L18

4102

10.29.2013

abhi shelat

Shortest paths, negative weights

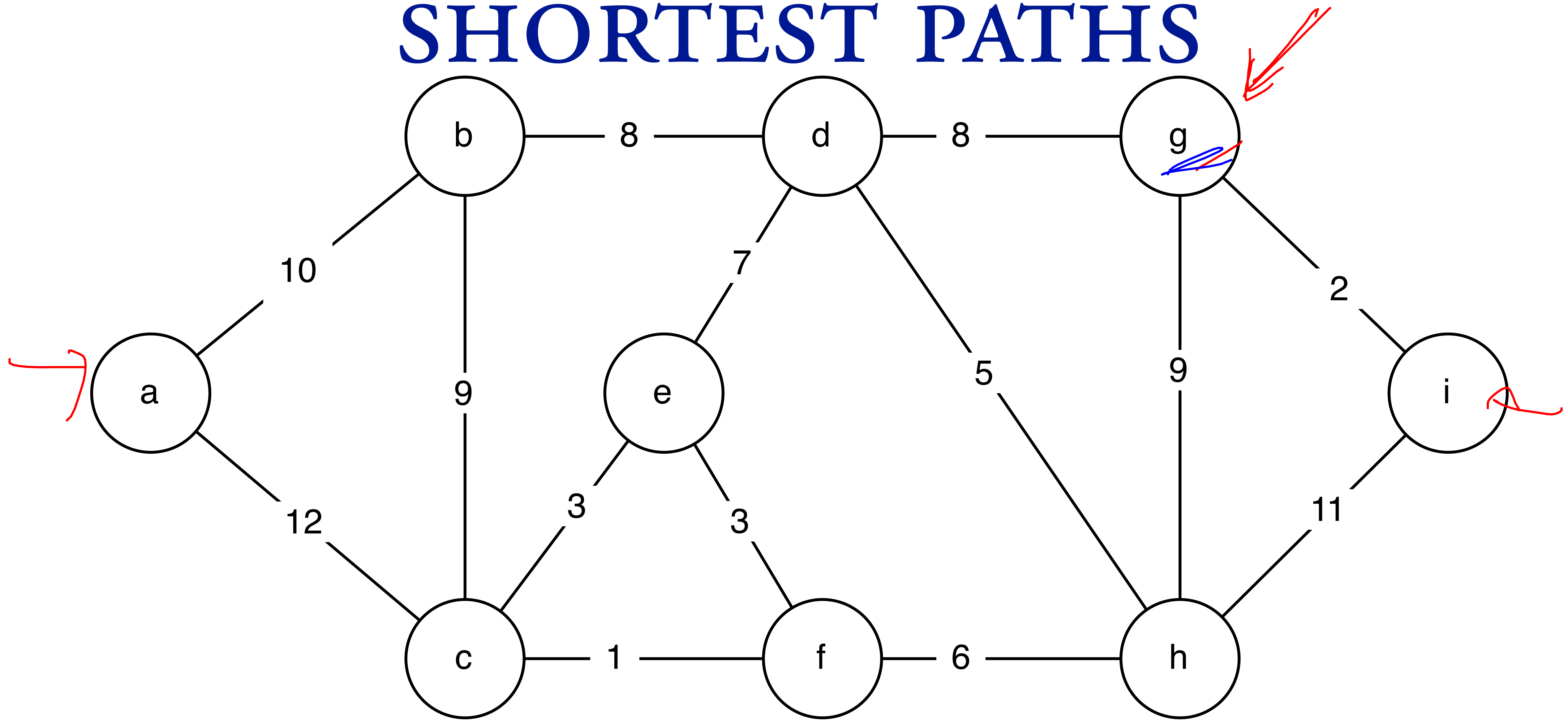
BFS

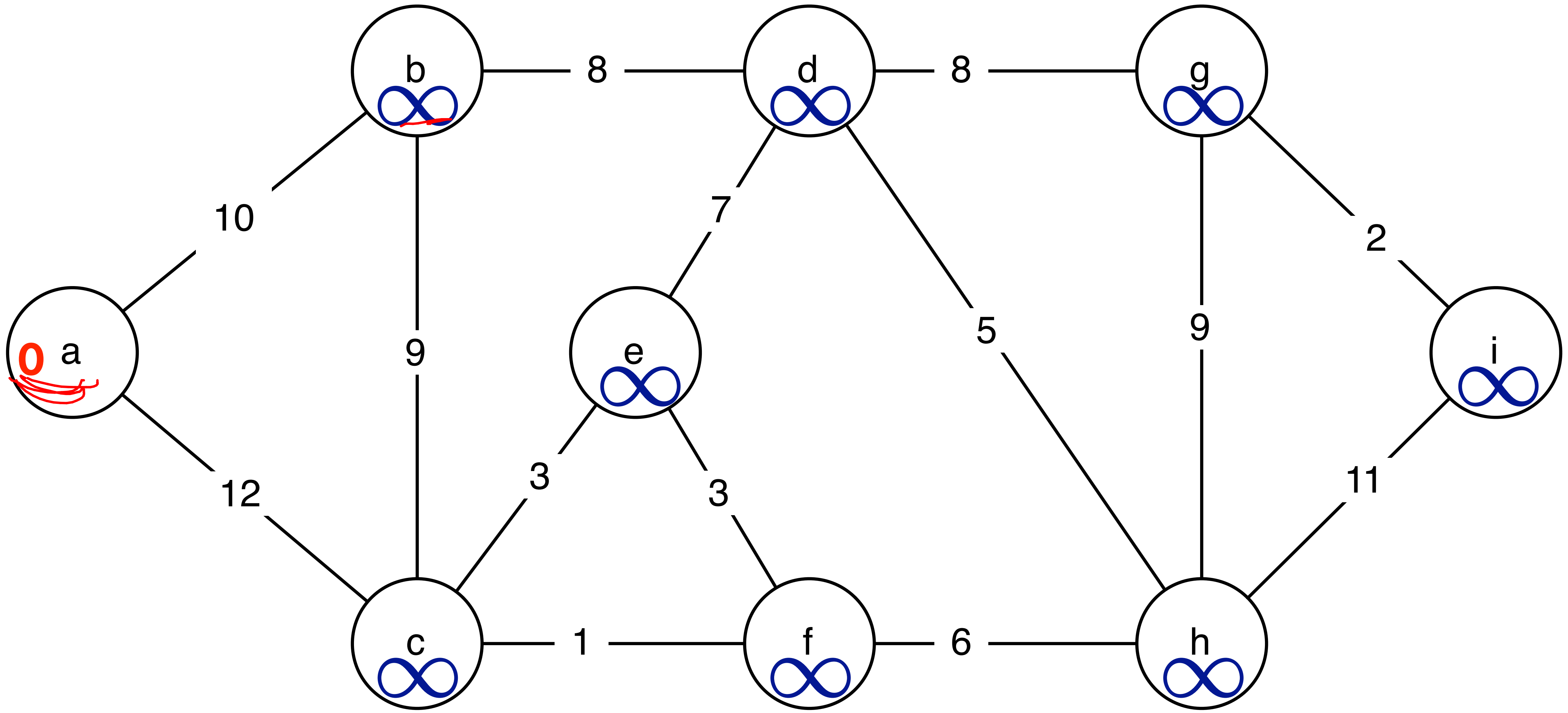
implementation

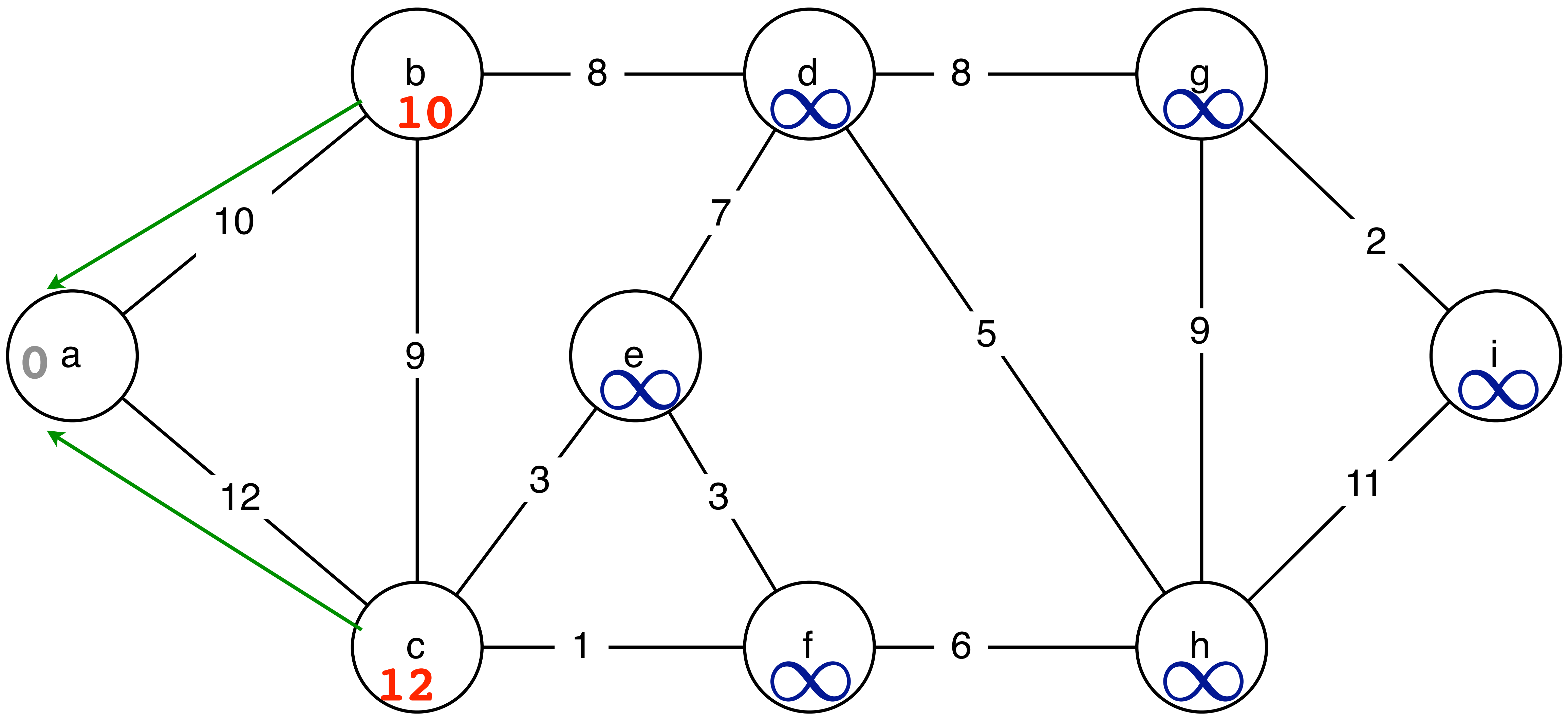
use a priority queue to keep track of light edges

	priority queue	fibonacci heap	
insert:	O(log n)	log n	
makequeue:	n	n	
extractmin:	O(log n)	log n	amortized
decreasekey:	O(log n)	O(1)	amortized

