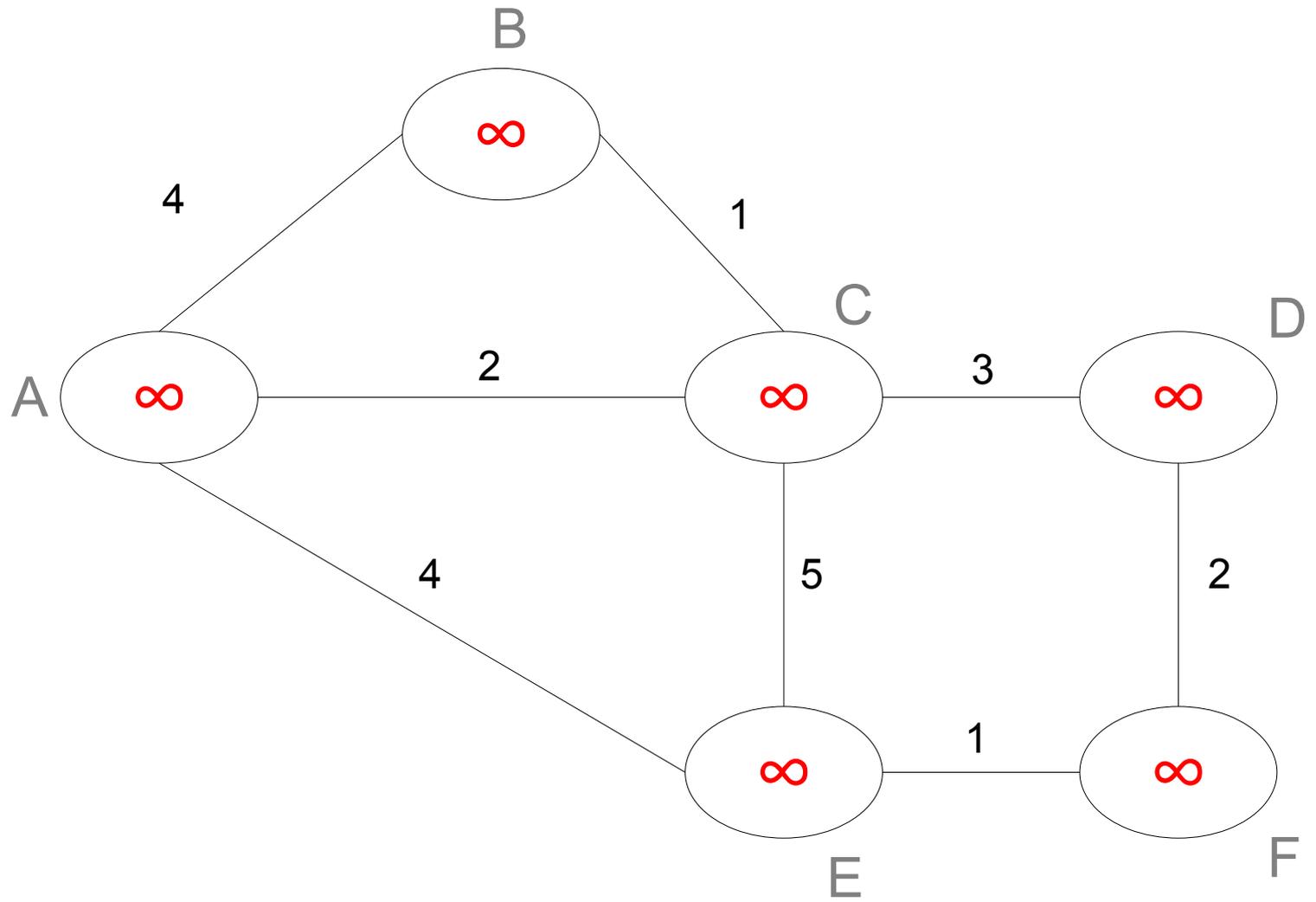
A		
B		
C		
D		
E		
F		



Vorarbeit 1:

- setze jeden Knoten als unbesucht
- setze jede Distanz auf unendlich
- setze jeden Vorgänger auf *null*

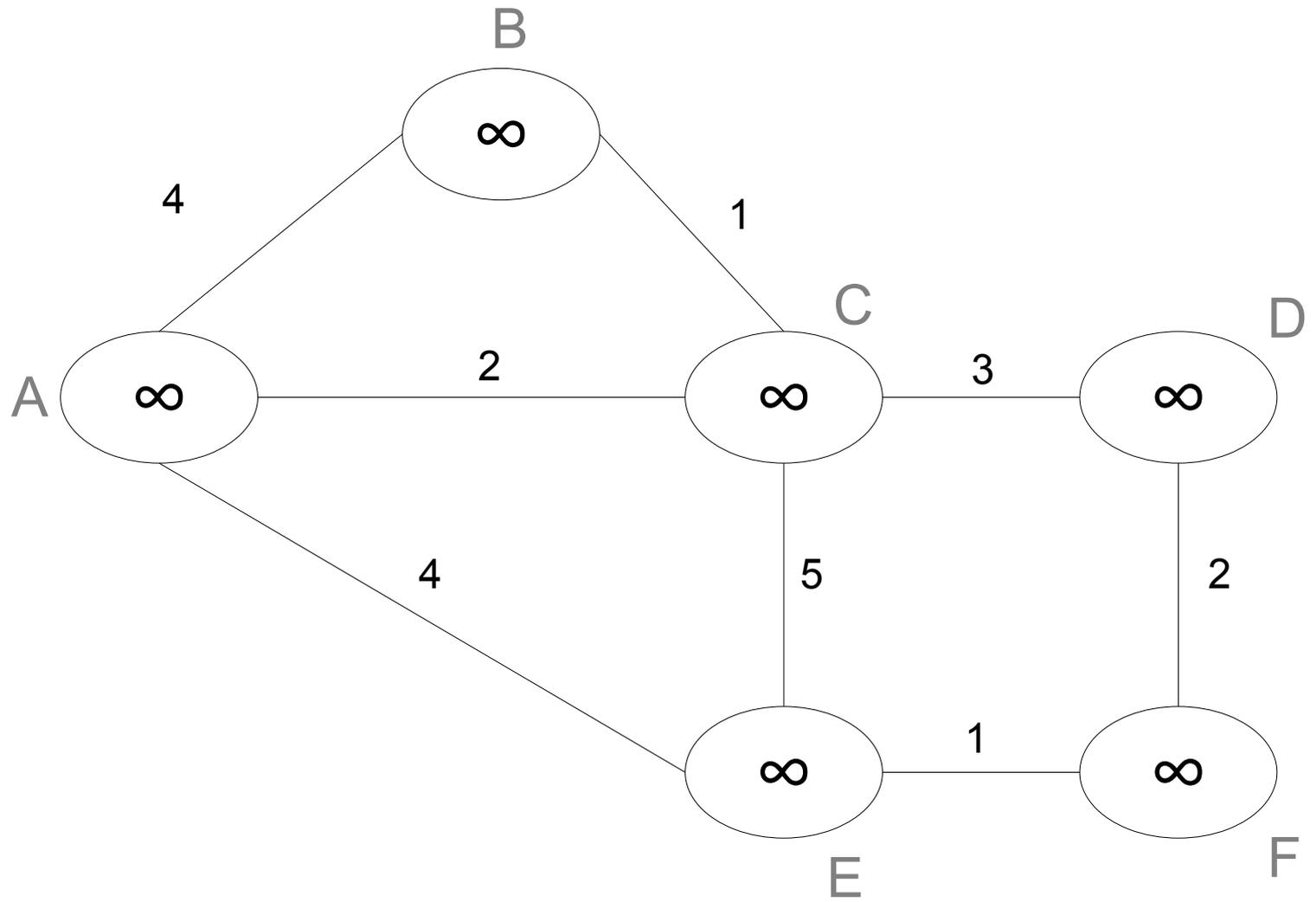
A	$\infty$	
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	
F	$\infty$	



### Vorarbeit 1:

- setze jeden Knoten als unbesucht
- setze jede Distanz auf unendlich
- setze jeden Vorgänger auf *null*

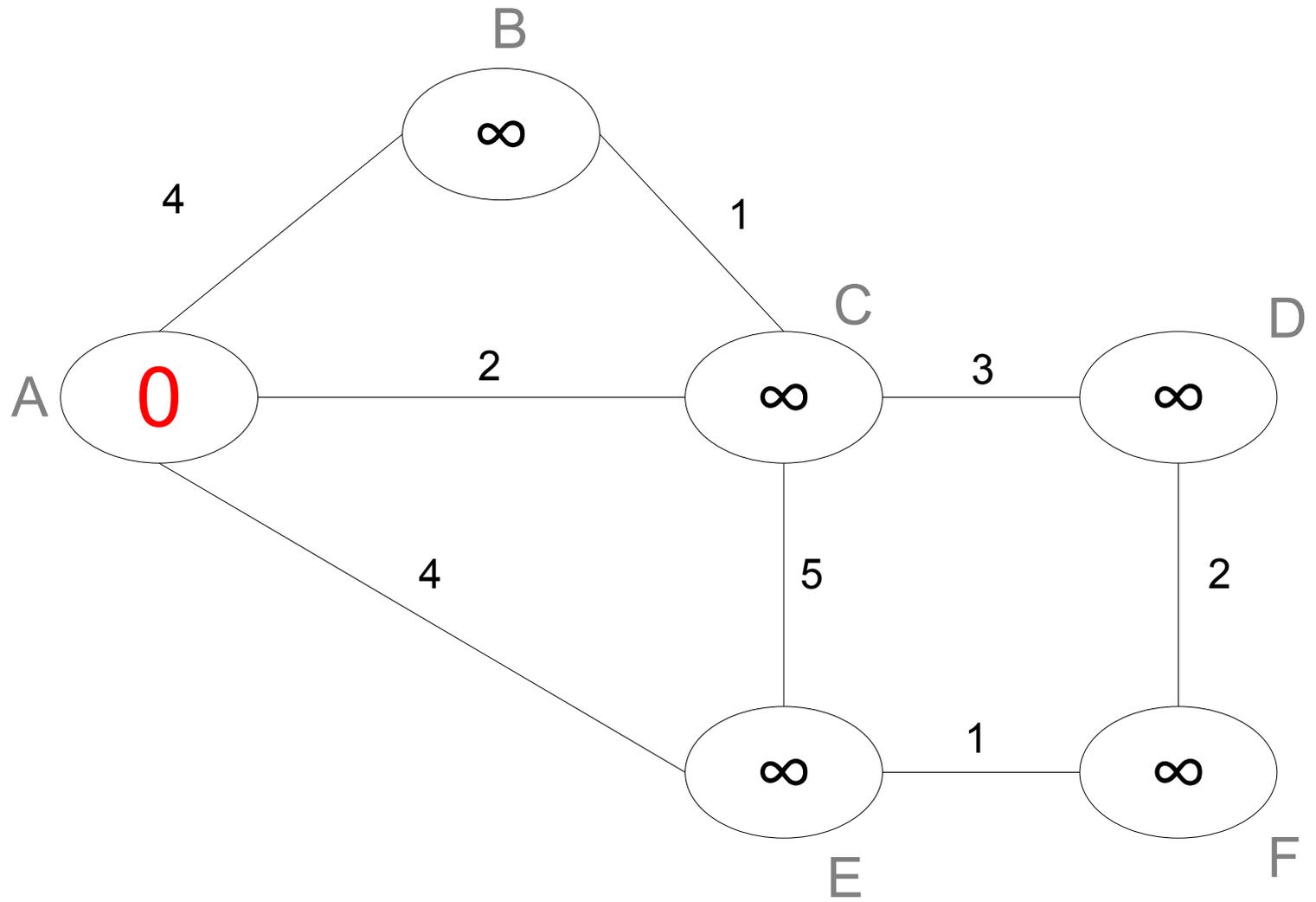
A	$\infty$	
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	
F	$\infty$	



Vorarbeit 2:

- setze die Distanz des Startknotens auf 0
- setze seinen Vorgänger auf sich selber (hilfreich, aber nicht nötig)

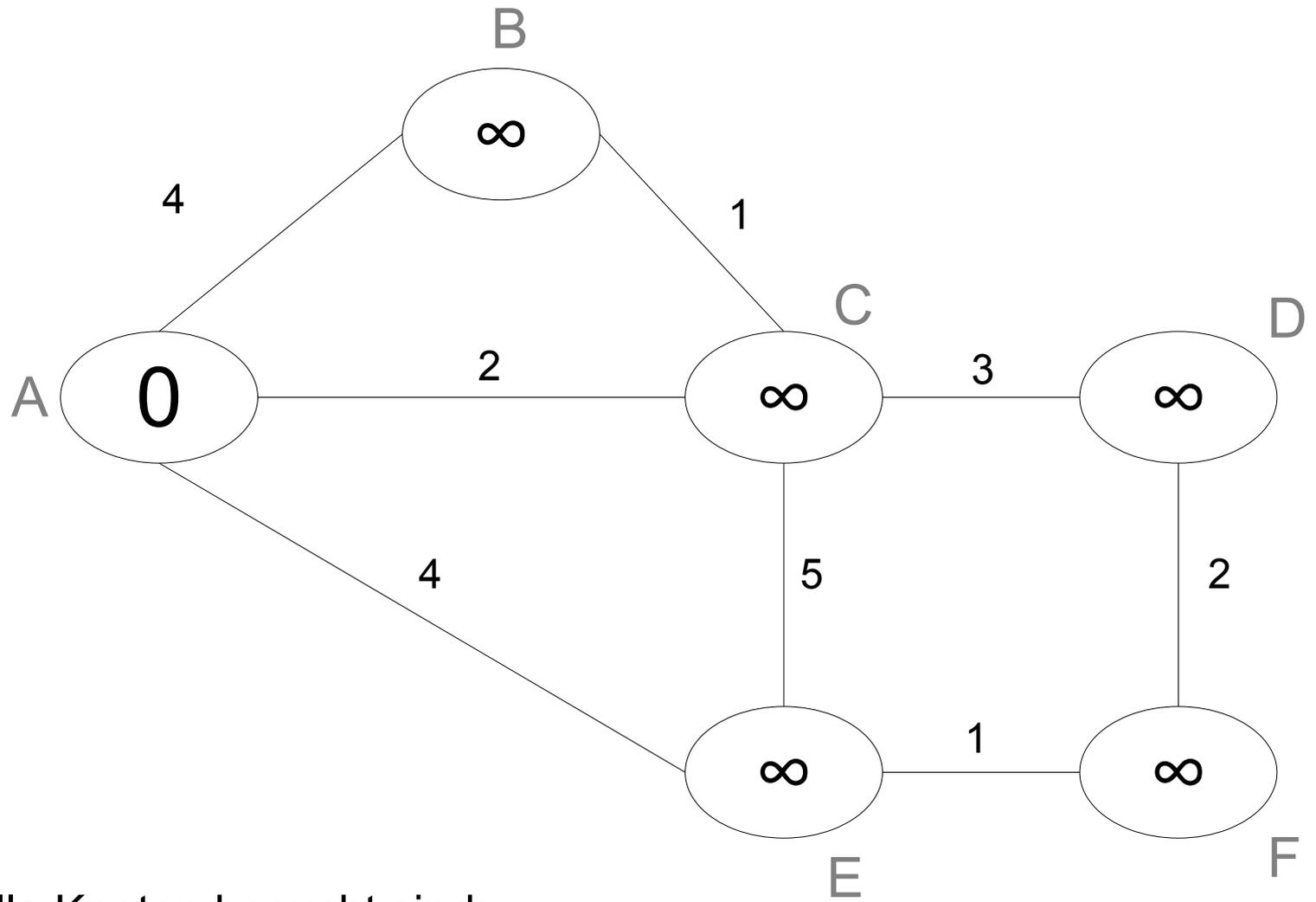
A	0	A
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	
F	$\infty$	



### Vorarbeit 2:

- setze die Distanz des Startknotens auf 0
- setze seinen Vorgänger auf sich selber (hilfreich, aber nicht nötig)

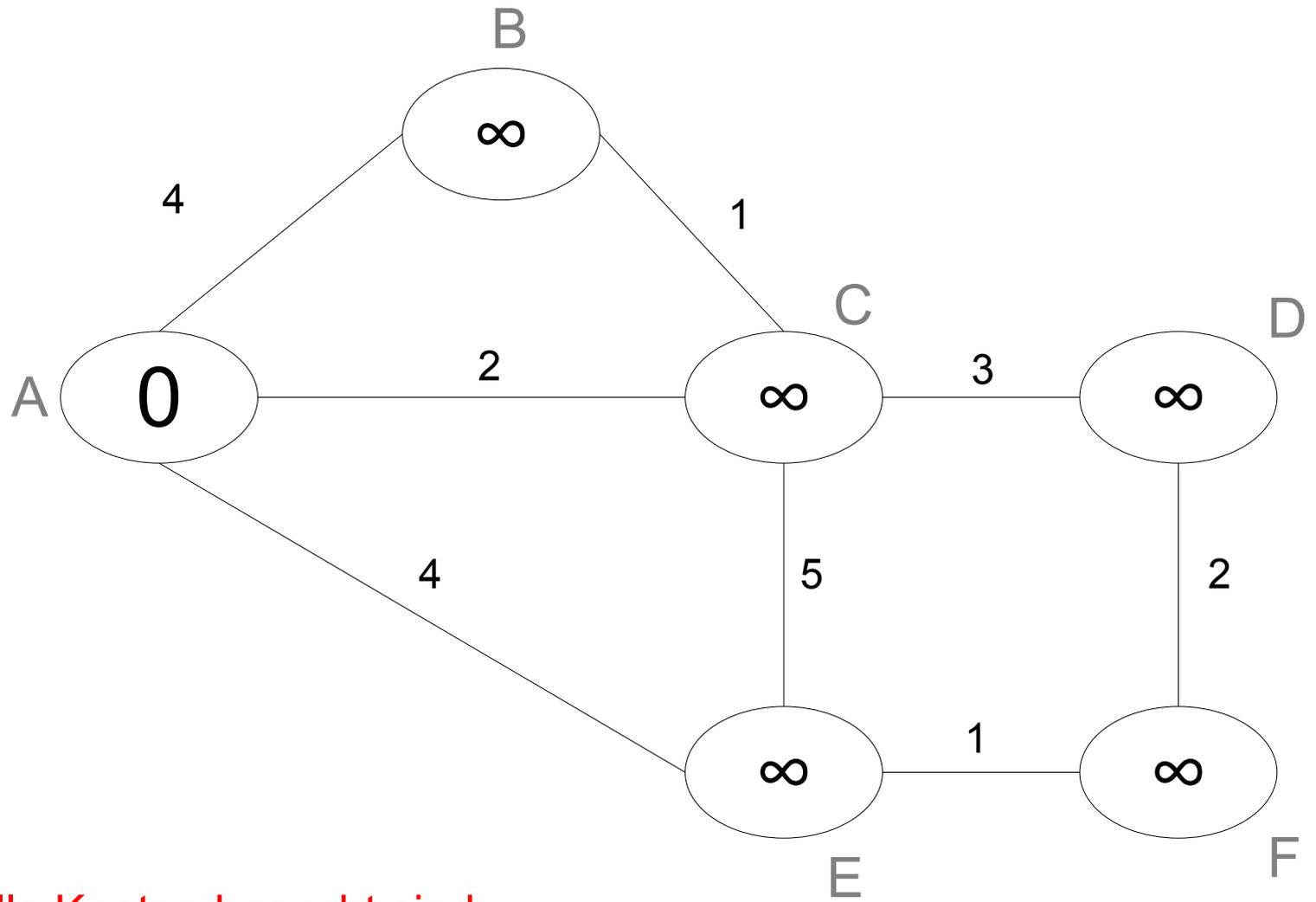
A	0	A
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

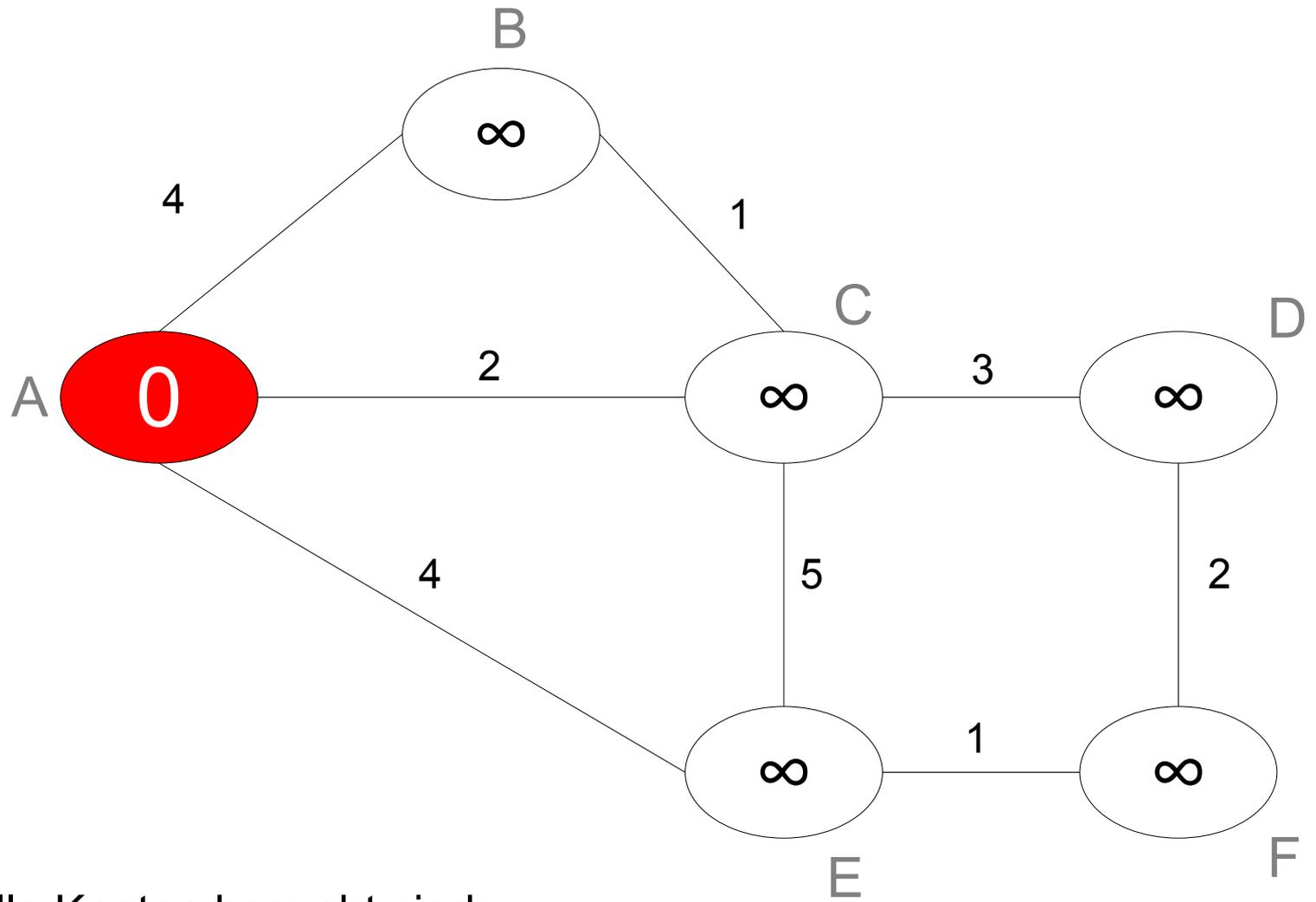
A	0	A
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

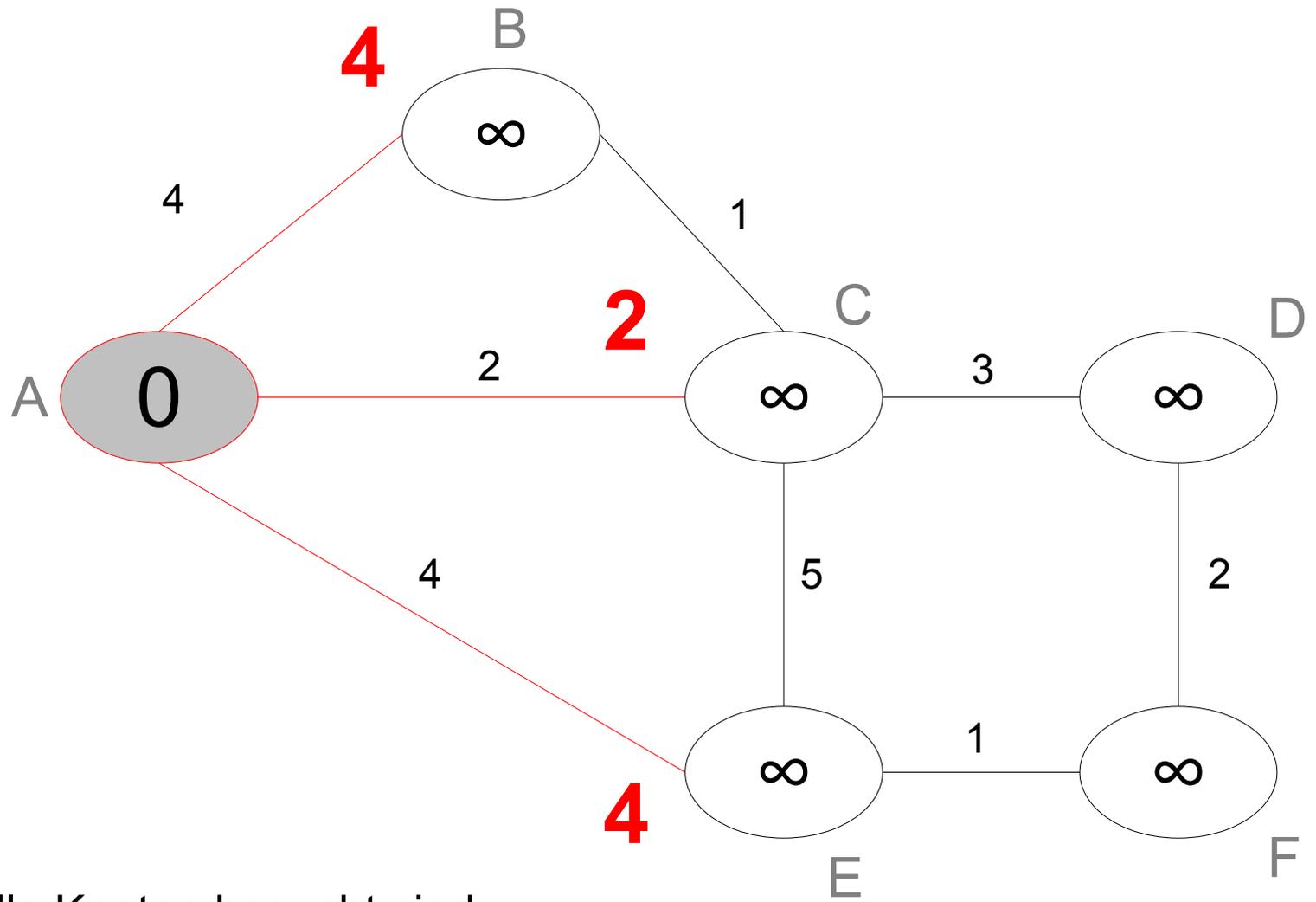
<b>A</b>	0	A
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- **setze den unbesuchten Knoten mit der geringsten Distanz als aktuell und besucht**
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