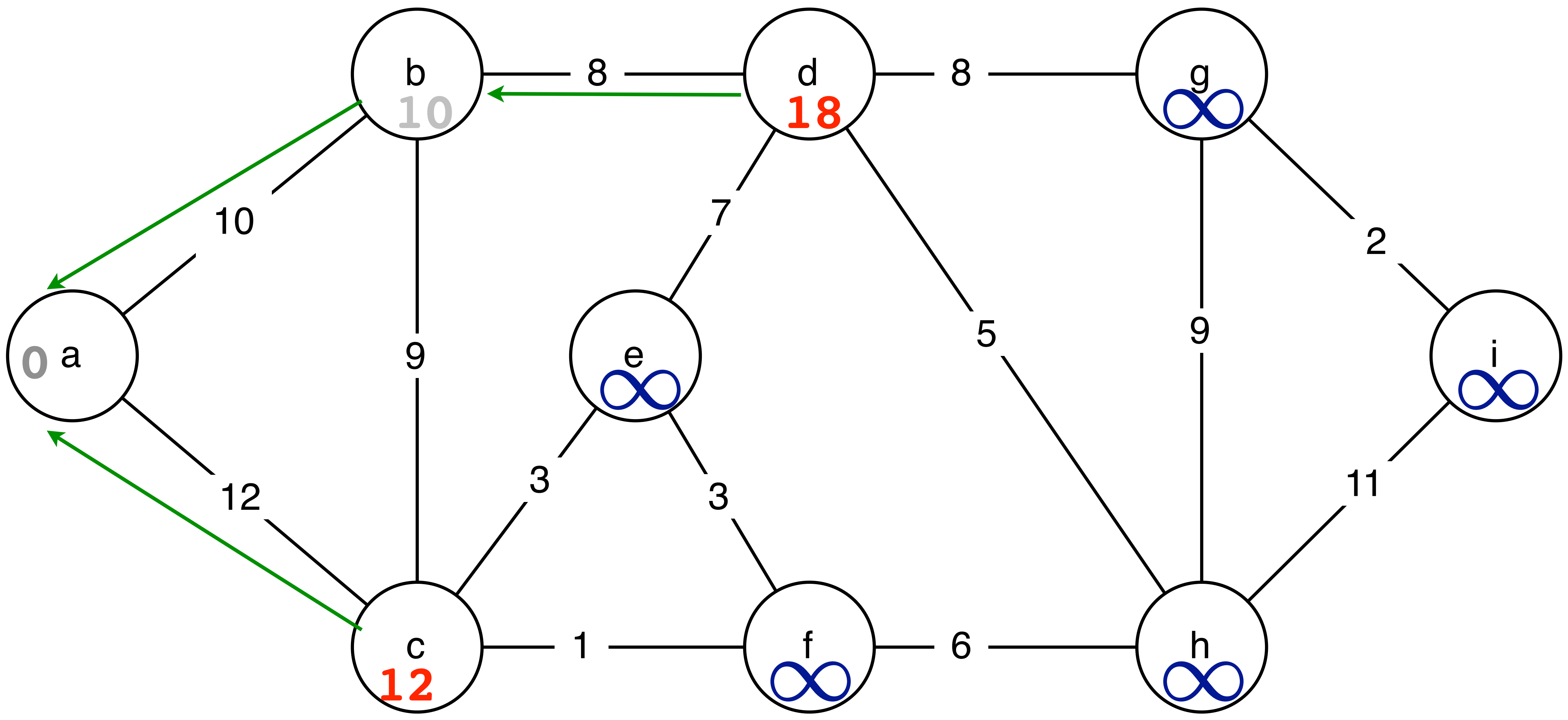
SHORTEST PATHS



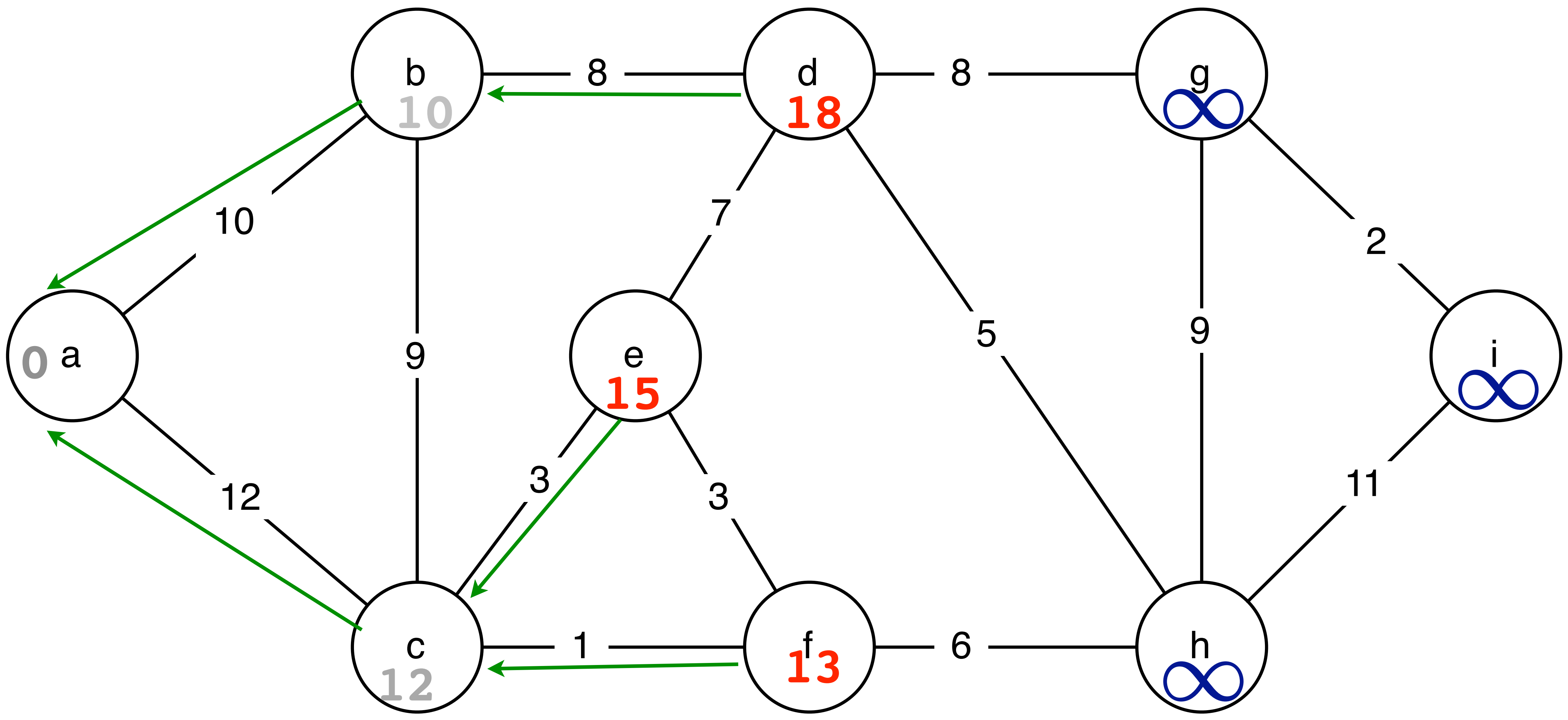

a
0
b
c
d
e
f
g
h
i
∞
10
12
9
8
8
7
3
3
1
6
5
9
2
11



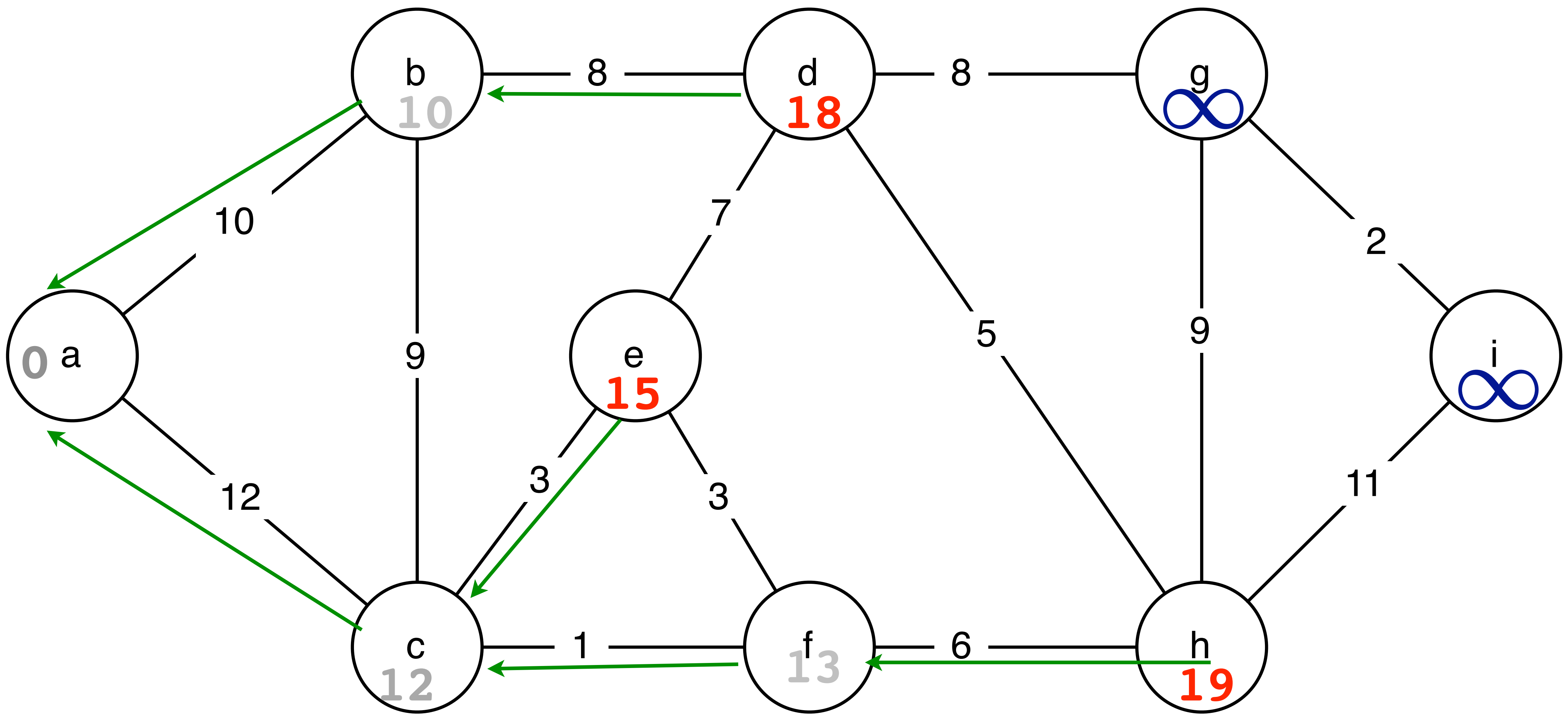
a
b
c
d
e
f
g
h
i
0
10
12
10
12
8
8
9
7
5
9
2
3
3
1
6
11



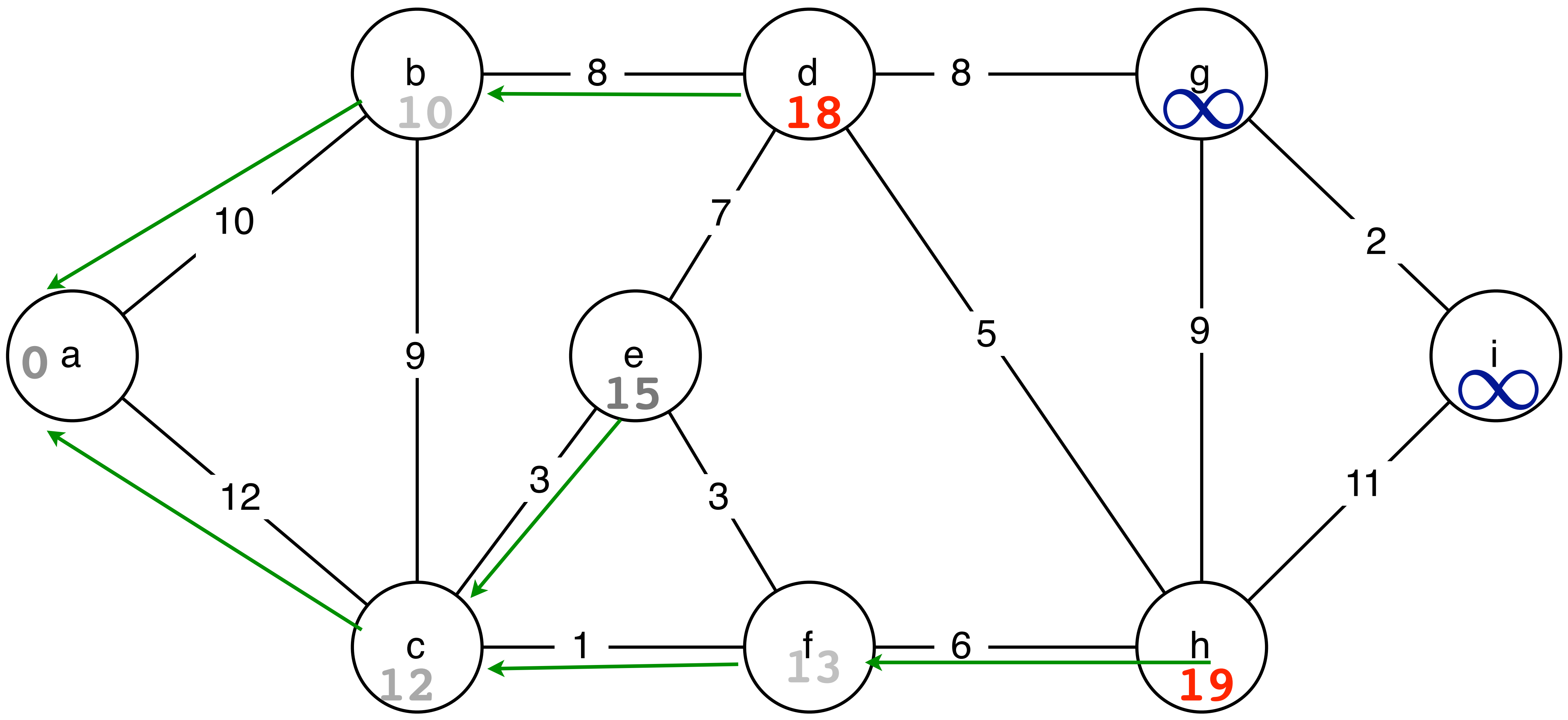
a
0
b
10
c
12
d
18
e
f
g
h
i
8
8
10
9
7
5
9
2
12
3
3
11
1
6



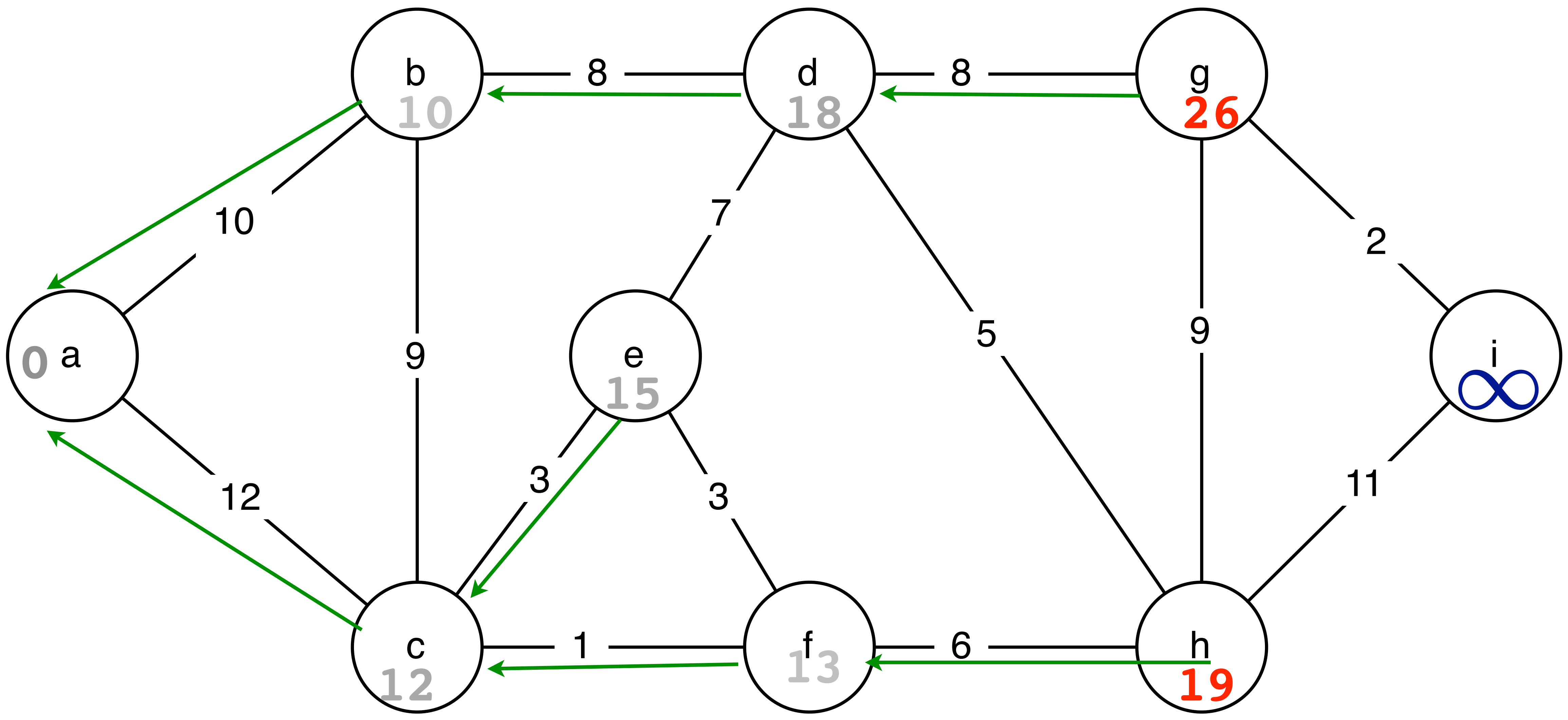
b
10
d
18
g
∞
a
0
e
15
i
∞
c
12
f
13
h
∞
8
8
10
7
2
9
5
9
12
3
3
11
1
6