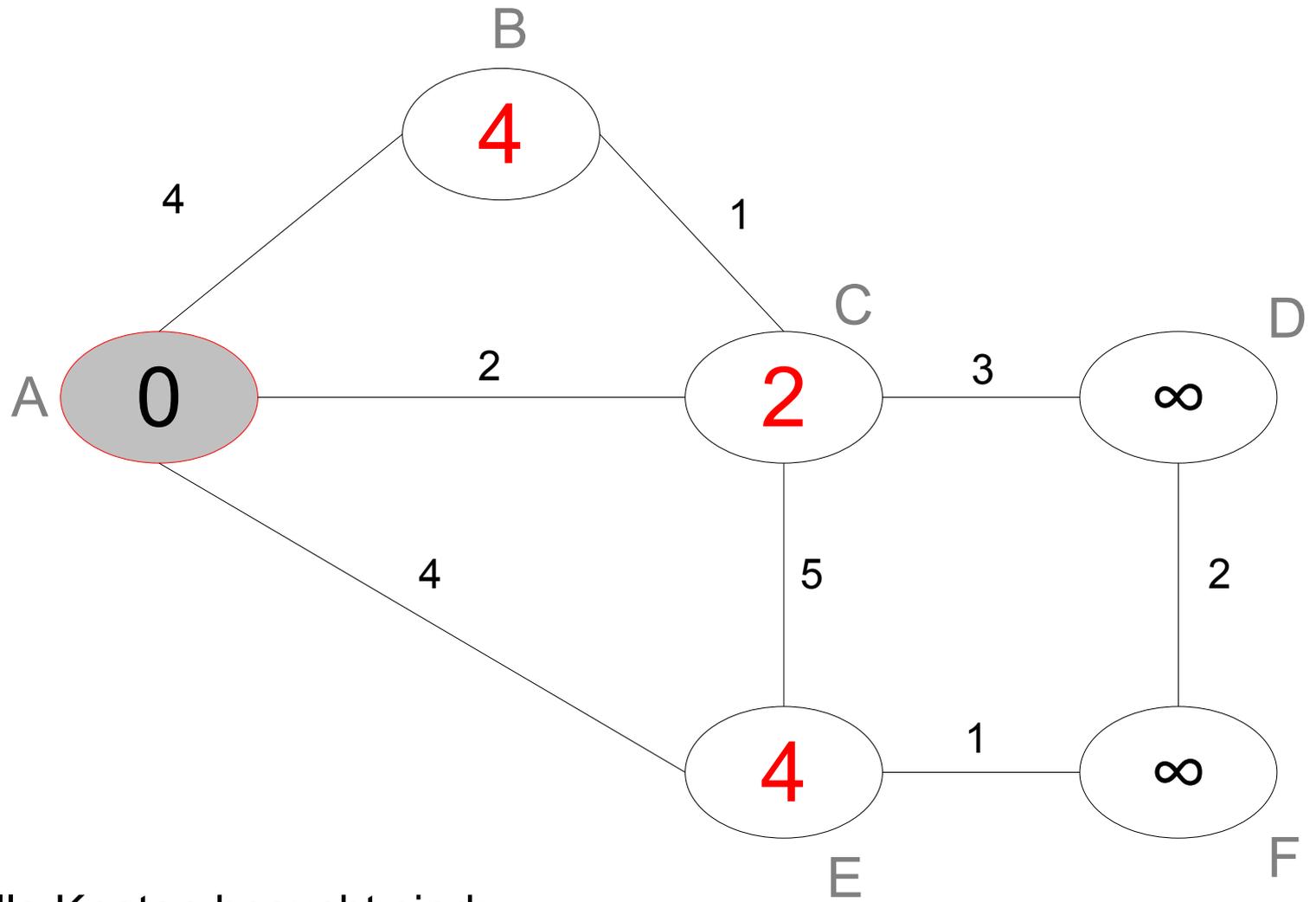
A	0	A
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

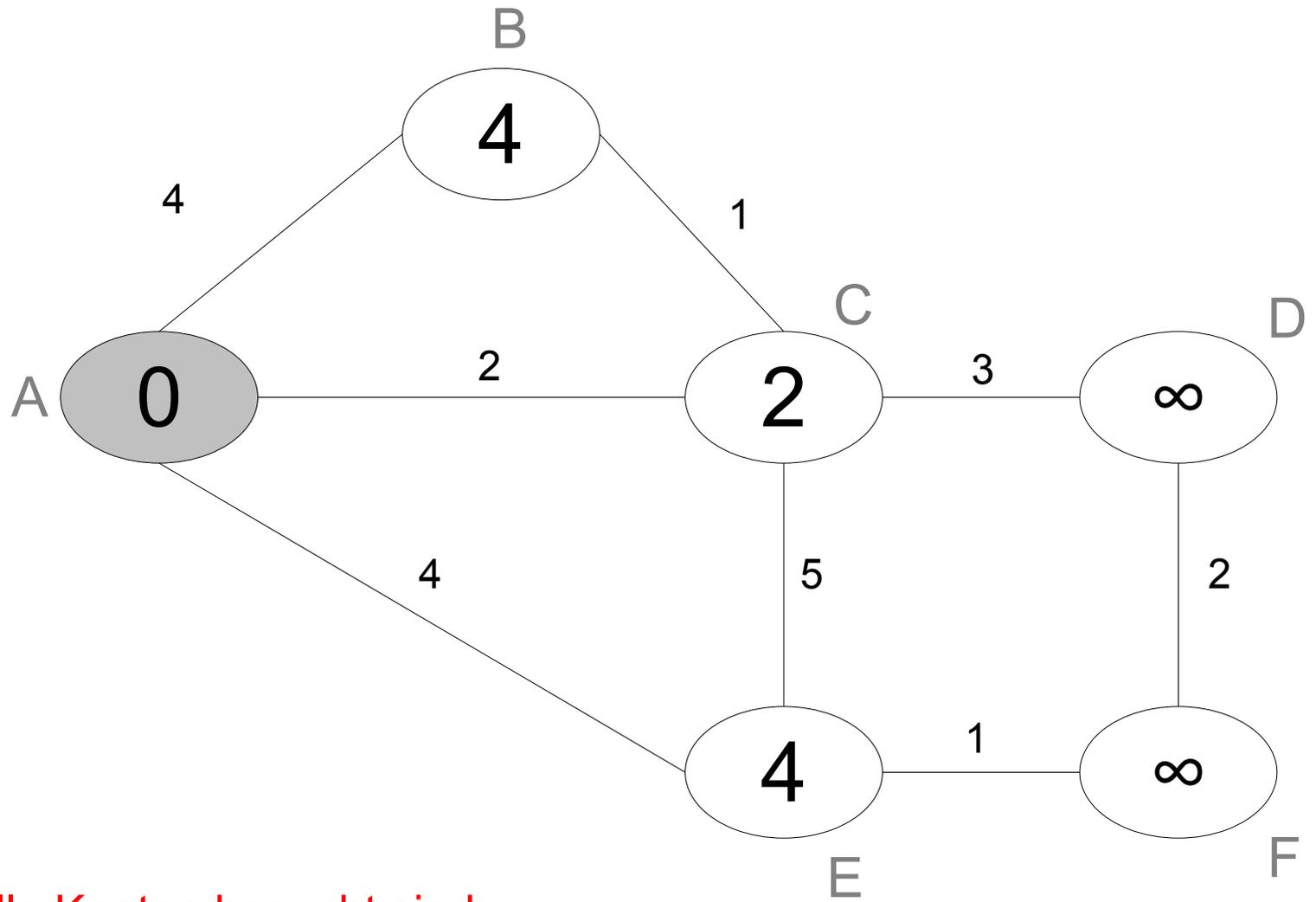
A	0	A
B	4	A
C	2	A
D	$\infty$	
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

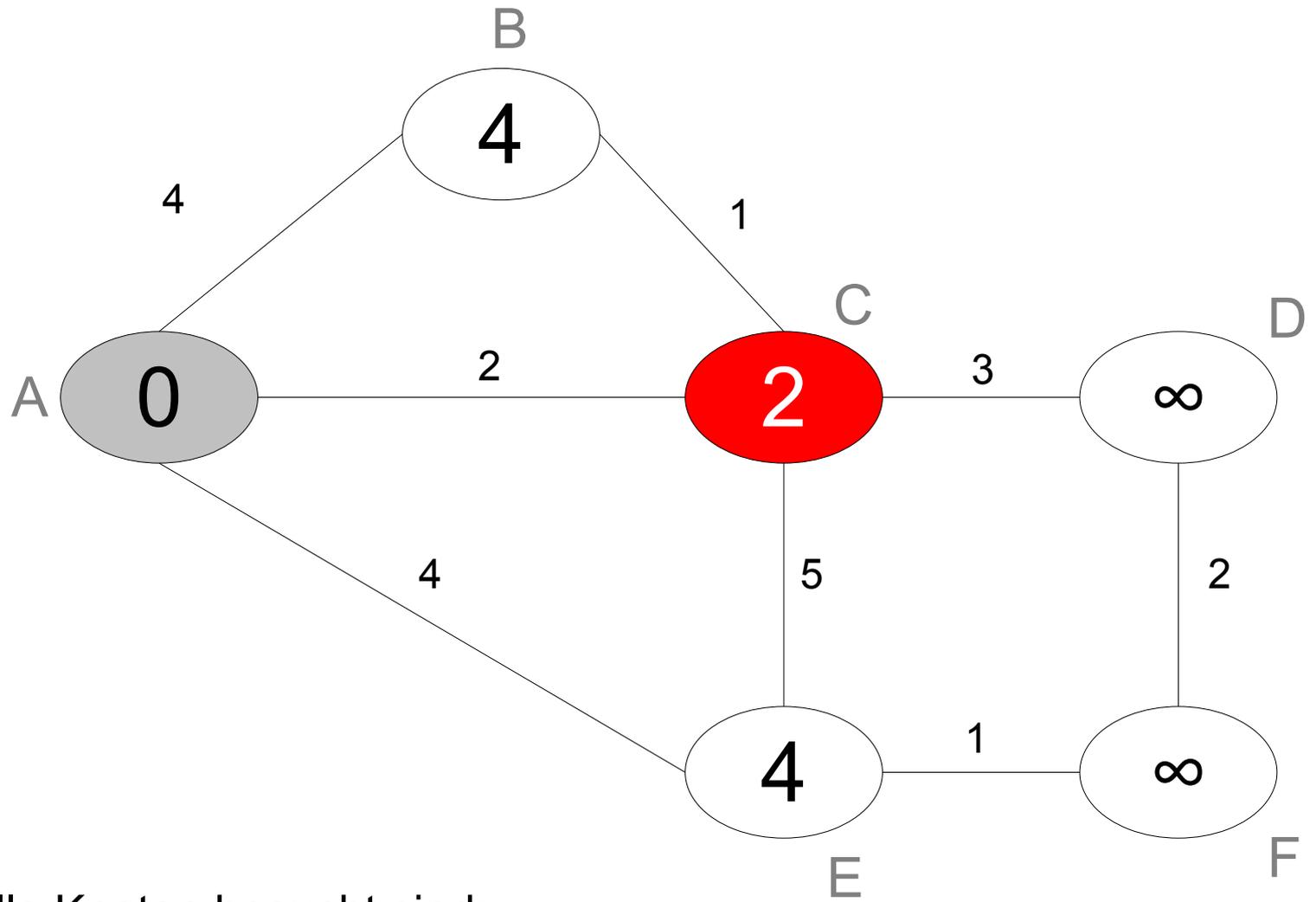
A	0	A
B	4	A
C	2	A
D	$\infty$	
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

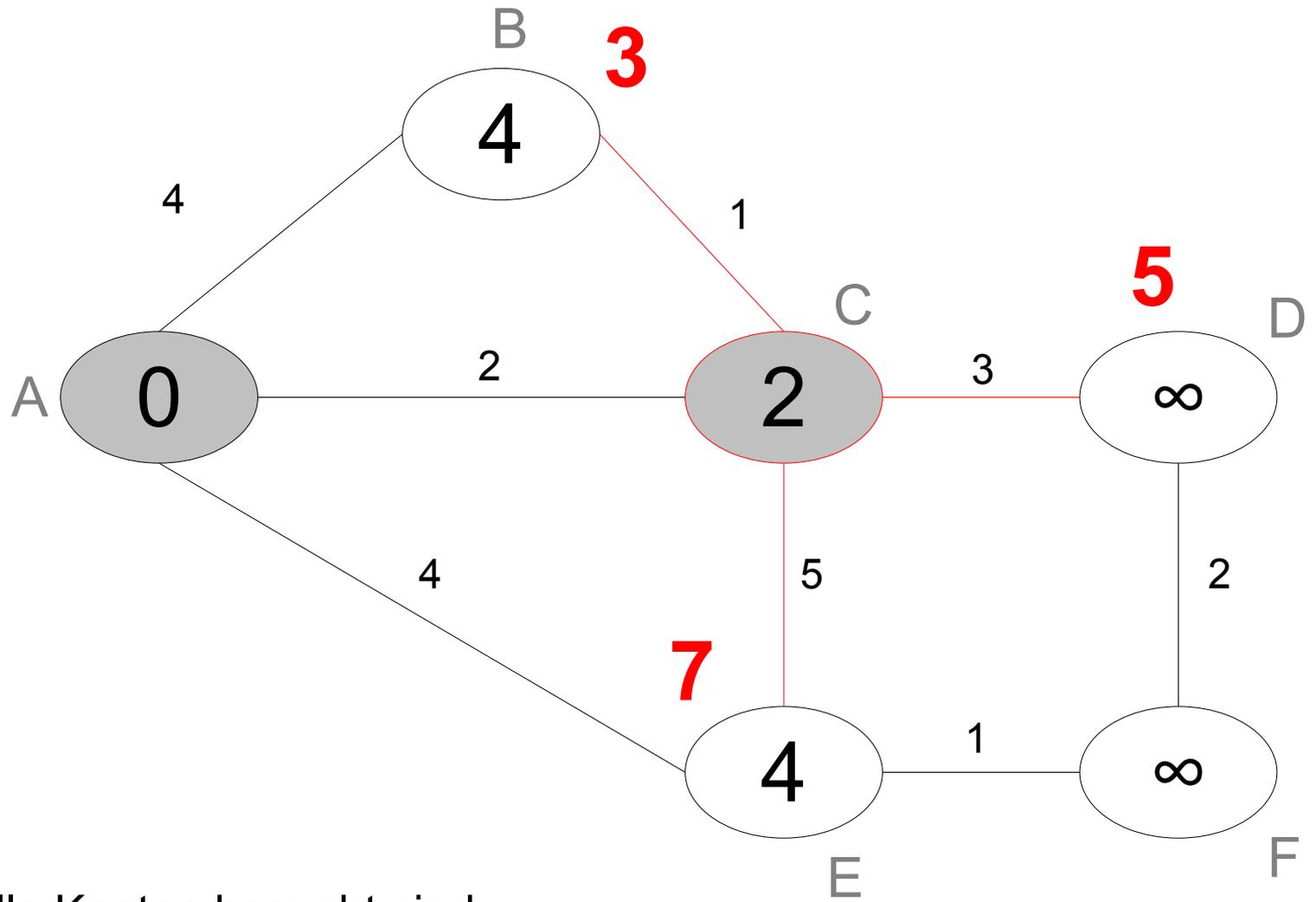
A	0	A
B	4	A
C	2	A
D	$\infty$	
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- **setze den unbesuchten Knoten mit der geringsten Distanz als aktuell und besucht**
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

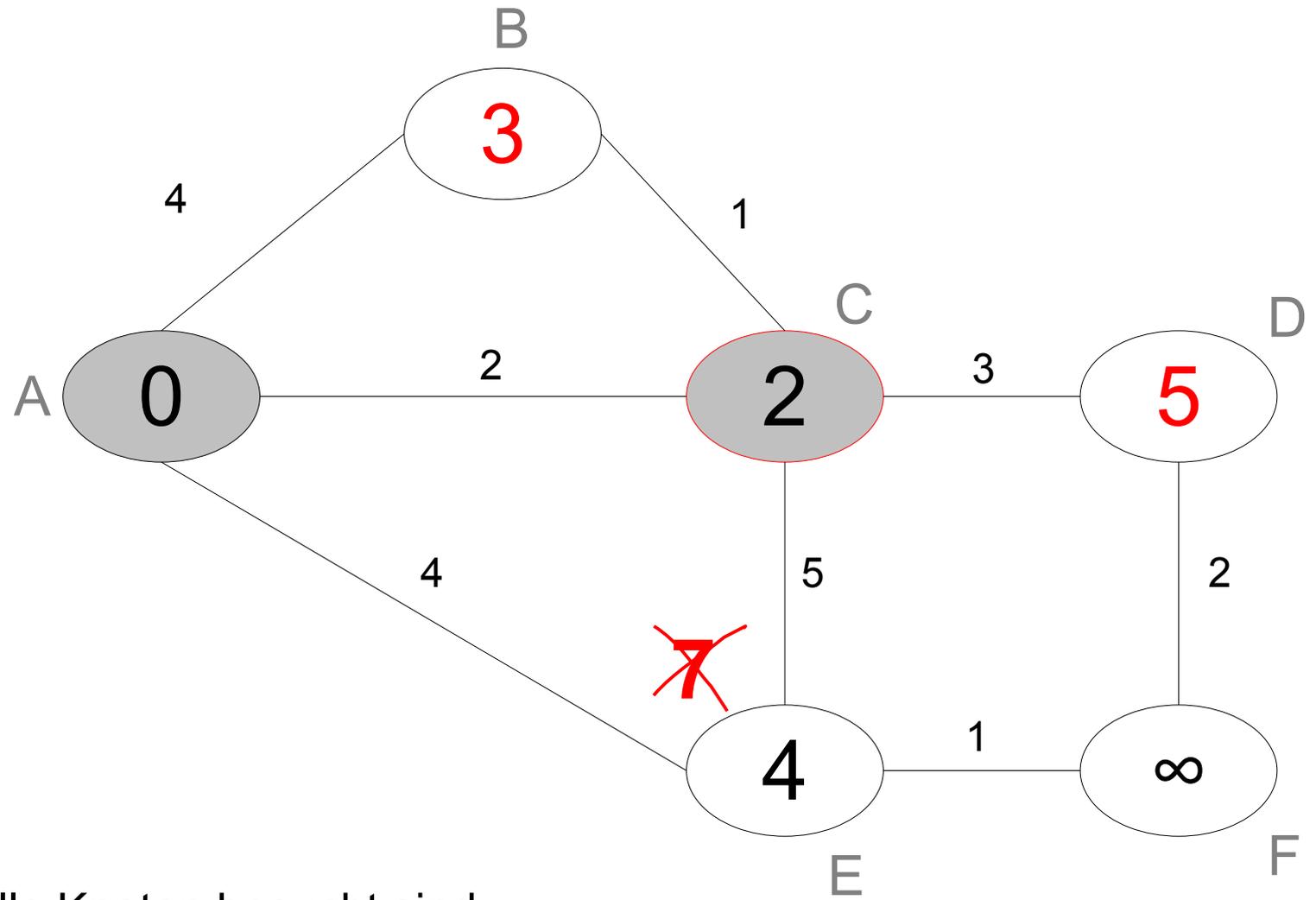
A	0	A
B	4	A
C	2	A
D	$\infty$	
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

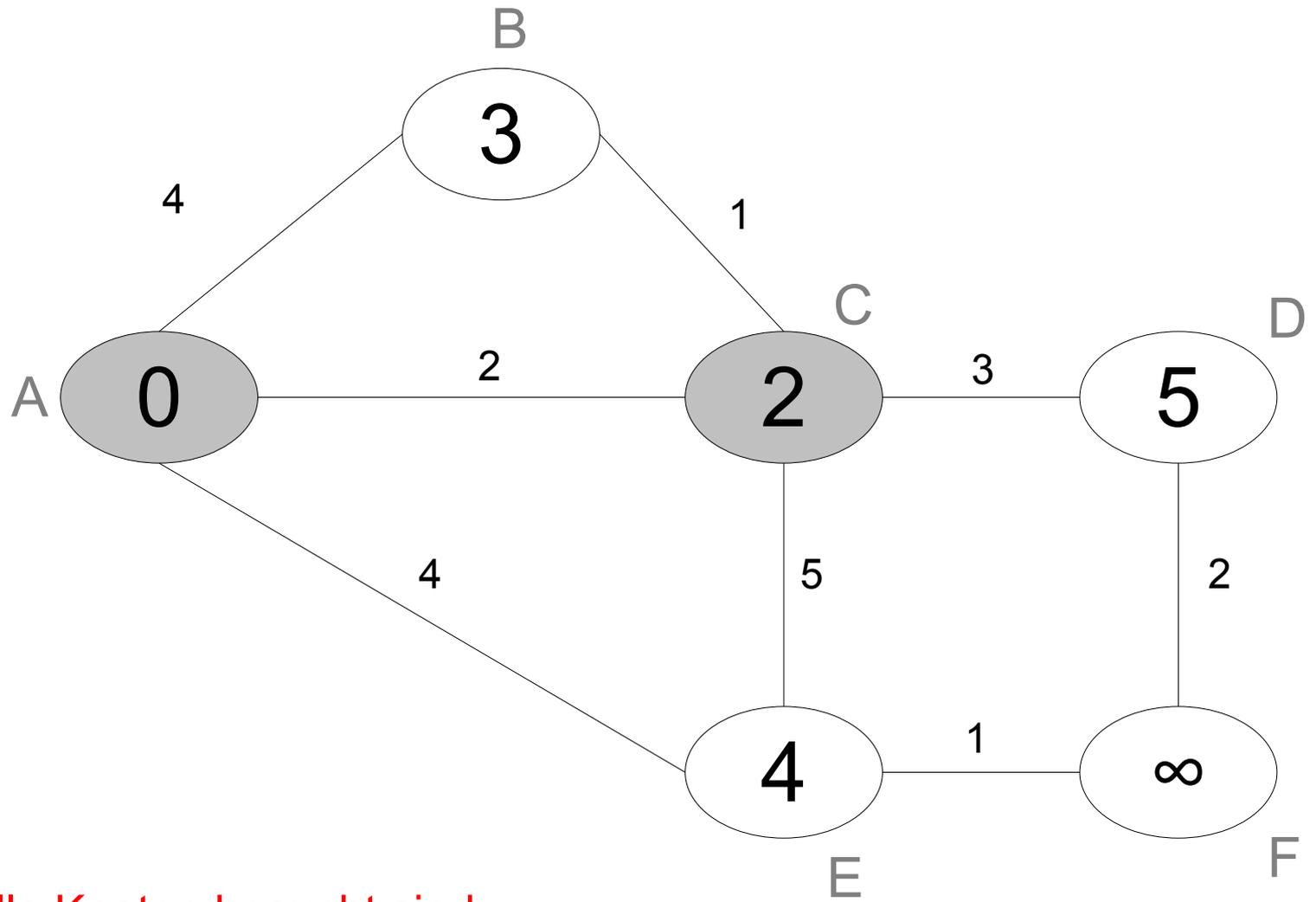
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

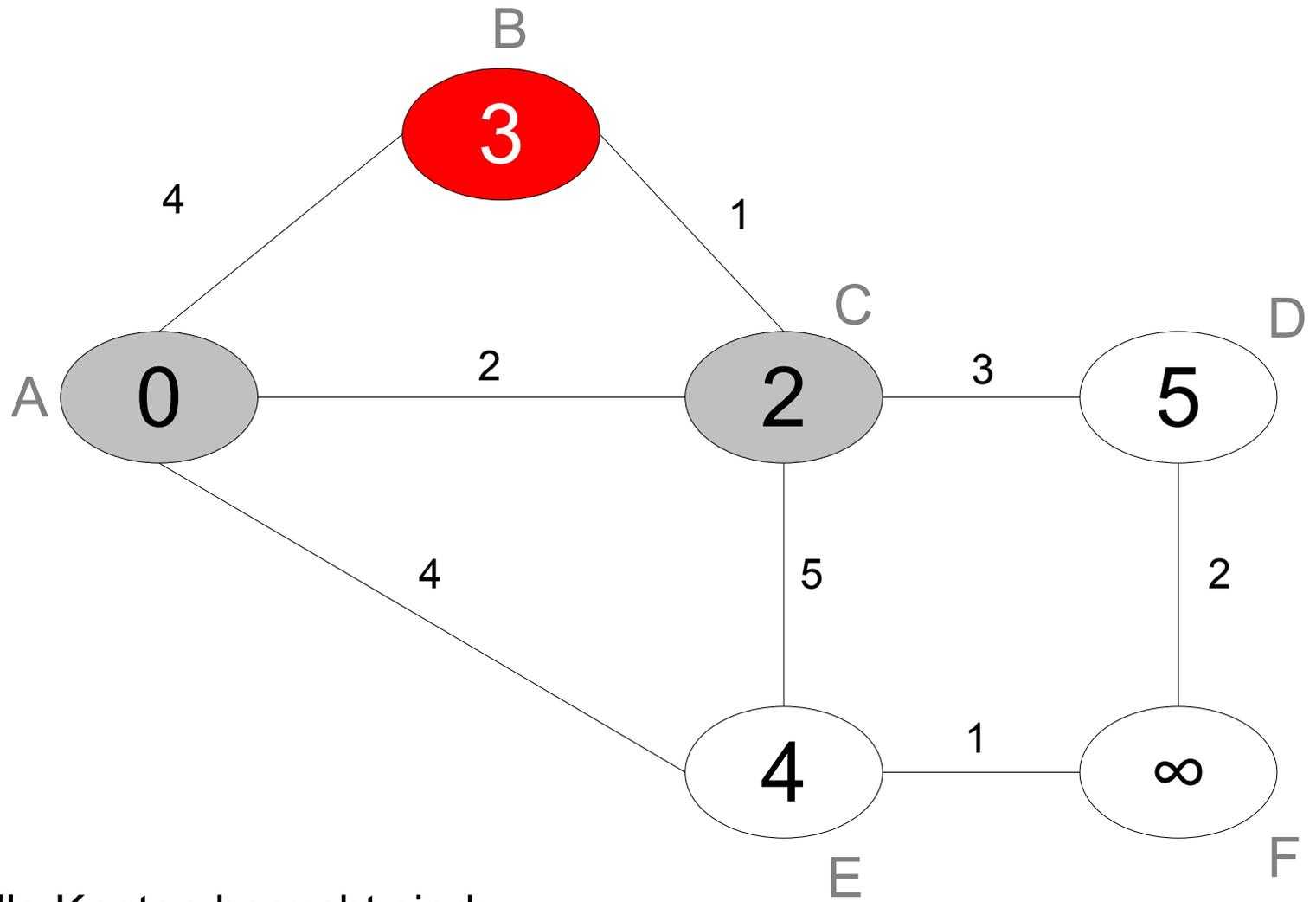
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	$\infty$	