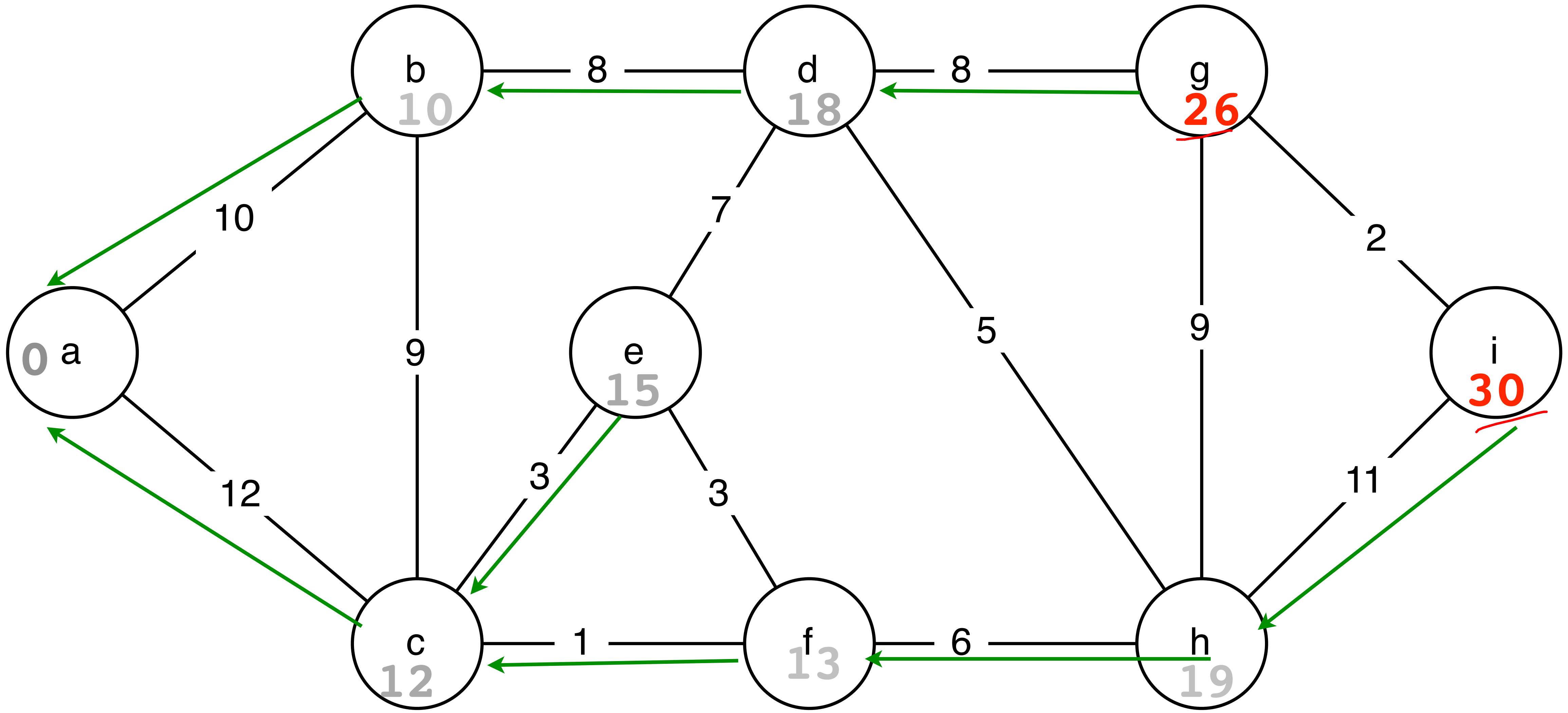
a
0
b
10
c
12
d
18
e
15
f
13
g
h
19
i
8
8
10
9
7
5
9
2
12
3
3
1
6
11



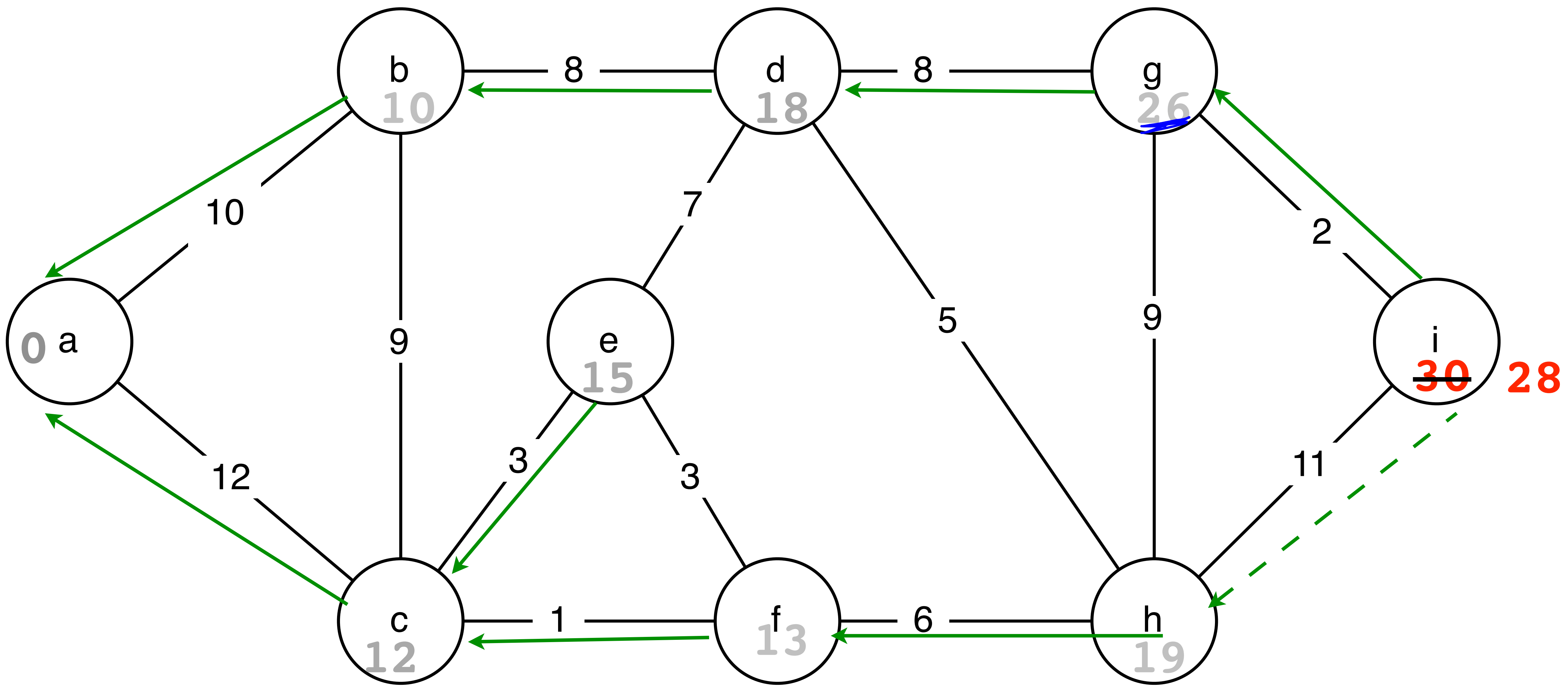
a
0
b
10
c
12
d
18
e
15
f
13
g
∞
h
19
i
∞
8
8
10
9
7
5
9
2
12
3
3
1
6
11



a
0
b
10
c
12
d
18
e
15
f
13
g
26
h
19
i
∞
8
8
10
9
7
5
9
2
12
3
3
1
6
11



a
0
b
10
c
12
d
18
e
15
f
13
g
26
h
19
i
30
8
8
10
9
7
5
9
2
12
3
3
11
1
6


a
0
b
10
c
12
d
18
e
15
f
13
g
26
h
19
i
30
28
8
8
10
9
7
5
9
2
12
3
3
11
1
6

DIJKSTRA$(G = (V, E), s)$

for all $v \in V$
    do $d_u \leftarrow \infty$
        $\pi_u \leftarrow$ NIL
$d_s \leftarrow 0$
$Q \leftarrow$ MAKEQUEUE(V) ▷ use d_u as key
while $Q \neq \emptyset$
    do $u \leftarrow$ EXTRACTMIN(Q)
        for each $v \in Adj(u)$
            do if $d_v > d_u + w(u, v)$
                then $d_v \leftarrow d_u + w(u, v)$
                    $\pi_v \leftarrow u$
                    DECREASEKEY(Q, v)

```
DIJKSTRA(G = (V, E), s)
for all v ∈ V
        do d_u ← ∞
           π_u ← NIL
d_s ← 0
Q ← MAKEQUEUE(V)    ▷ use d_u as key
while Q ≠ ∅
        do u ← EXTRACTMIN(Q)
           for each v ∈ Adj(u)
               do if d_v > d_u + w(u, v)
                     then d_v ← d_u + w(u, v)
                          π_v ← u
                          DECREASEKEY(Q, v)
```

```
PRIM(G = (V, E))
Q ← ∅    ▷ Q is a Priority Queue
Initialize each v ∈ V with key k_v ← ∞, π_v ← NIL
Pick a starting node r and set k_r ← 0
Insert all nodes into Q with key k_v.
while Q ≠ ∅
        do u ← EXTRACT-MIN(Q)
           for each v ∈ Adj(u)
               do if v ∈ Q and w(u, v) < k_v
                     then π_v ← u
                          DECREASE-KEY(Q, v, w(u, v))    ▷ Sets k_v ← w(
```

RUNNING TIME

```
DIJKSTRA(G = (V, E), s)
for all v ∈ V
      do d_u ← ∞
         π_u ← NIL
d_s ← 0
Q ← MAKEQUEUE(V)        ▷ use d_u as key
while Q ≠ ∅
      do u ← EXTRACTMIN(Q)
         for each v ∈ Adj(u)
               do if d_v > d_u + w(u, v)
                     then d_v ← d_u + w(u, v)
                          π_v ← u
                          DECREASEKEY(Q, v)
```

$V \log(v)$

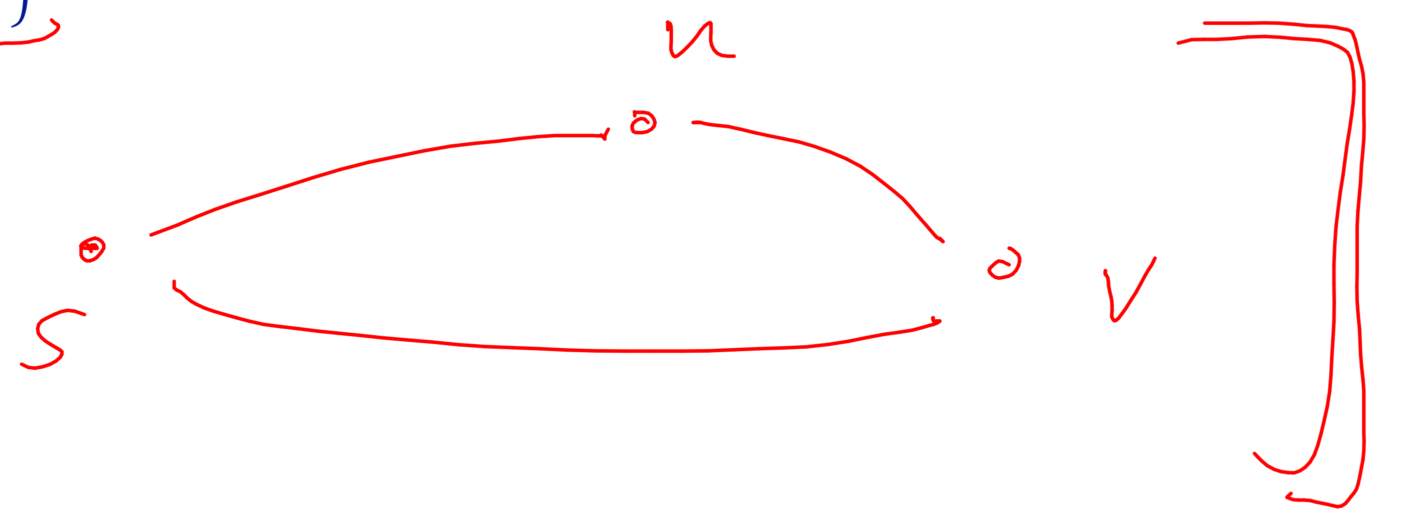
$E \log(v)$

$\Theta(E \log V)$

WHY DOES DIJKSTRA WORK?

TRIANGLE INEQUALITY: $\forall (u, v) \in E, \ \delta(s, v) \leq \delta(s, u) + w(u, v)$

UPPER BOUND: $d_v \geq \delta(s, v)$



Let set S consist of the nodes not in Q. $S = \emptyset$ at line 4.

P1: $\forall x \in S \quad d_x = \delta(s, x)$.

P1 holds @ line 4. Spse it holds for the first i iterations of the loop.