**Wiederhole, bis alle Knoten besucht sind:**

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

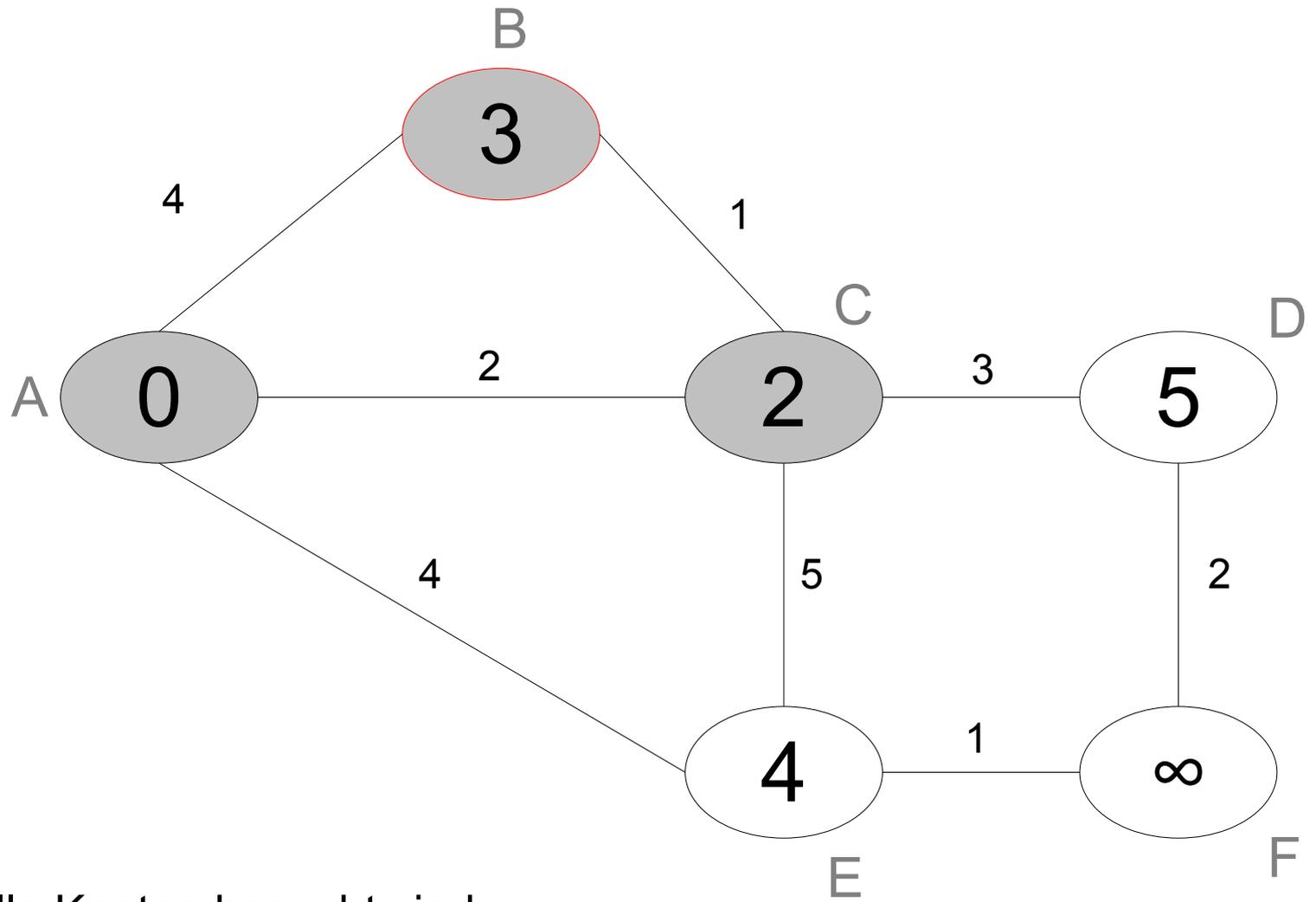
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- **setze den unbesuchten Knoten mit der geringsten Distanz als aktuell und besucht**
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

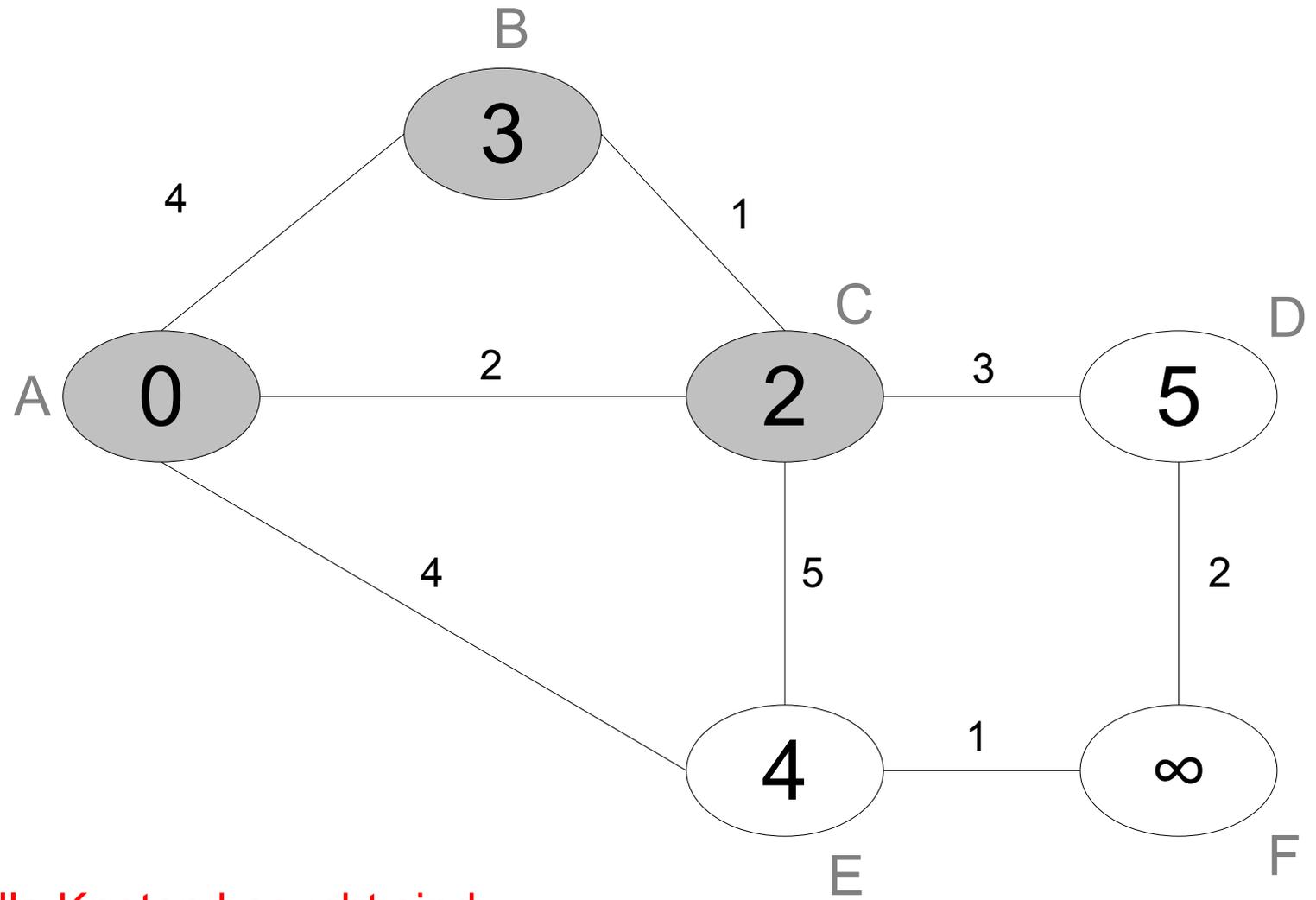
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

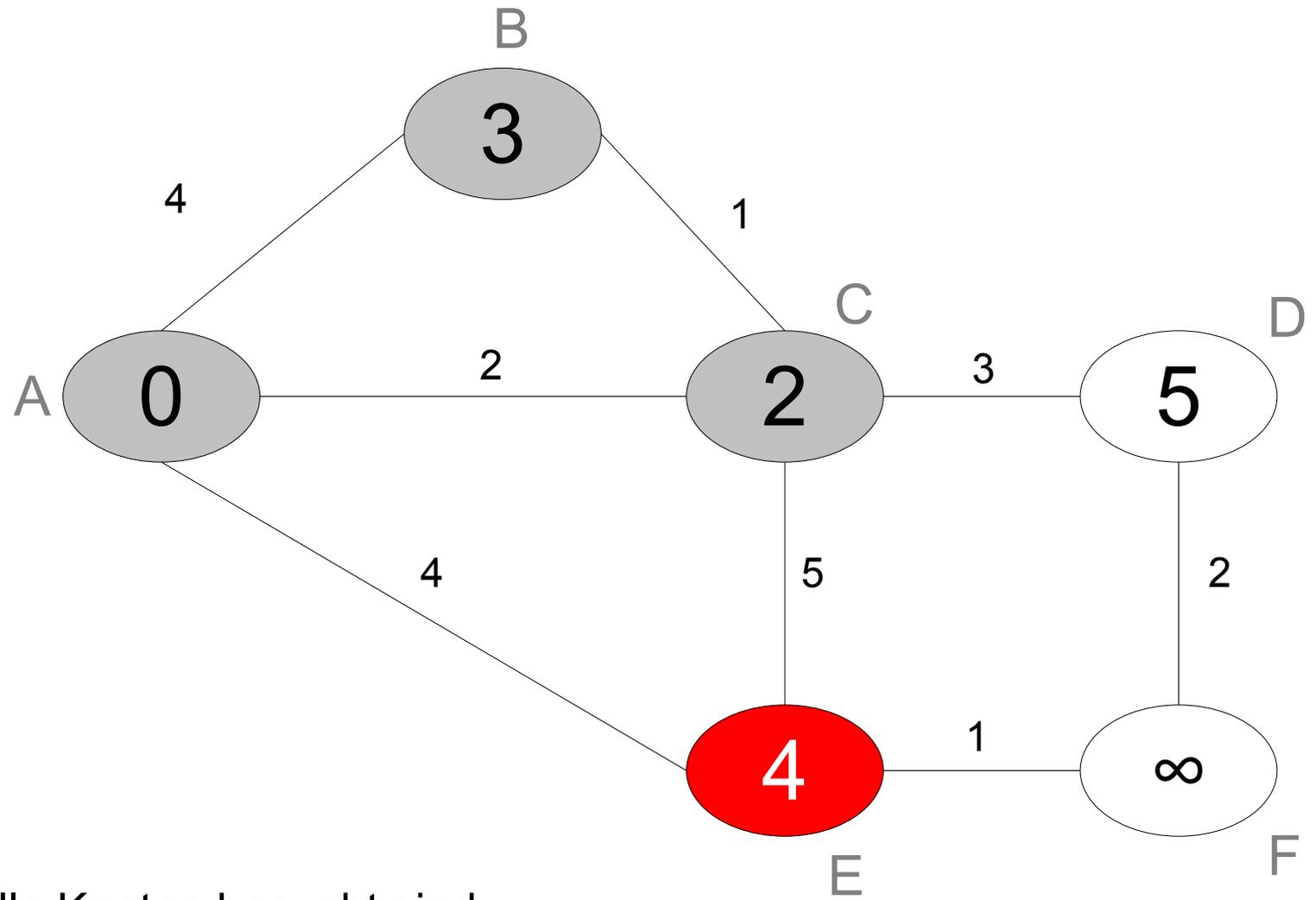
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	$\infty$	



**Wiederhole, bis alle Knoten besucht sind:**

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

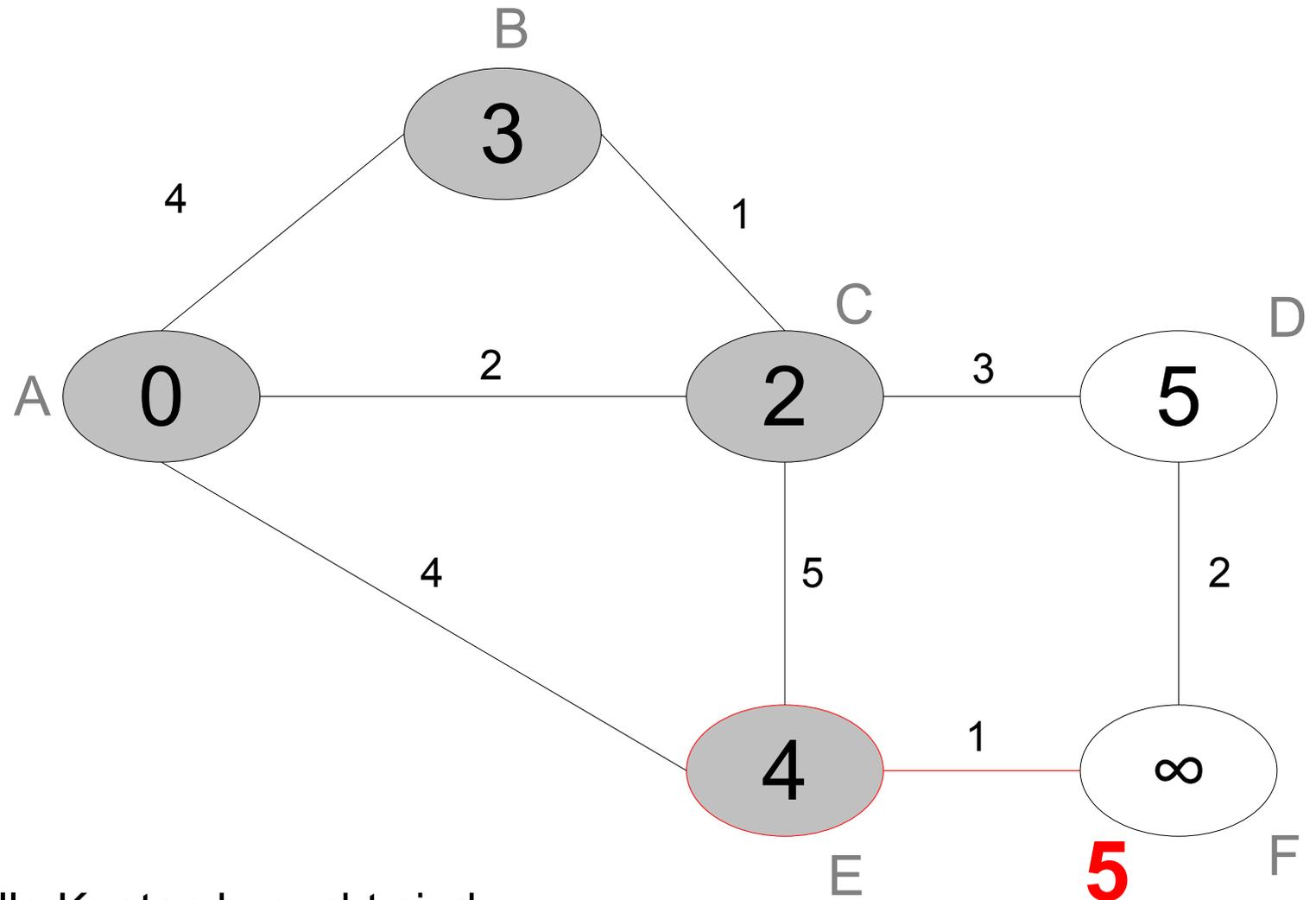
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- **setze den unbesuchten Knoten mit der geringsten Distanz als aktuell und besucht**
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

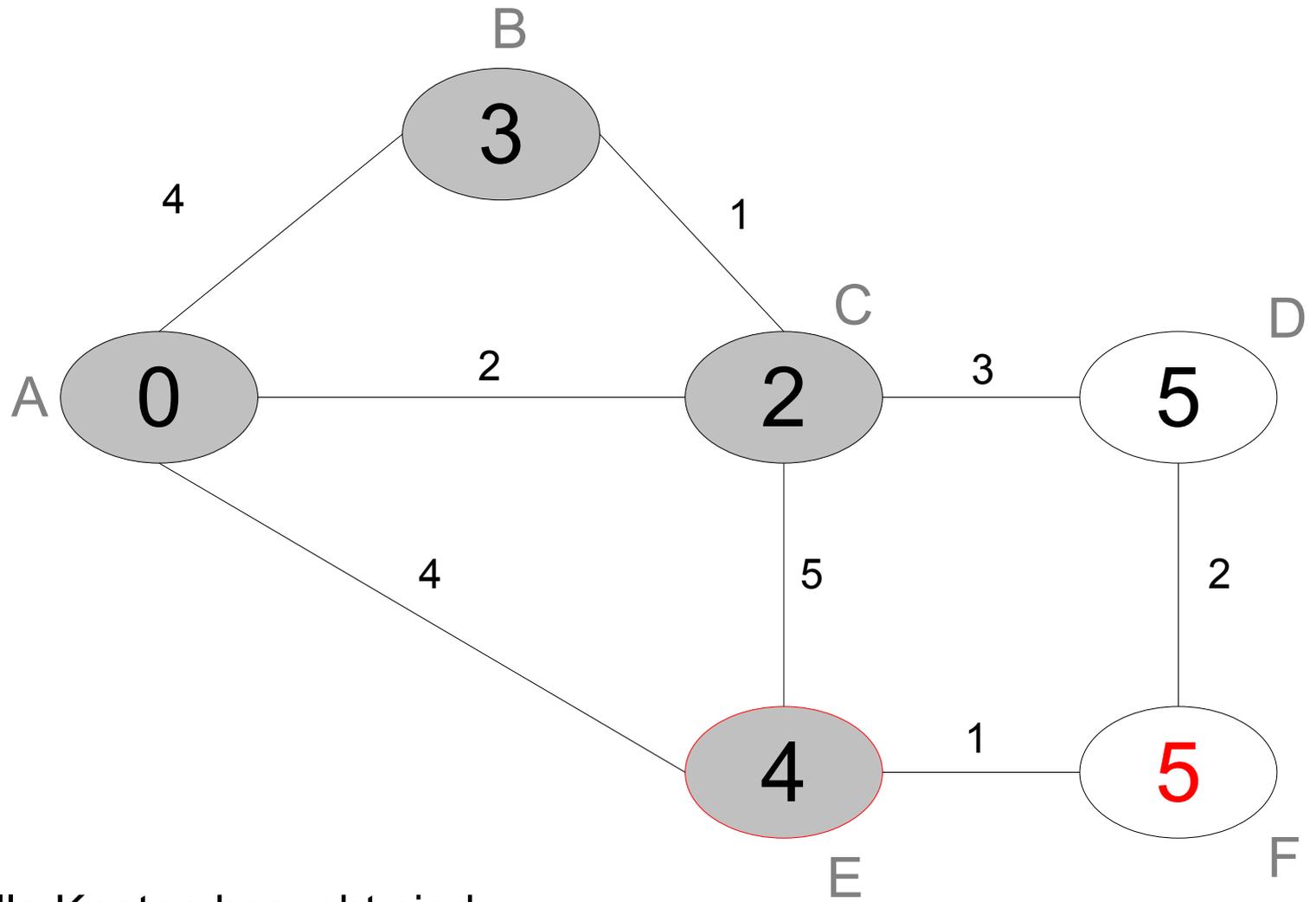
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	$\infty$	



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

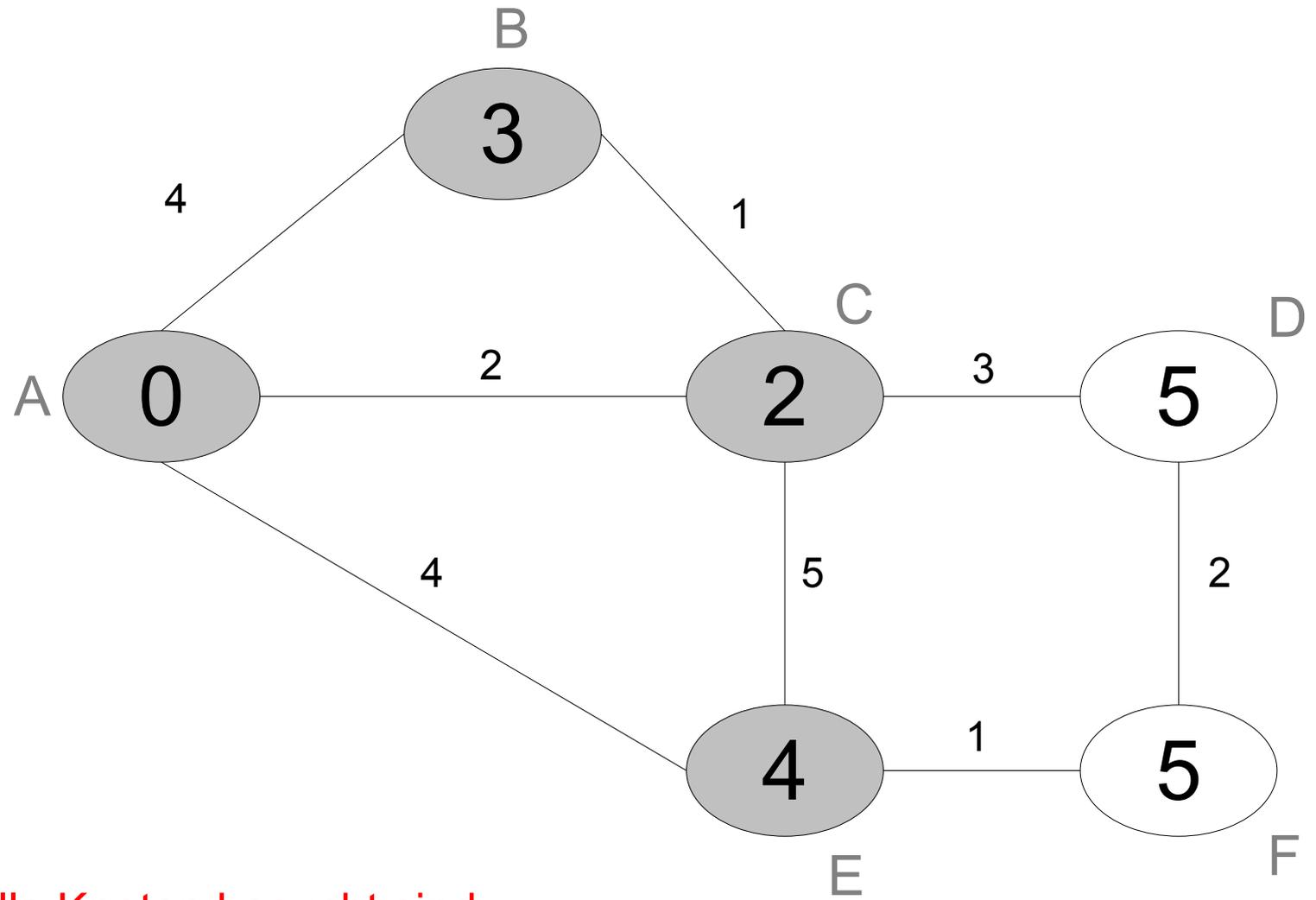
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

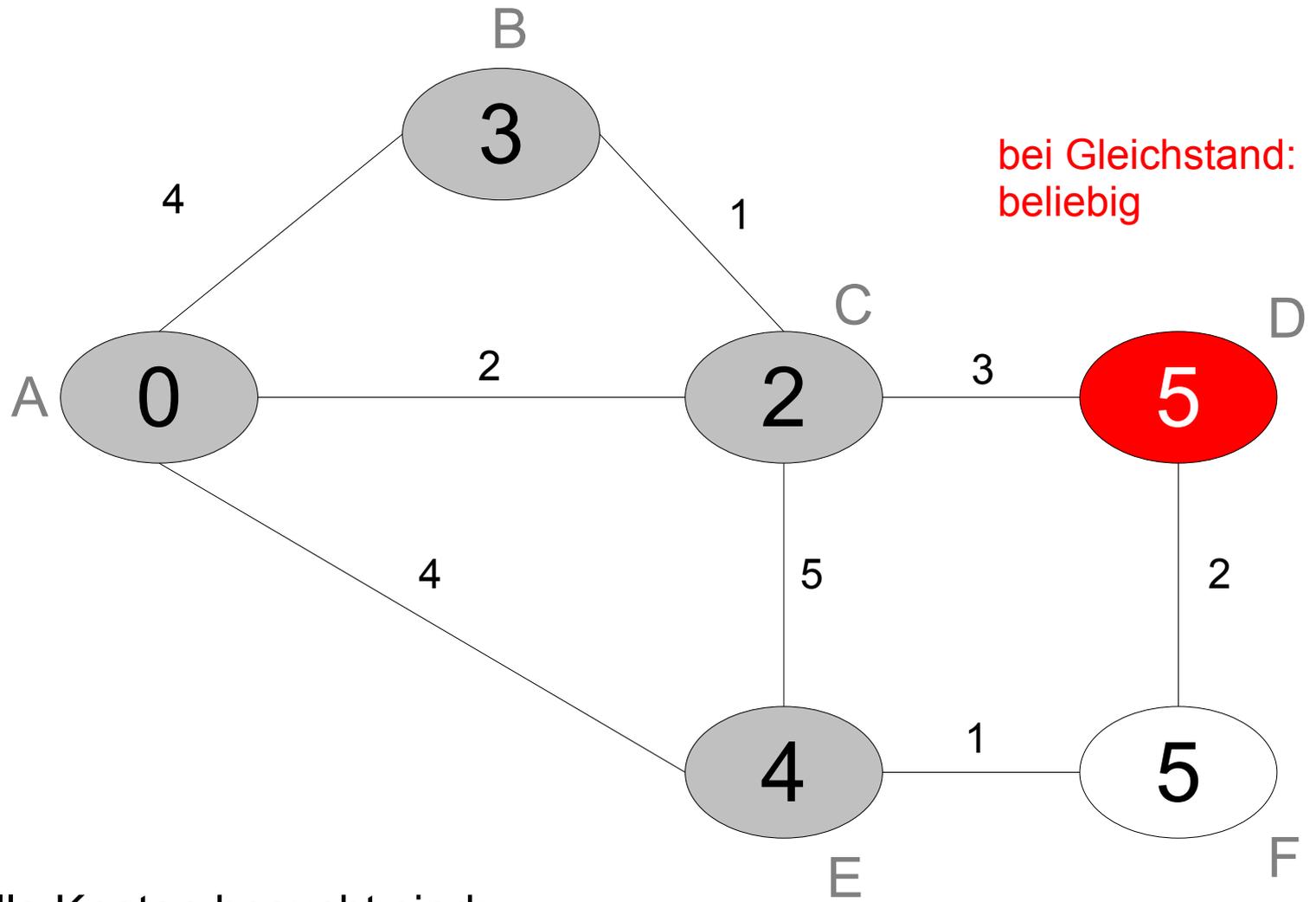
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