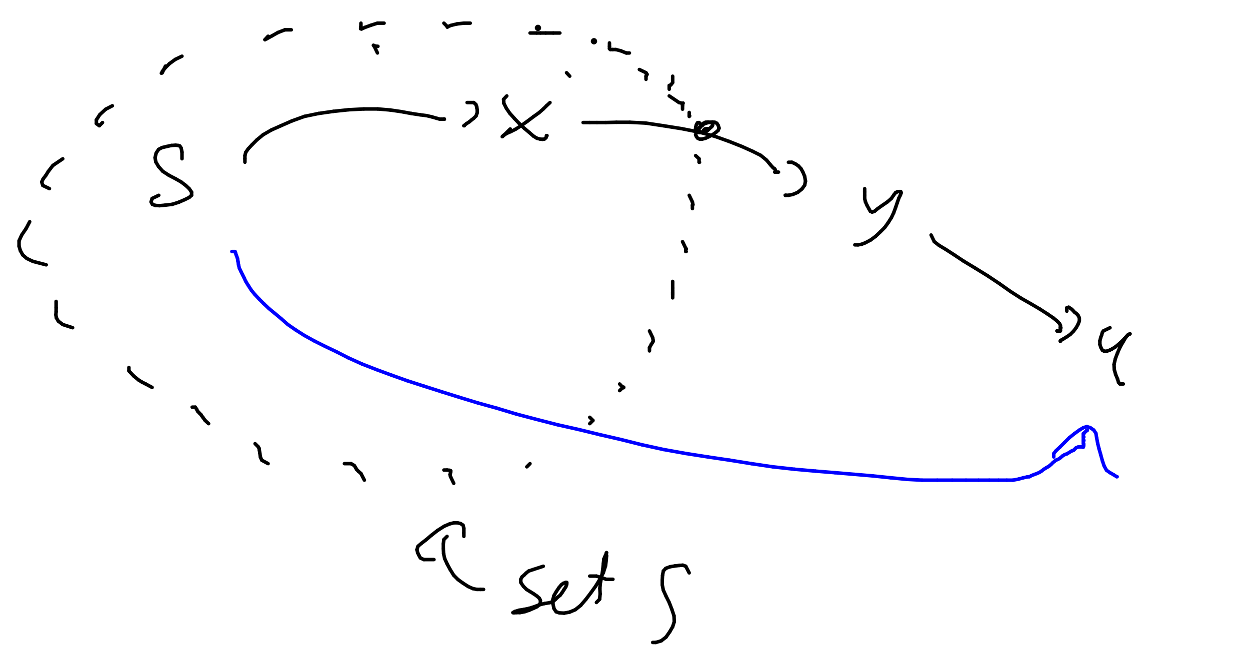
Consider iteration $i+1$:

- Node u is removed from Q in line 6. so u is added to S.

* Spse that $d_u \neq \delta(s,u)$ (1). Recall that $d_u \geq \delta(s,u)$ (2)

This implies $\exists$ some path from $s \rightsquigarrow u$. If not, then $\delta(s,u) = \infty$, which contradicts (1) (2)

Consider any path p from s to u and let x be the last node in S along this path.



$$\ell(p) = \delta(s,x) + w(x,y) + \ell(y \rightsquigarrow u)$$

$$\geq d_y + \ell(y \rightsquigarrow u)$$

$$\geq d_u + \ell(y \rightsquigarrow u) \geq d_u \geq \delta(s,u)$$

↑ L7 ↑ $\ell(y,u) \geq 0$ ↑ (2)

$$\ell(p) \geq d_u \geq \delta(s,u)$$

$\Rightarrow$ Thus $d_u = \delta(s,u)$. SANDWICH.

This contradicts our assumption $*$. $\Rightarrow$ $d_u = \delta(s,u)$.