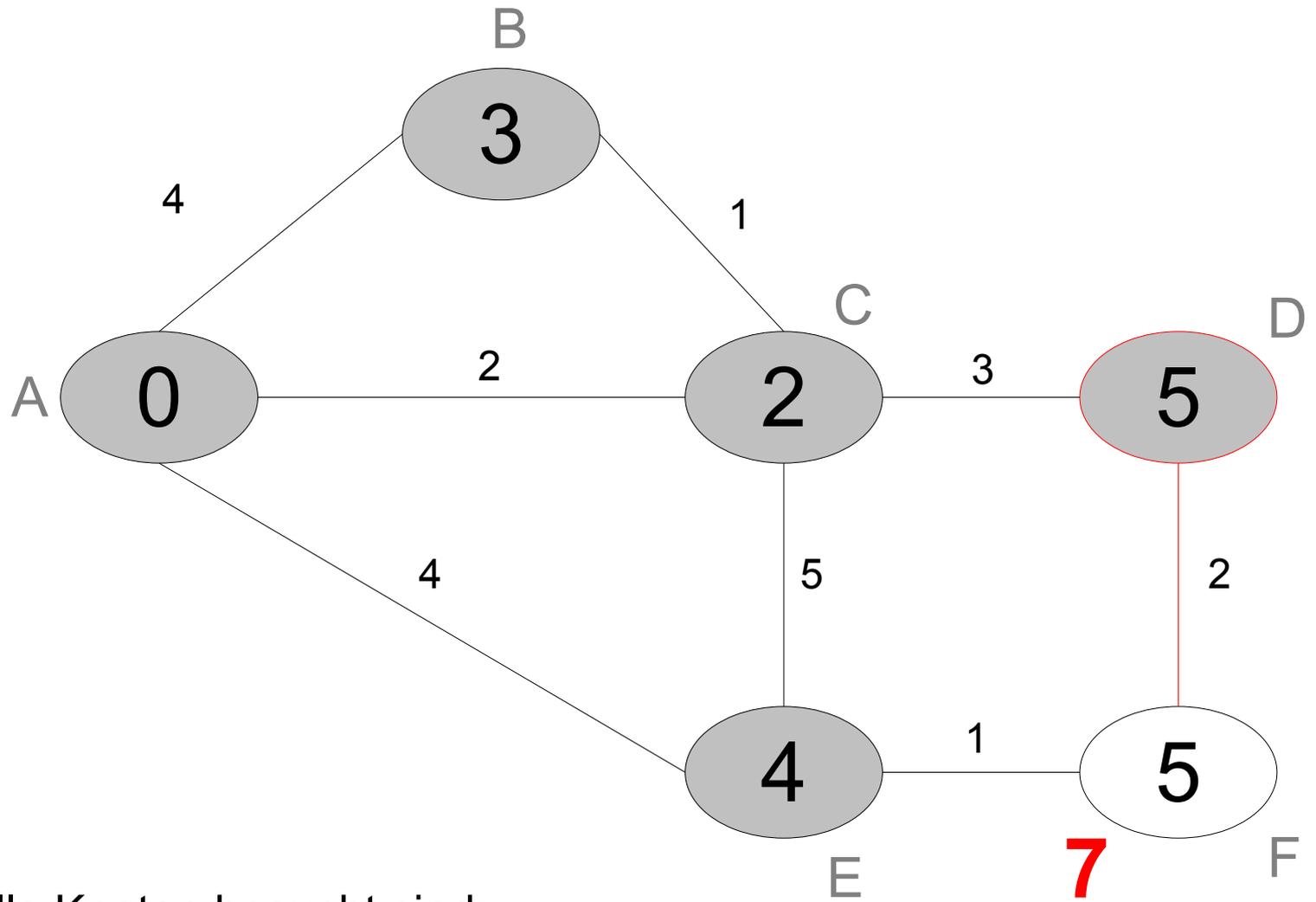
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- **setze den unbesuchten Knoten mit der geringsten Distanz als aktuell und besucht**
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

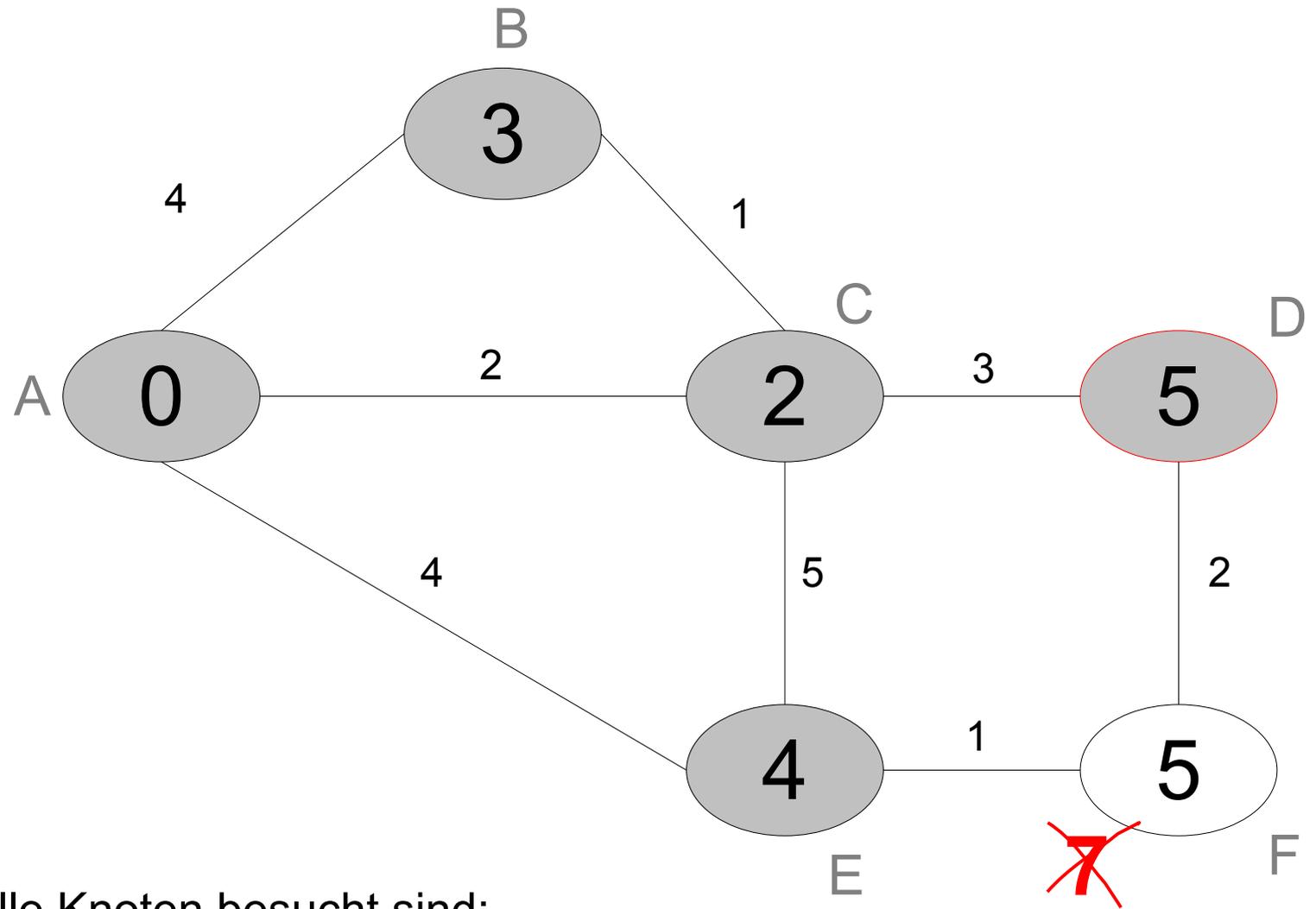
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

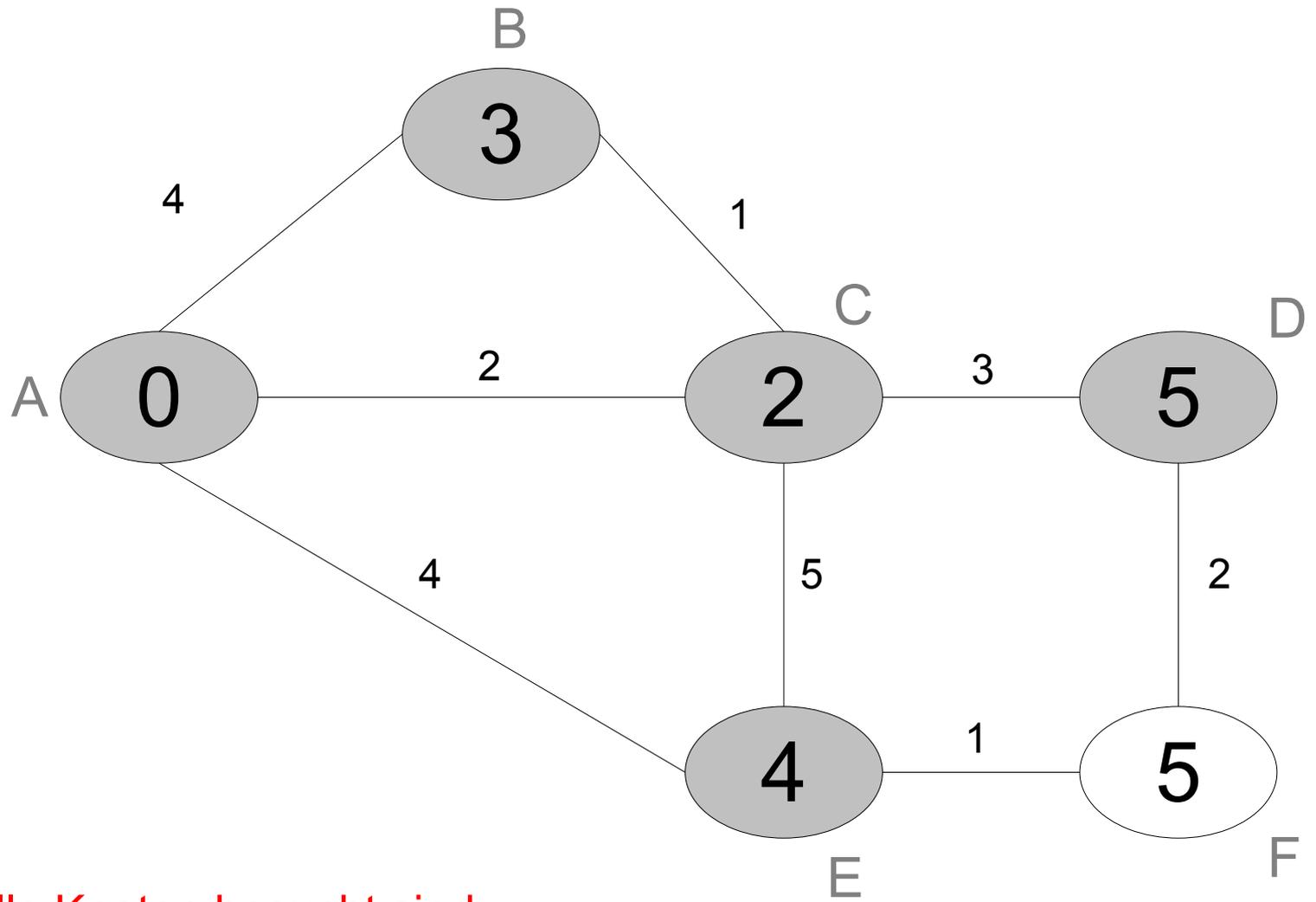
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

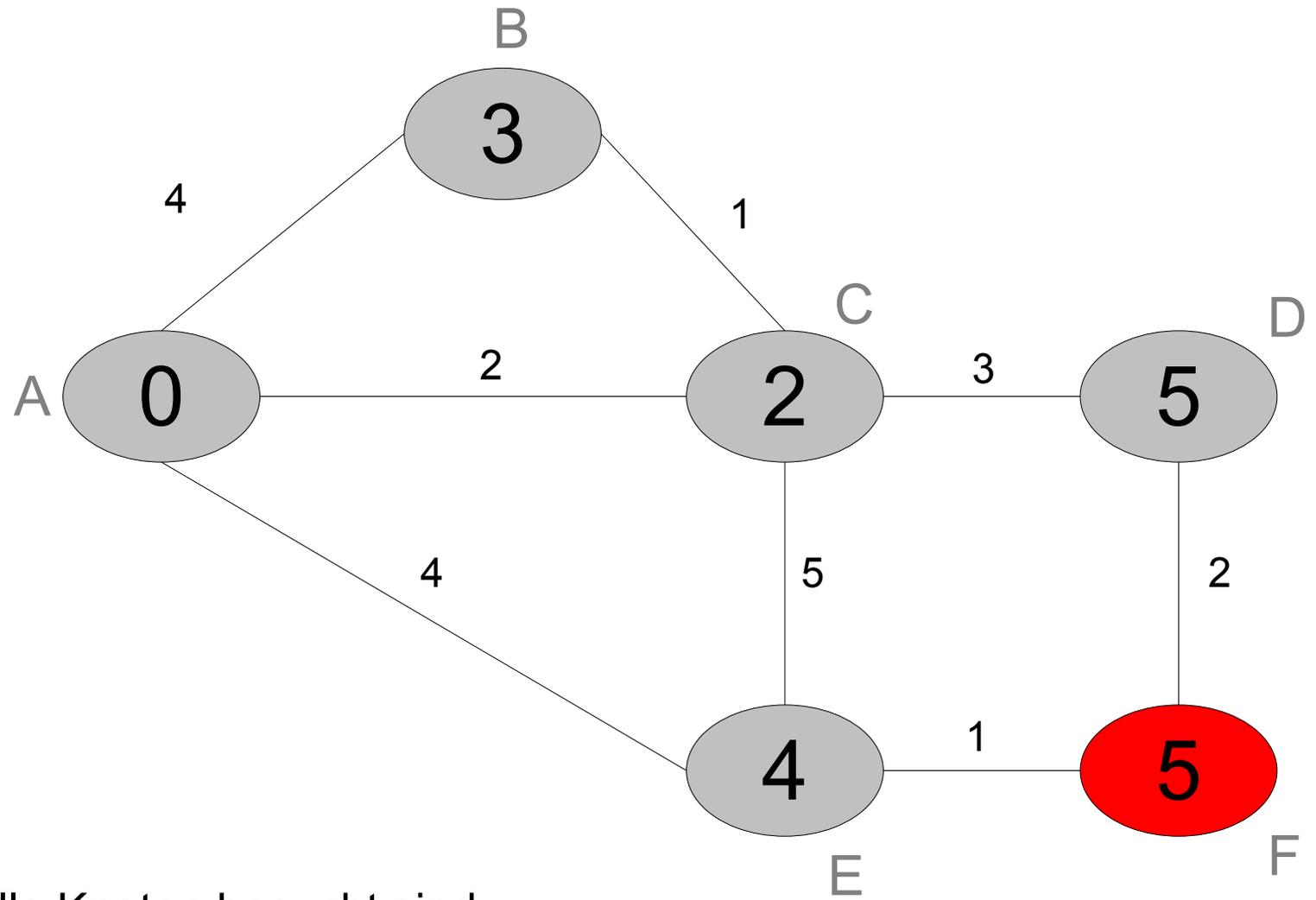
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