b ∞

h ∞

c ∞

0

i ∞

$$\delta(s,v) \leq \delta(s,u) + w(u,v) \quad \text{for all } u \in V$$

BREADTH FIRST SEARCH

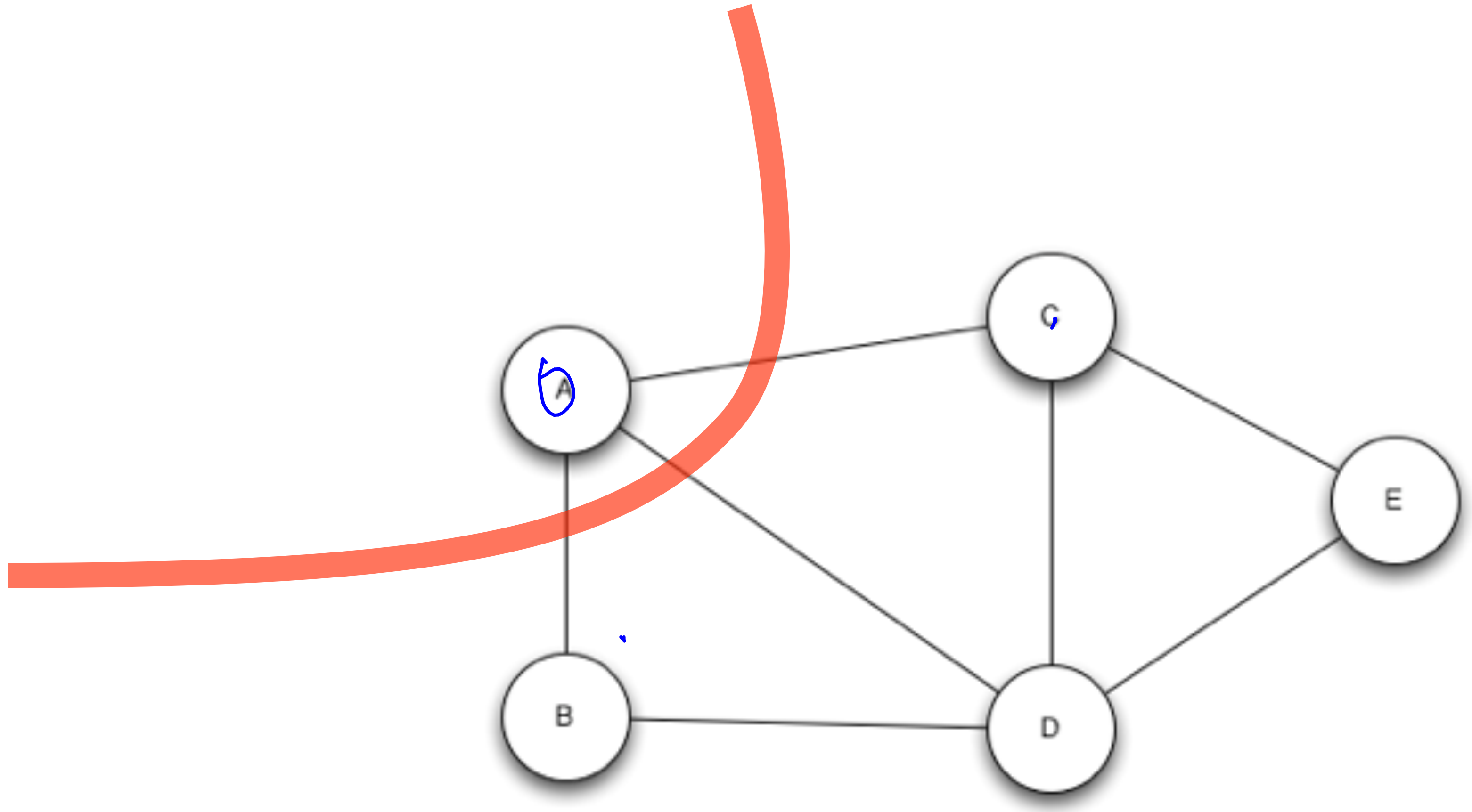
INPUT: $G = (V, E), s$

OUTPUT: $\forall v \in V \quad d_v = \delta(s, v)$

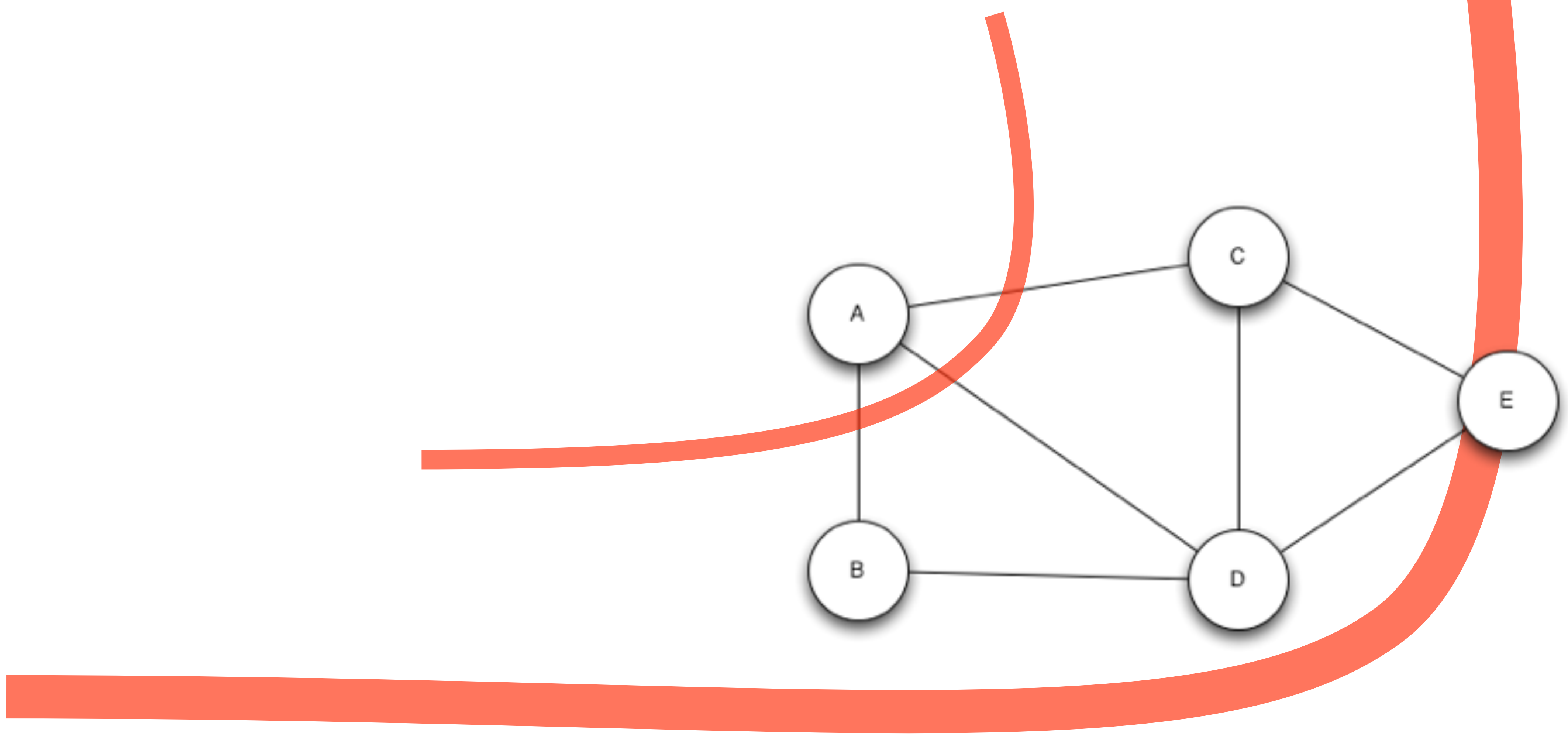
SMALLEST # OF EDGES FROM S TO V

$W : E \to \mathbb{R}$ $\quad w(x,y) = 1 \quad \forall (x,y) \in E$

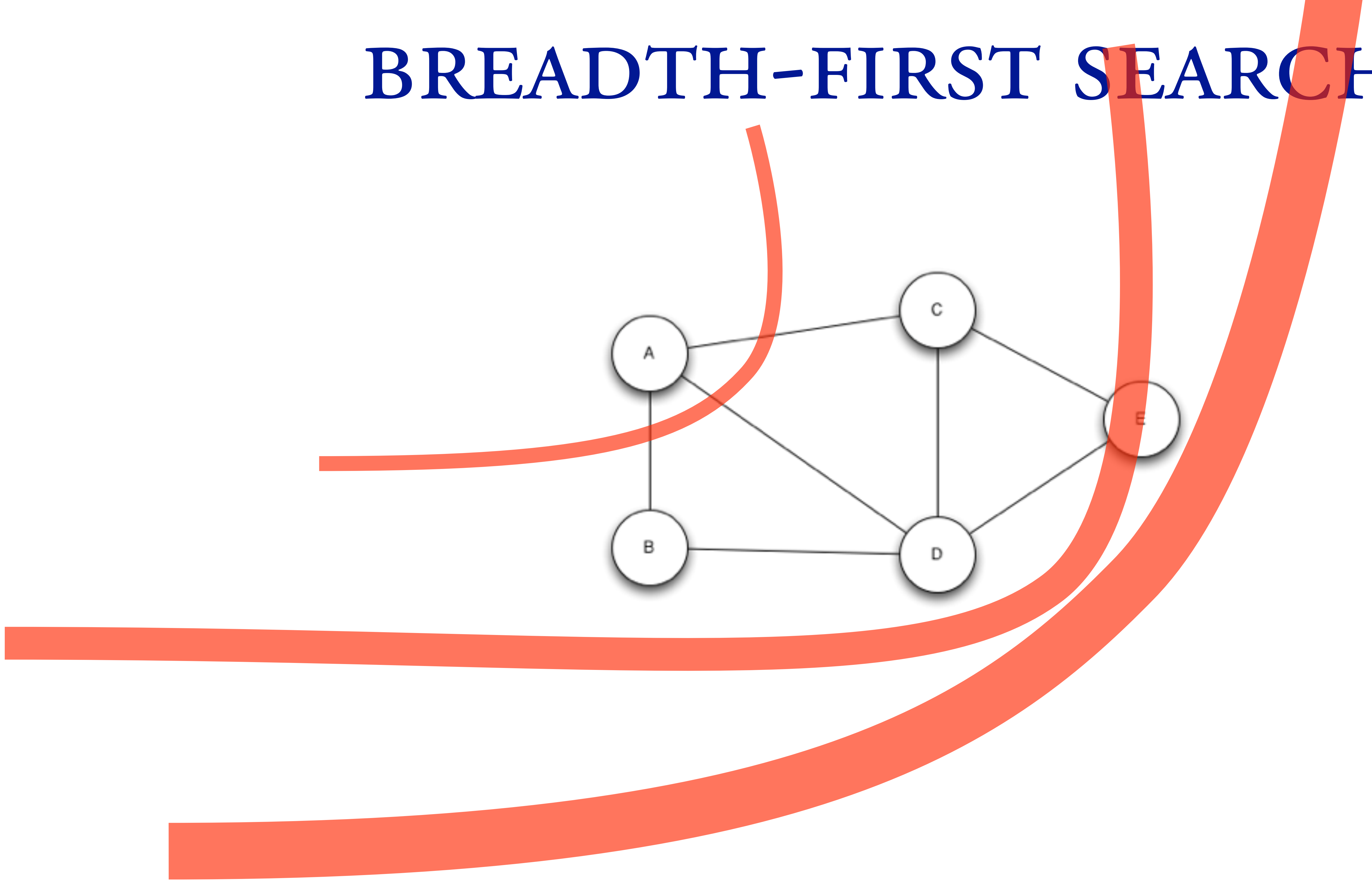
BREADTH-FIRST SEARCH



BREADTH-FIRST SEARCH



BREADTH-FIRST SEARCH

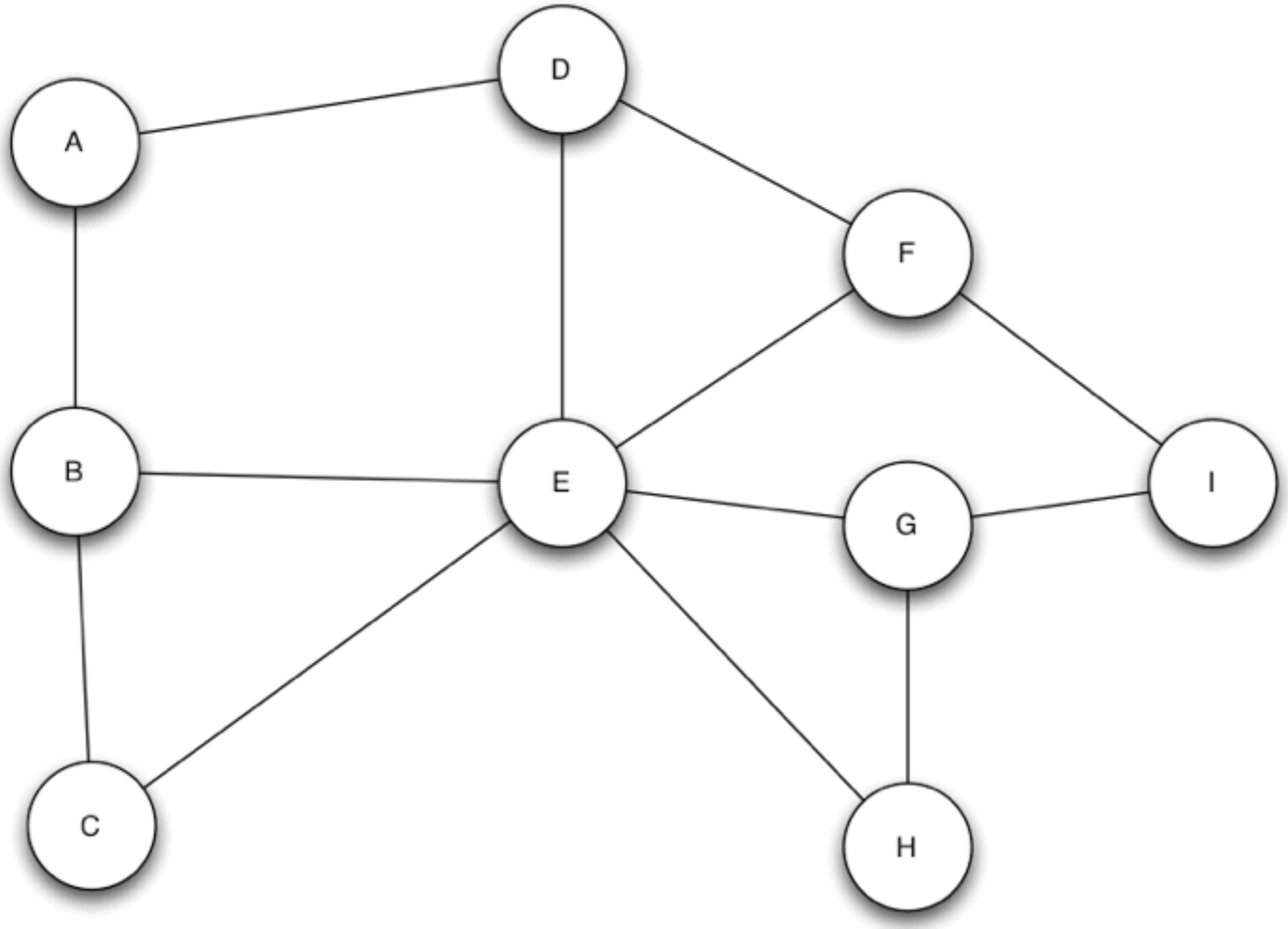


BREADTH FIRST SEARCH

INPUT: $G = (V, E), s$

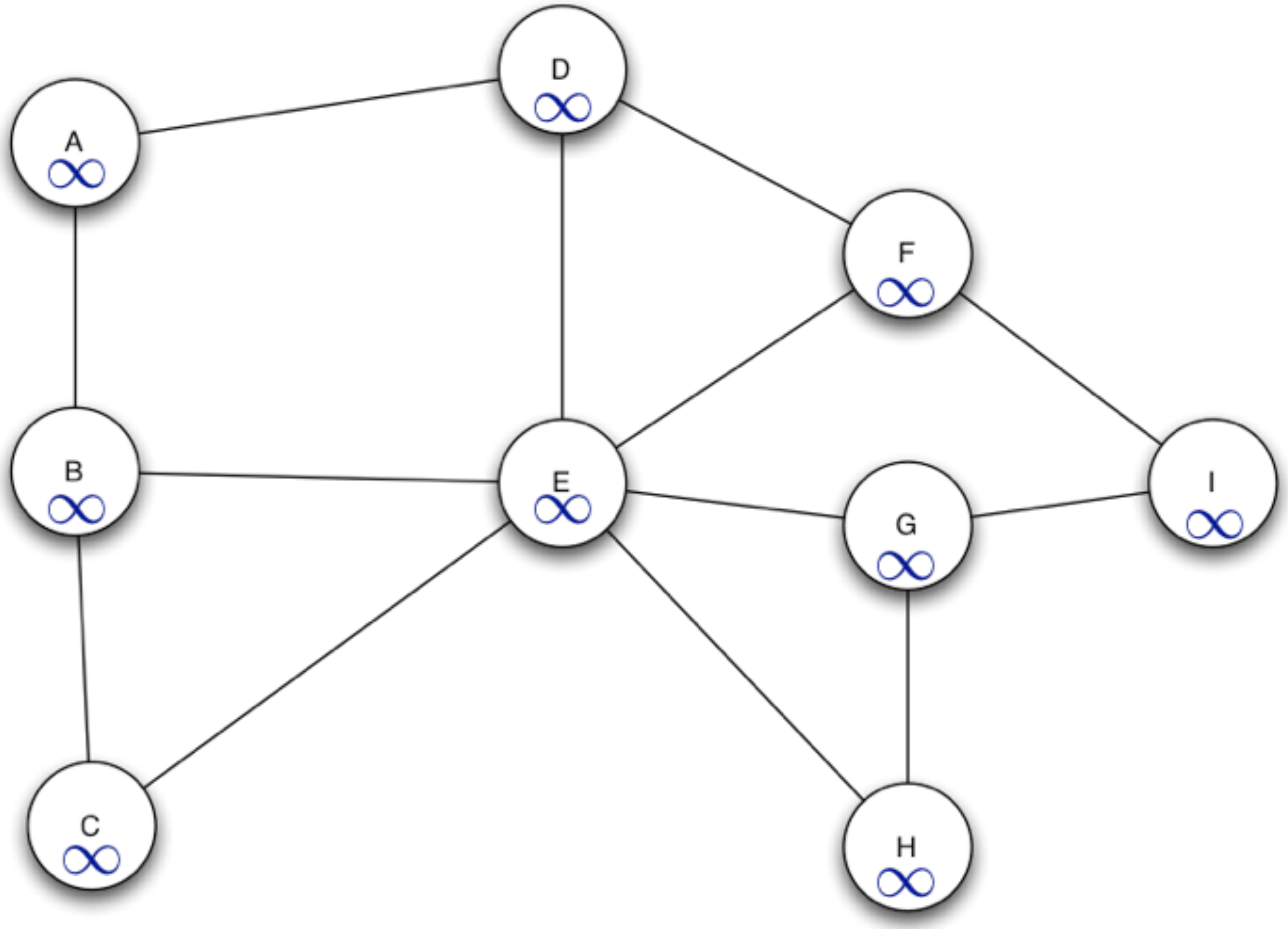
OUTPUT: d_v SMALLEST # OF EDGES FROM S TO V $\forall v \in V$

BFS(G, A)



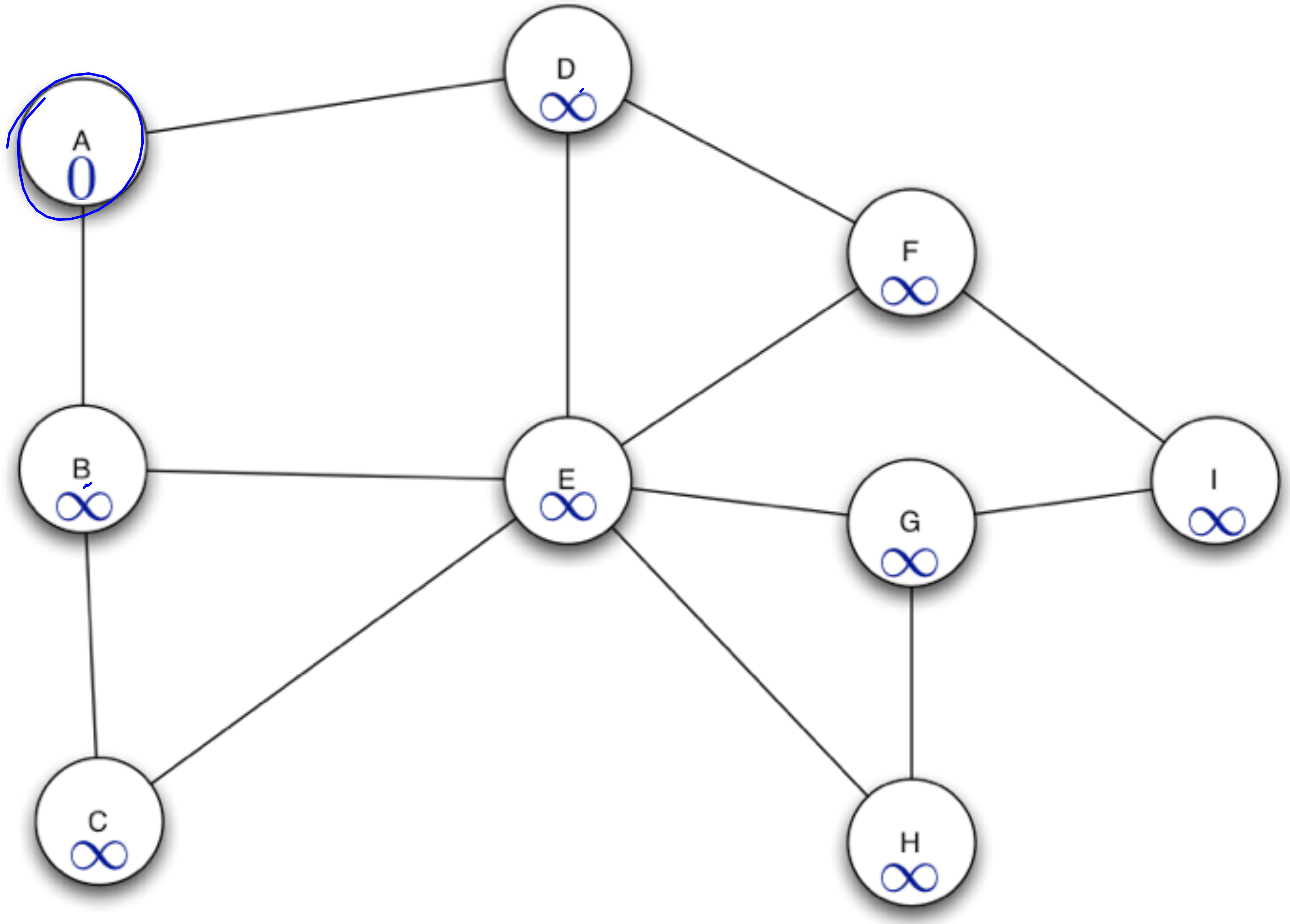
Q

BFS(G, A)

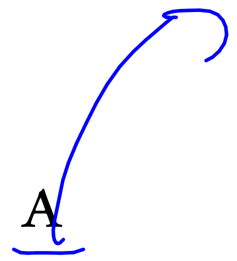


Q

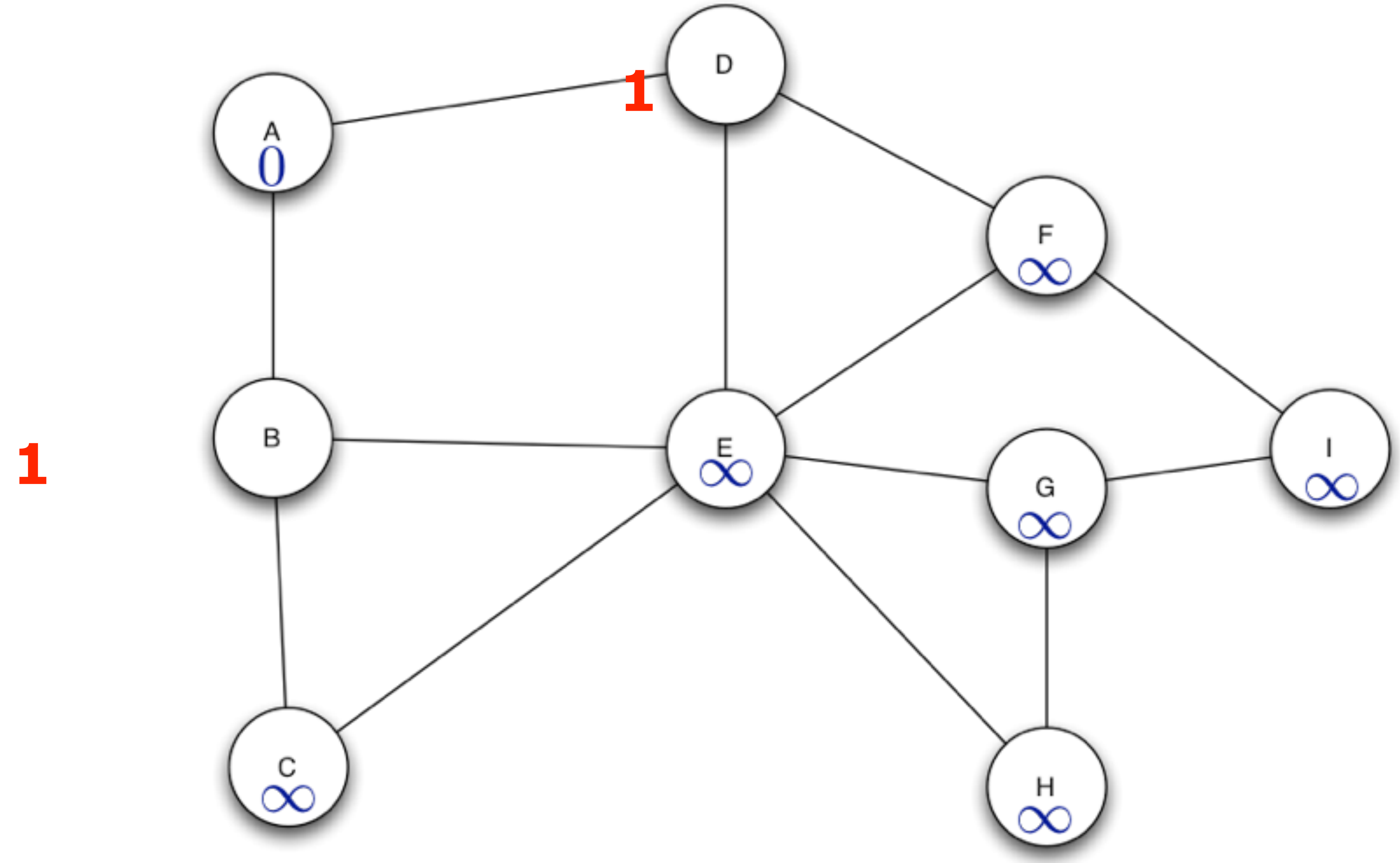
BFS(G, A)



Q



BFS(G, A)



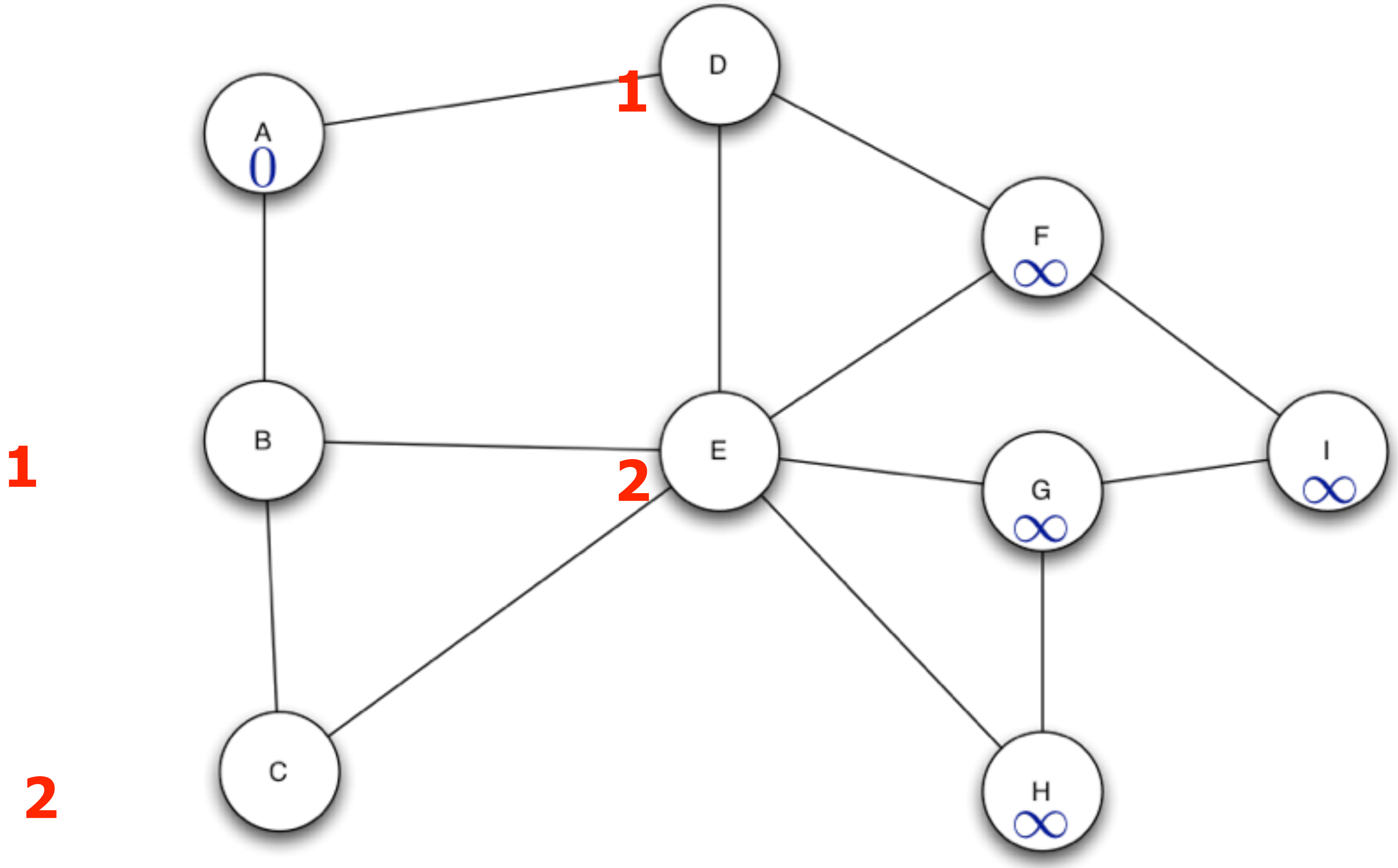
Q

A

B

D

BFS(G, A)



Q

A

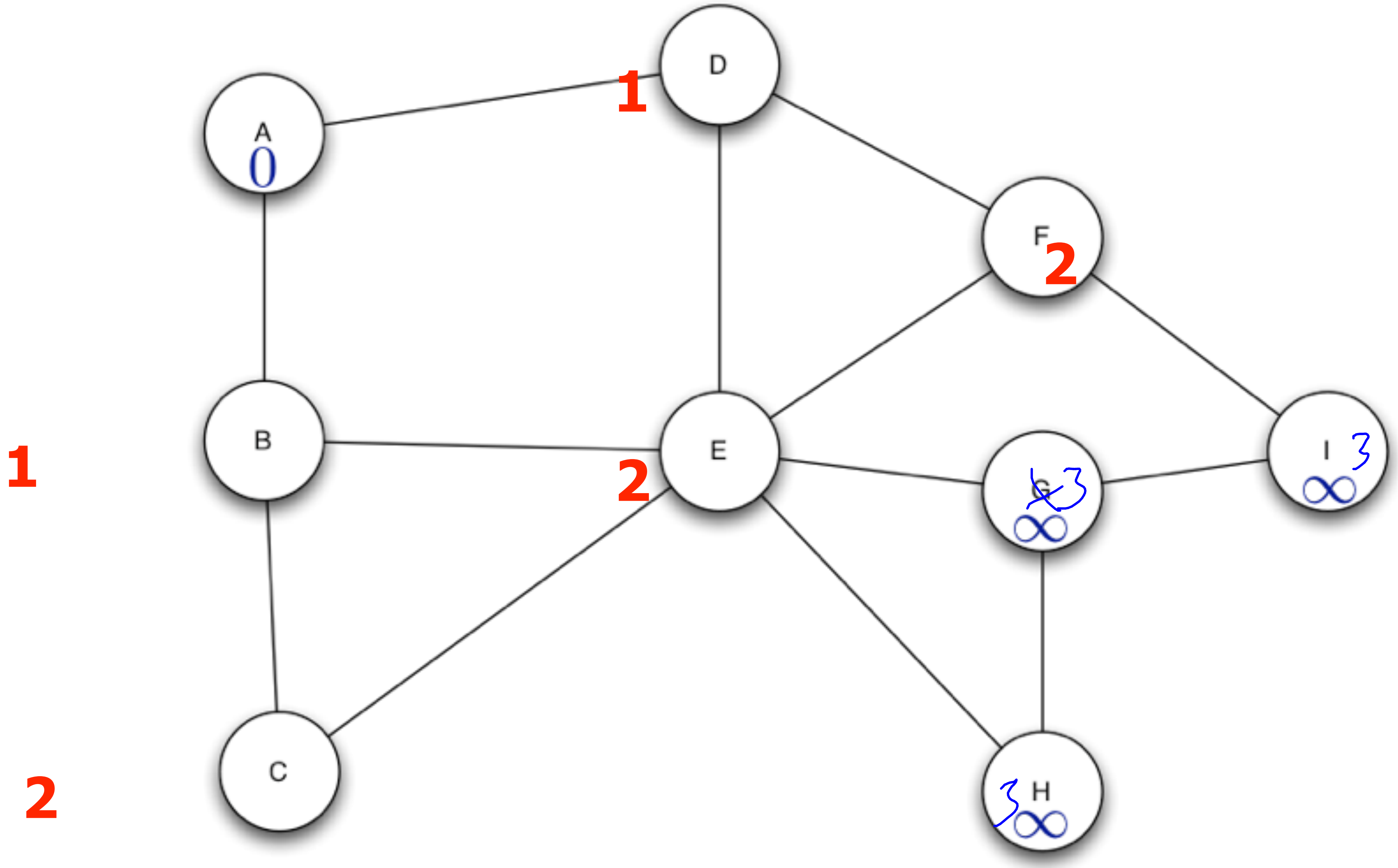
B

D

C

E

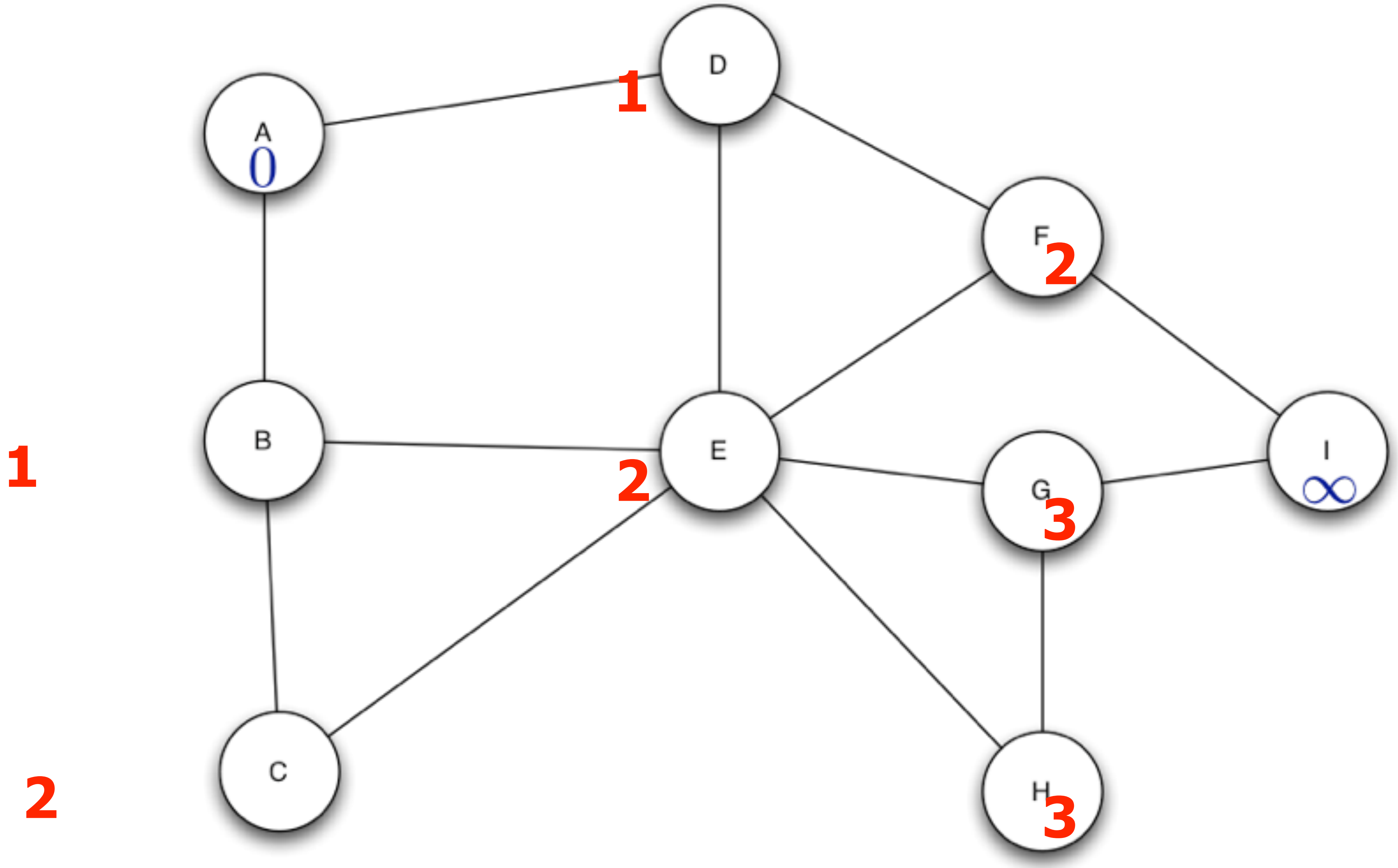
BFS(G, A)



Q

- ~~A~~
- ~~B~~
- ~~D~~
- ~~C~~
- ~~E~~
- ~~F~~
- H
- G
- I

BFS(G, A)