**Wiederhole, bis alle Knoten besucht sind:**

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

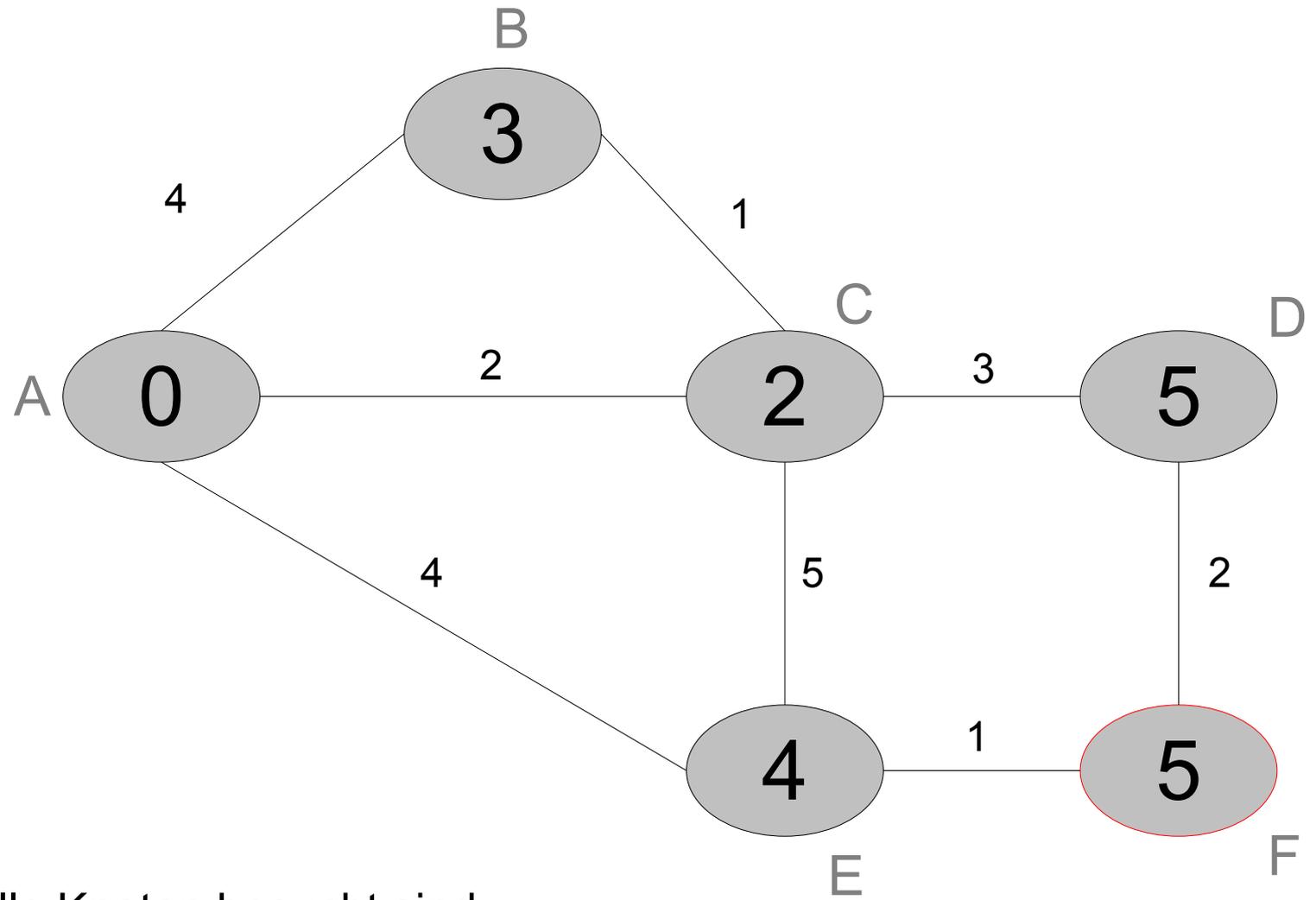
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- **setze den unbesuchten Knoten mit der geringsten Distanz als aktuell und besucht**
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
  - dann setze sie
  - und setze dich als seinen Vorgänger

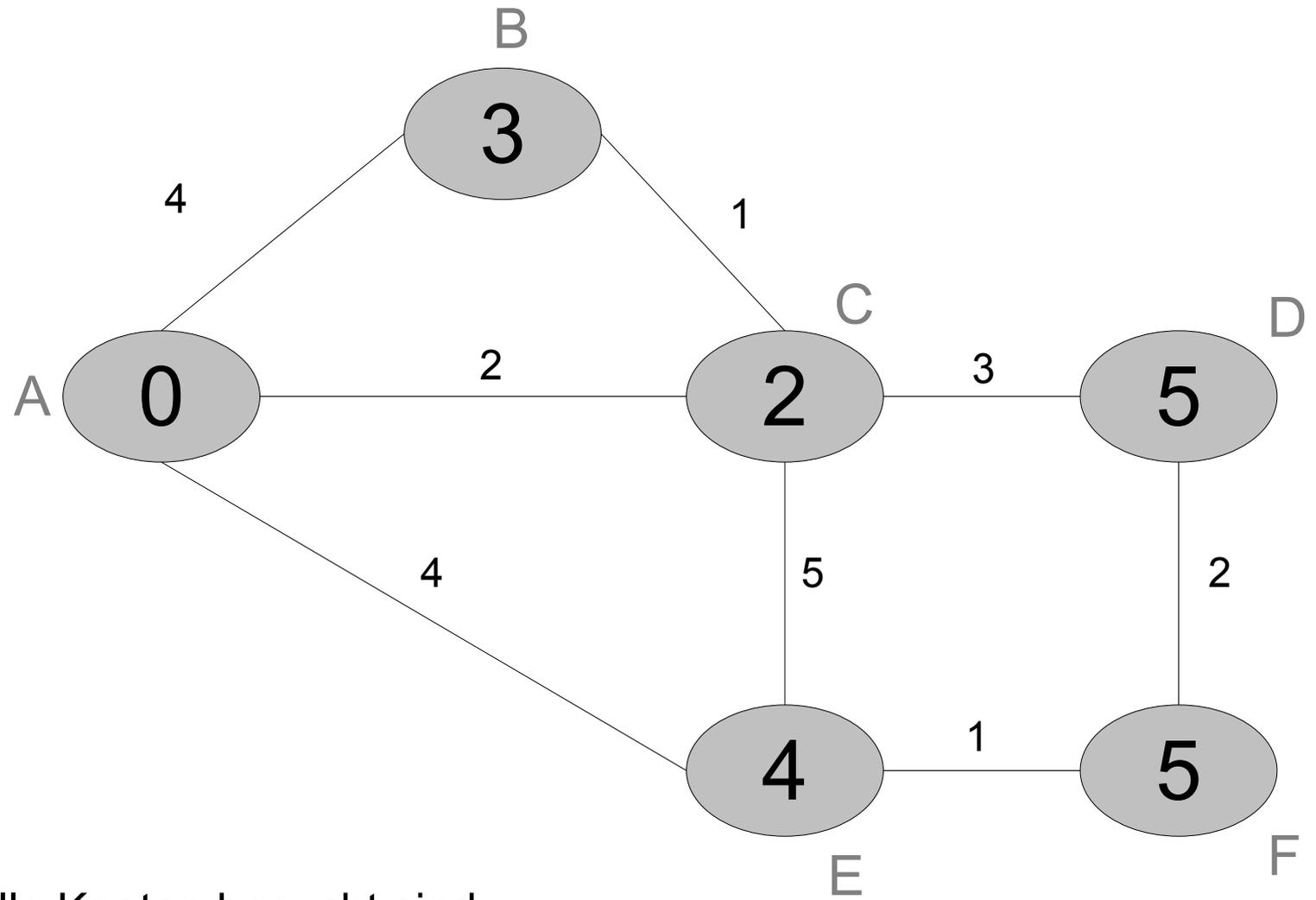
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

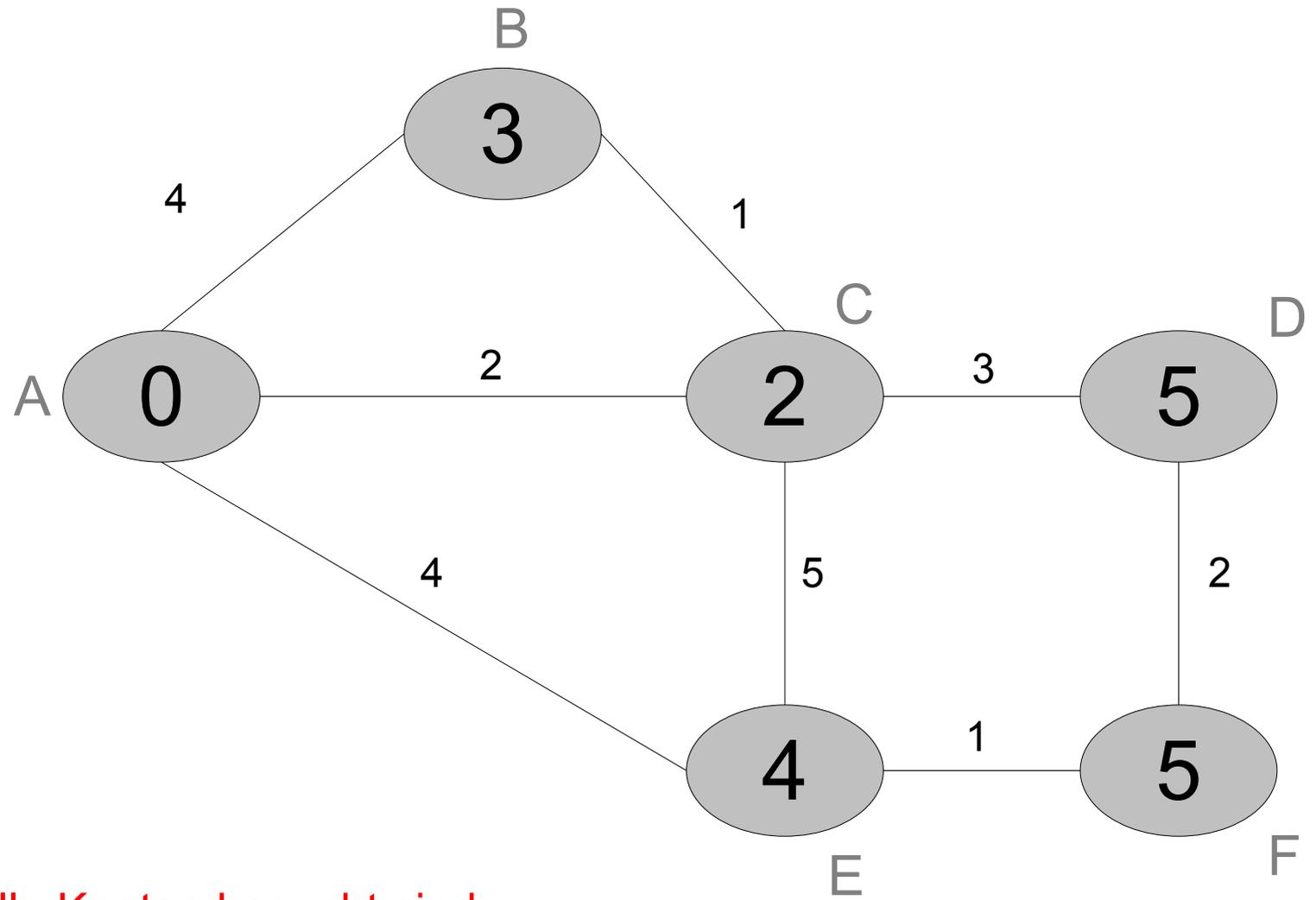
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Wiederhole, bis alle Knoten besucht sind:

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



**Wiederhole, bis alle Knoten besucht sind:**

- setze den unbesuchten **Knoten** mit der geringsten Distanz als aktuell und besucht
- für alle unbesuchten **Nachbarn**: addiere eigene Distanz und das Kantengewicht
  - wenn Summe geringer ist als deren aktuelle Distanz,
    - dann setze sie
    - und setze dich als seinen Vorgänger

# Der Dijkstra-Algorithmus

Von einem gegebenen Startknoten aus  
die kürzesten Wege zu allen anderen Knoten

(Single Source Shortest Path SSSP)

Angewendet auf alle Knoten im Graphen: liefert  
alle kürzesten Wege.