Q

A

B

D

C

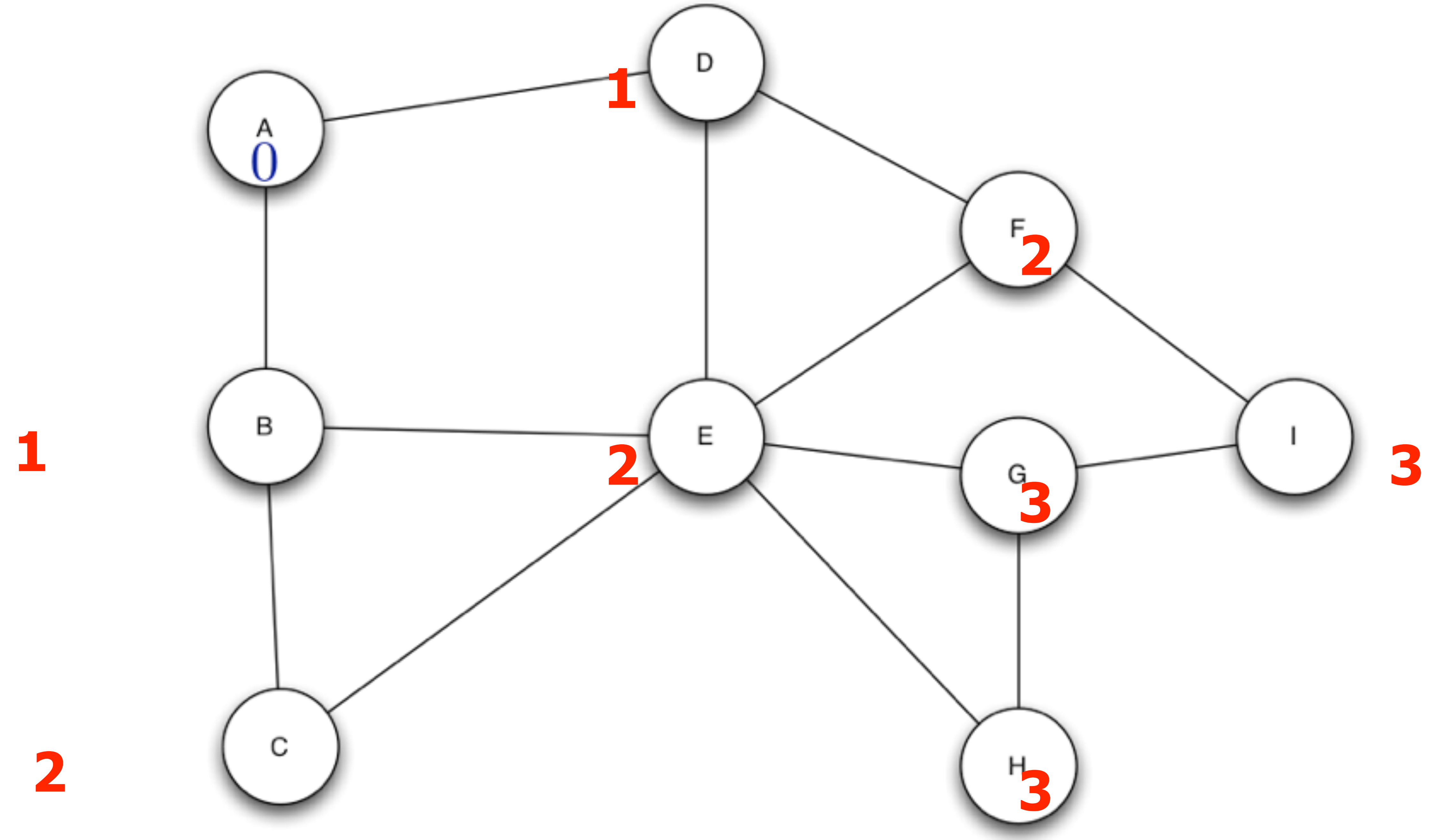
E

F

G

H

BFS(G, A)



Q

A

B

D

C

E

F

G

H

BFS(G, A)

BREADTH FIRST SEARCH

BFS(V, E, s)

for each $u \in V - \{s\}$
  do $d[u] \leftarrow \infty$
$d[s] \leftarrow 0$
$Q \leftarrow \emptyset$
ENQUEUE(Q, s)
while $Q \neq \emptyset$
  do $u \leftarrow$ DEQUEUE(Q) → $\Theta(1)$
   for each $v \in Adj[u]$
    do if $d[v] = \infty$
     then $d[v] \leftarrow d[u] + 1$
      ENQUEUE(Q, v)

$\Theta(1)$

$\Theta(V + E) = \Theta(E)$

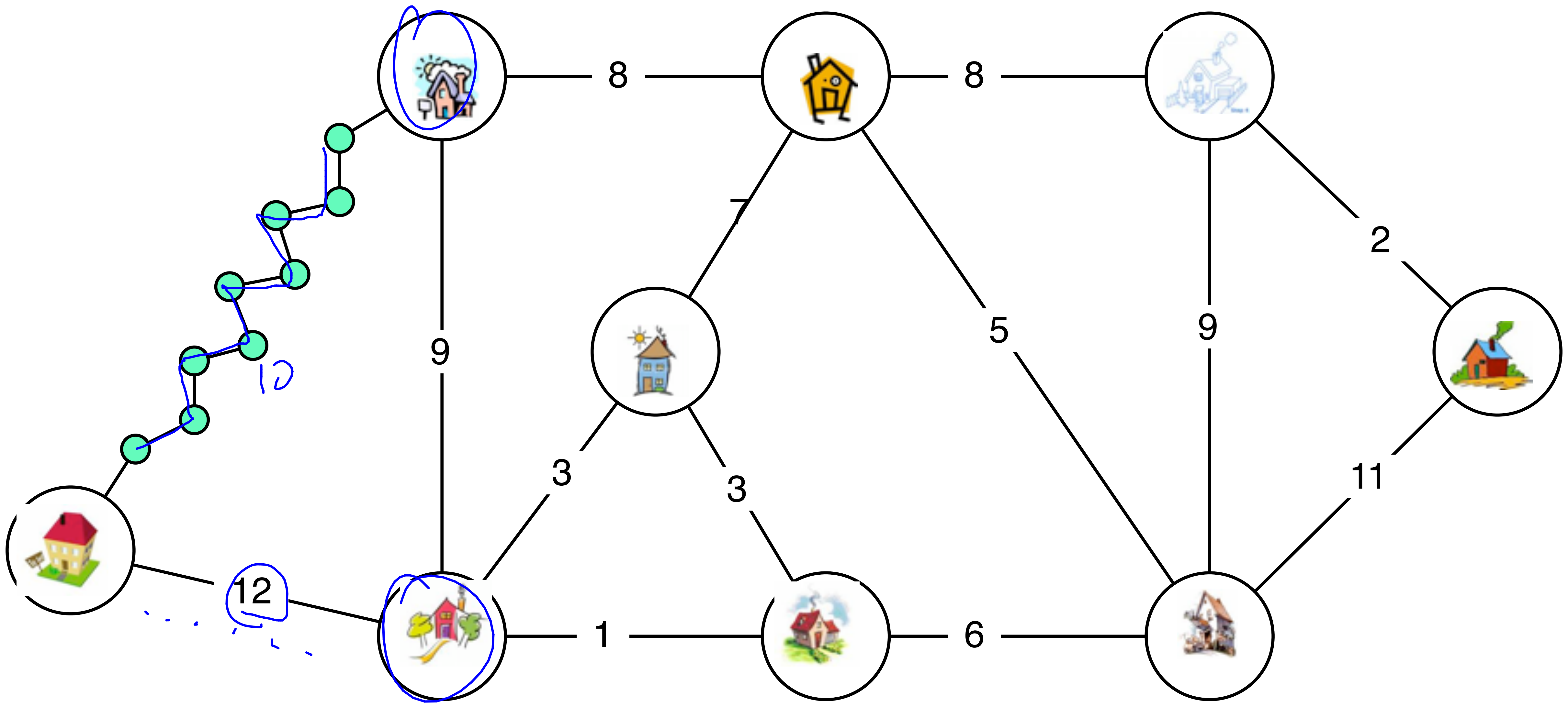
BFS THEOREM

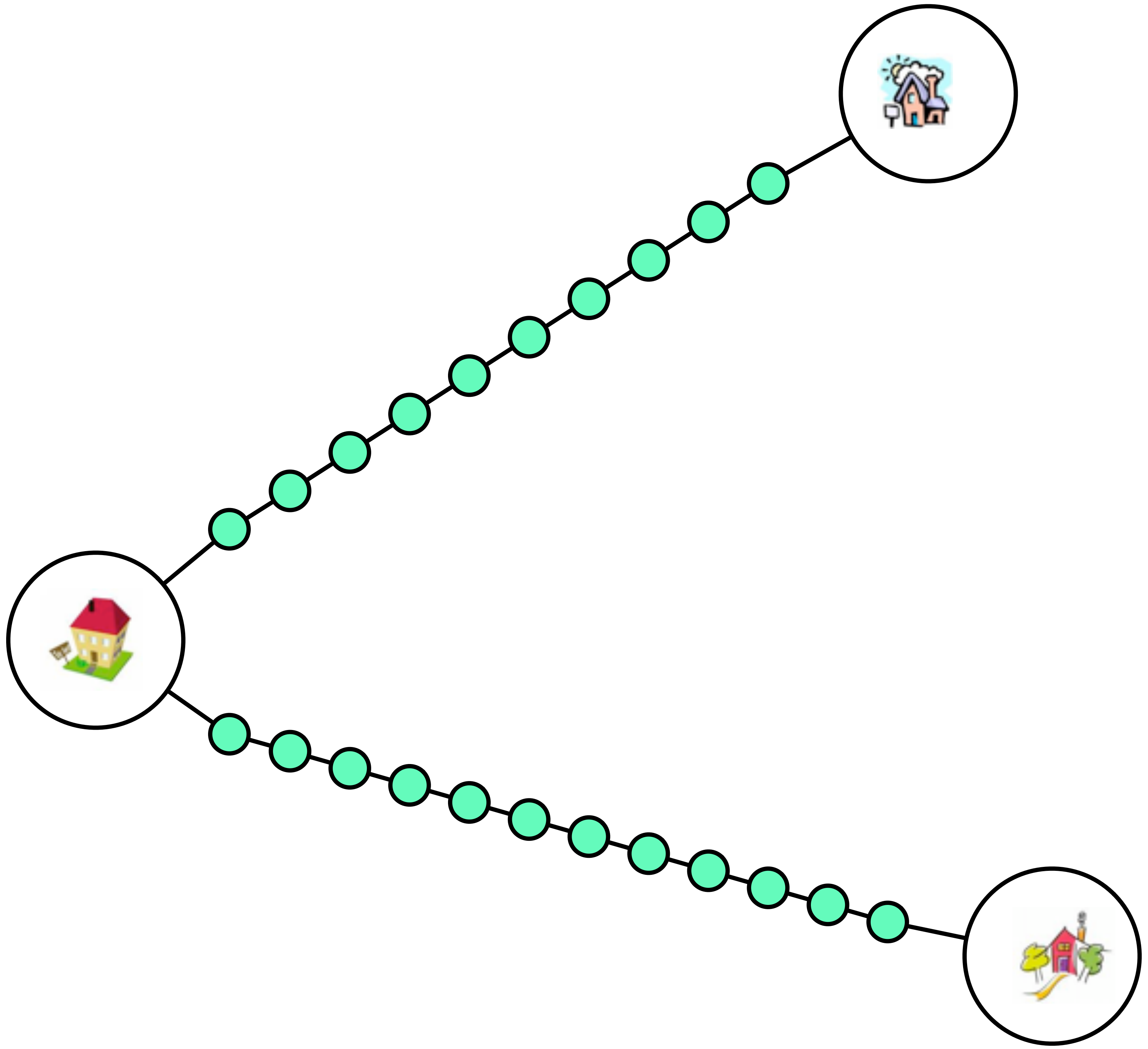
When BFS(G,s) terminates, then $d_x = \delta(s,x)$ for all $x \in V$. when $w(x,y)=1 \ \forall (x,y) \in E$.

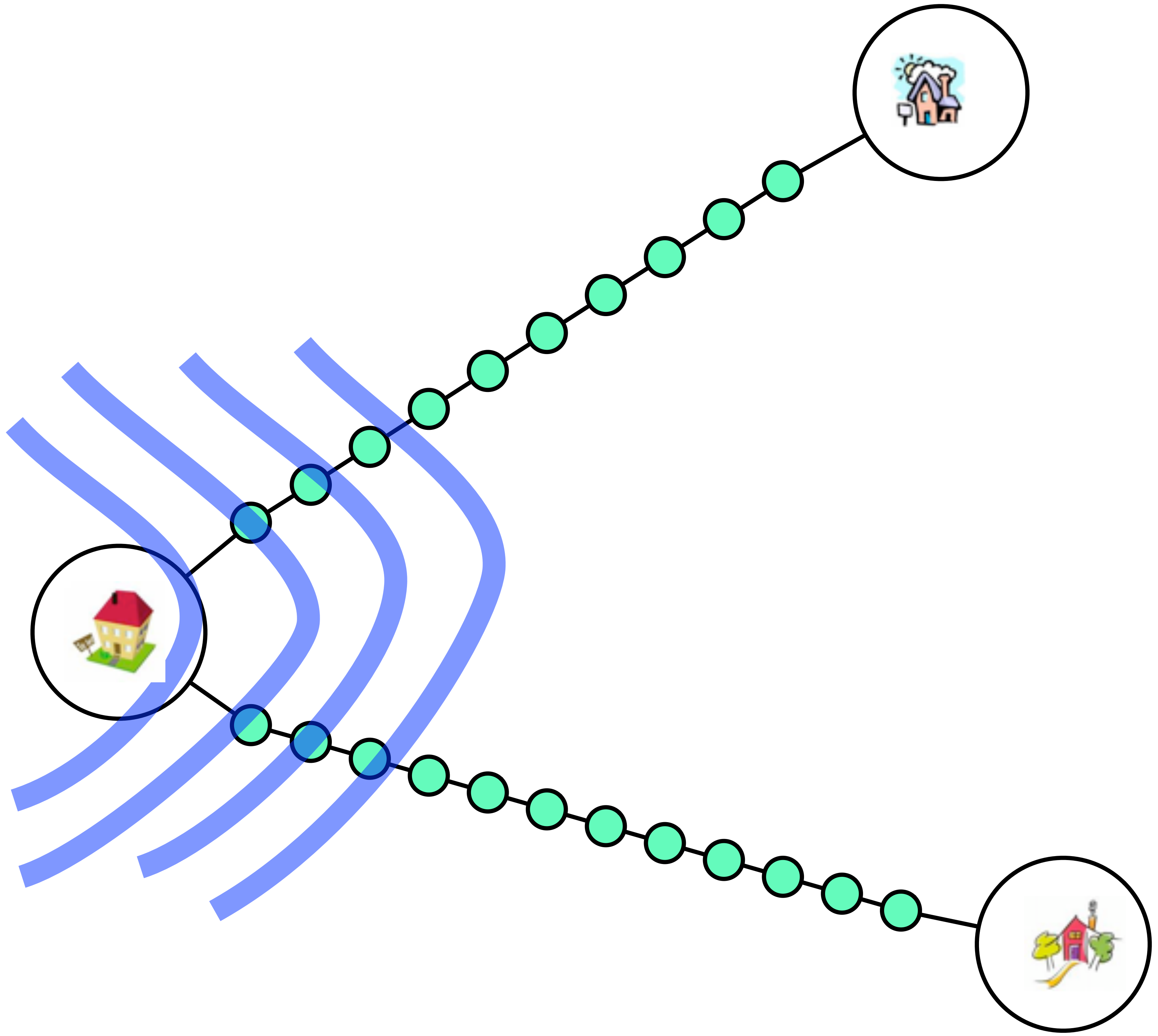
```
BFS(V, E, s)
for each u ∈ V − {s}
    do d[u] ← ∞
d[s] ← 0
Q ← ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    do u ← DEQUEUE(Q)
       for each v ∈ Adj[u]
           do if d[v] = ∞
                 then d[v] ← d[u] + 1
                      ENQUEUE(Q, v)
```

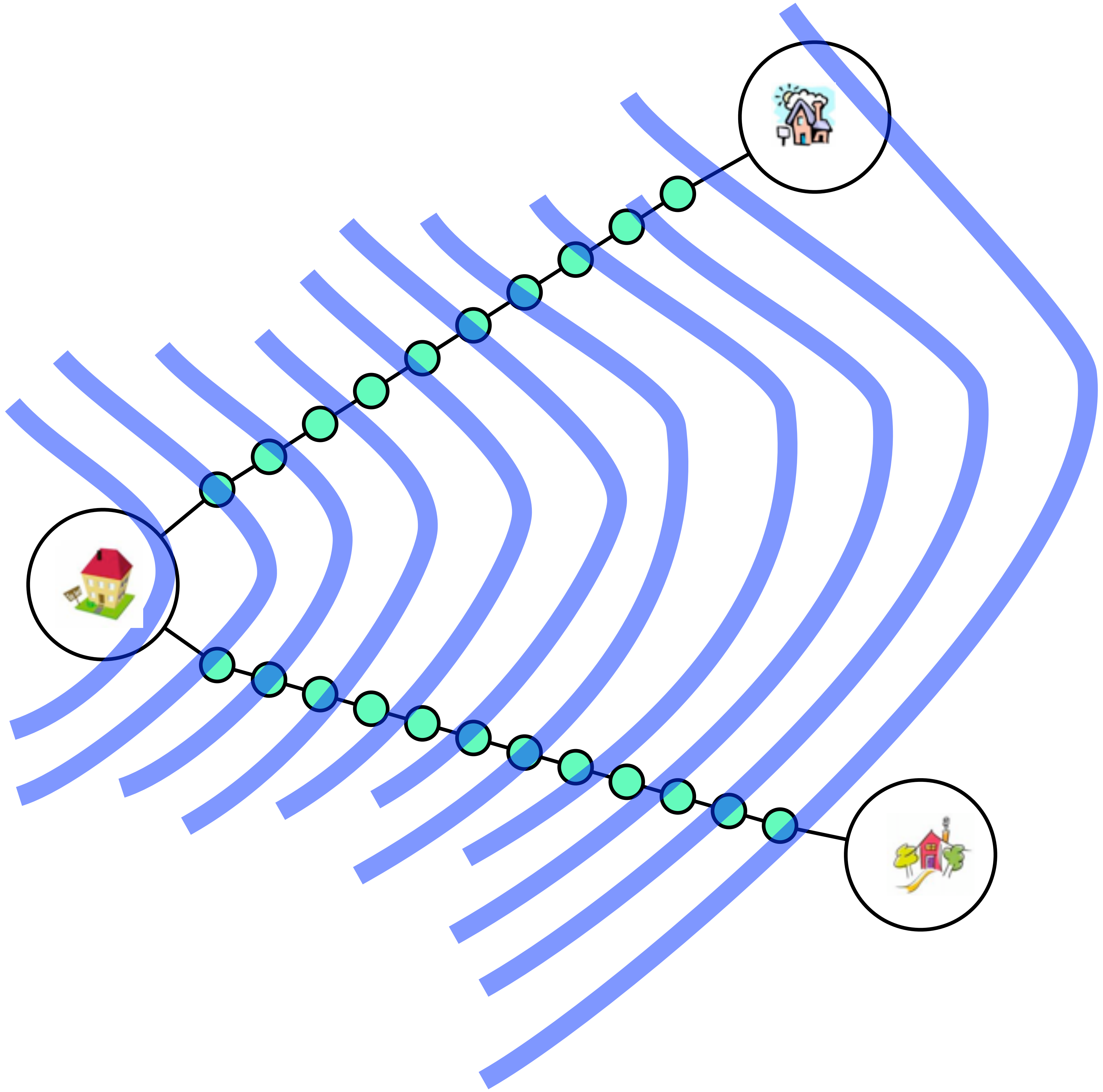
```
DIJKSTRA(G = (V, E), s)
for all v ∈ V
      do d_u ← ∞
         π_u ← NIL
d_s ← 0
Q ← MAKEQUEUE(V)    ▷ use d_u as key
while Q ≠ ∅
      do u ← EXTRACTMIN(Q)
         for each v ∈ Adj(u)
             do if d_v > d_u + w(u, v)
                   then d_v ← d_u + w(u, v)
                        π_v ← u
                        DECREASEKEY(Q, v)
```

BFS









SHORTEST PATHS

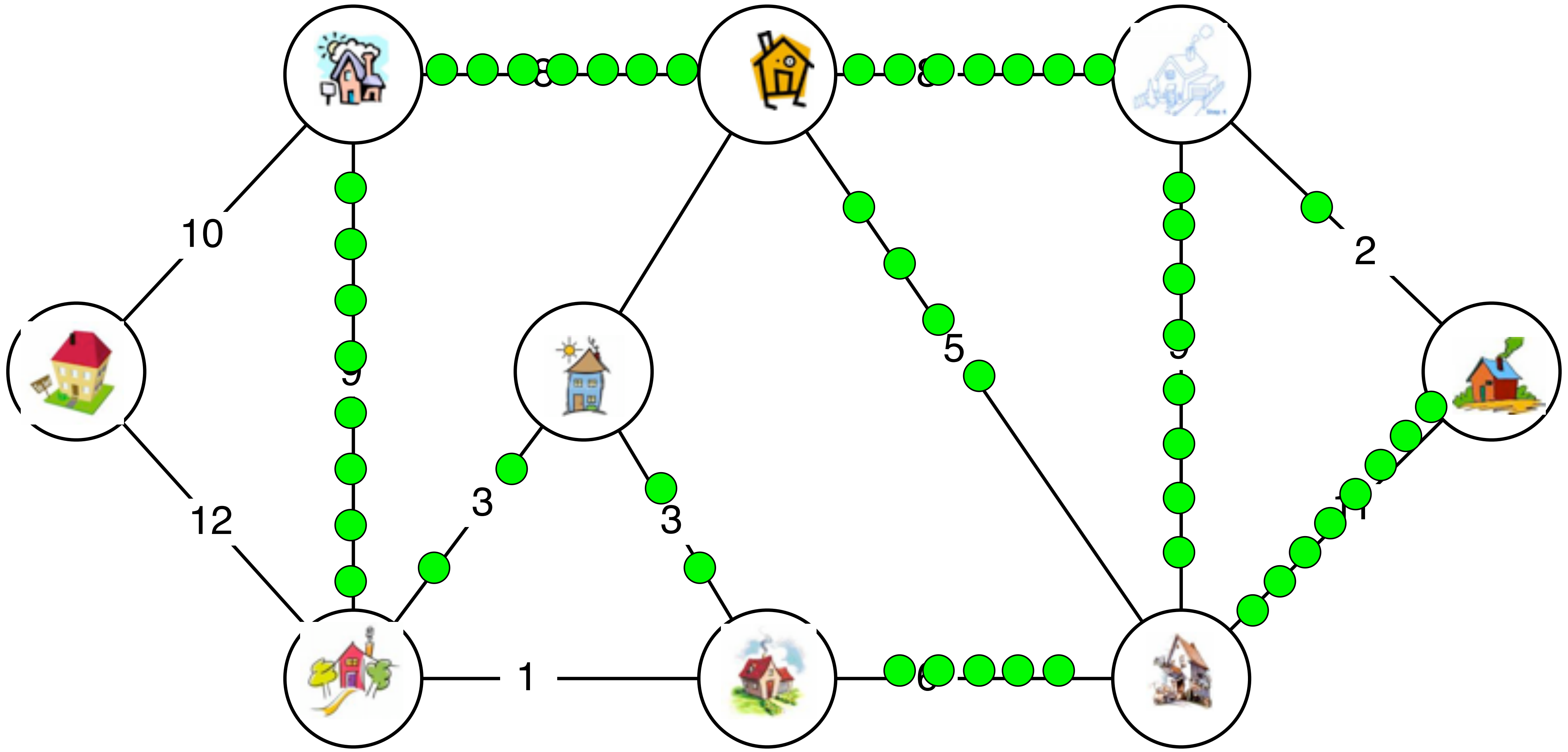


image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

WHAT ABOUT NEGATIVE EDGE WEIGHTS?

XE Live Exchange Rates

Change / Remove a currency ...

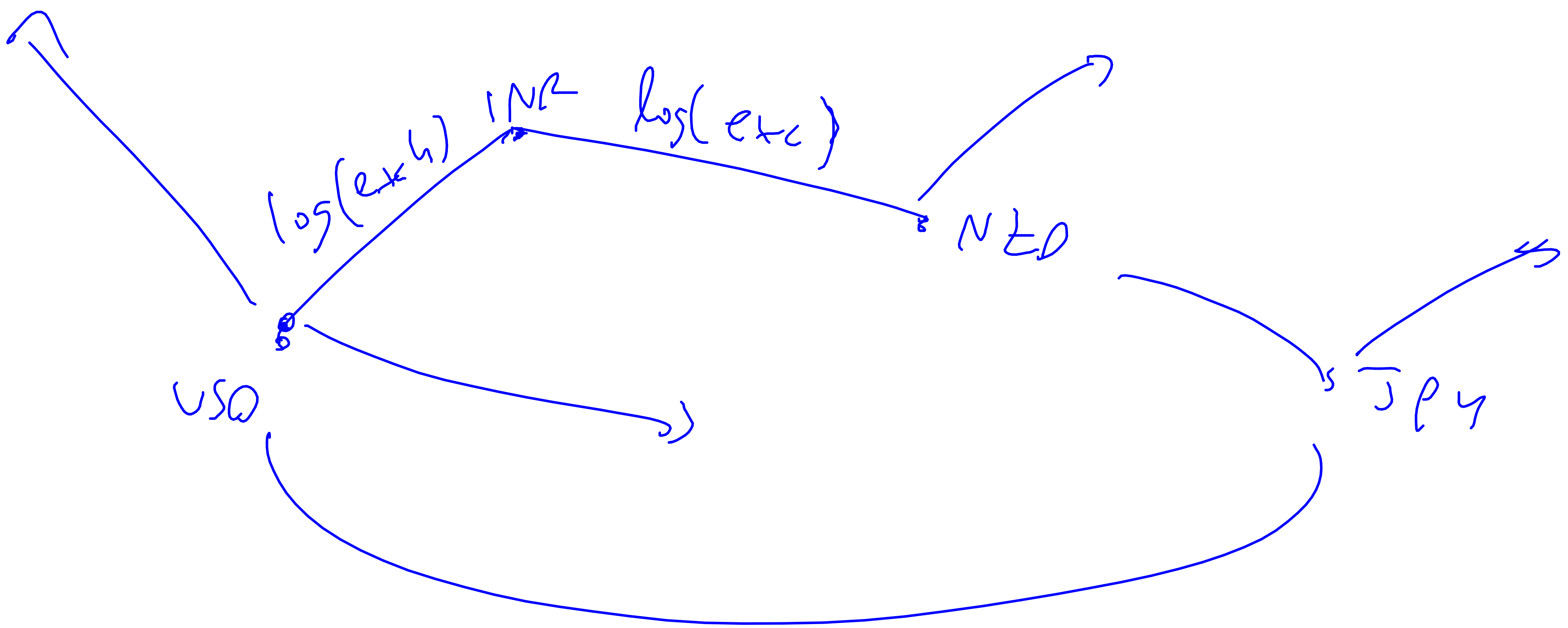
Auto-refresh 15x
0 : 56

	USD	EUR	GBP	INR	AUD	CAD	ZAR	NZD	JPY
1 USD	1.00000	0.72611	0.62261	61.3426	1.05366	1.04474	9.87360	1.21095	98.0247
Inverse:	1.00000	1.37721	1.60613	0.01630	0.94907	0.95718	0.10128	0.82580	0.01020
1 EUR	1.37721	1.00000	0.85747	84.4815	1.45111	1.43882	13.5980	1.66772	135.000
Inverse:	0.72611	1.00000	1.16622	0.01184	0.68913	0.69501	0.07354	0.59962	0.00741
1 GBP	1.60613	1.16622	1.00000	98.5241	1.69231	1.67799	15.8582	1.94494	157.440
Inverse:	0.62261	0.85747	1.00000	0.01015	0.59091	0.59595	0.06306	0.51416	0.00635
1 BMD	1.00000	0.72611	0.62261	61.3426	1.05366	1.04474	9.87360	1.21095	98.0247
Inverse:	1.00000	1.37721	1.60613	0.01630	0.94907	0.95718	0.10128	0.82580	0.01020

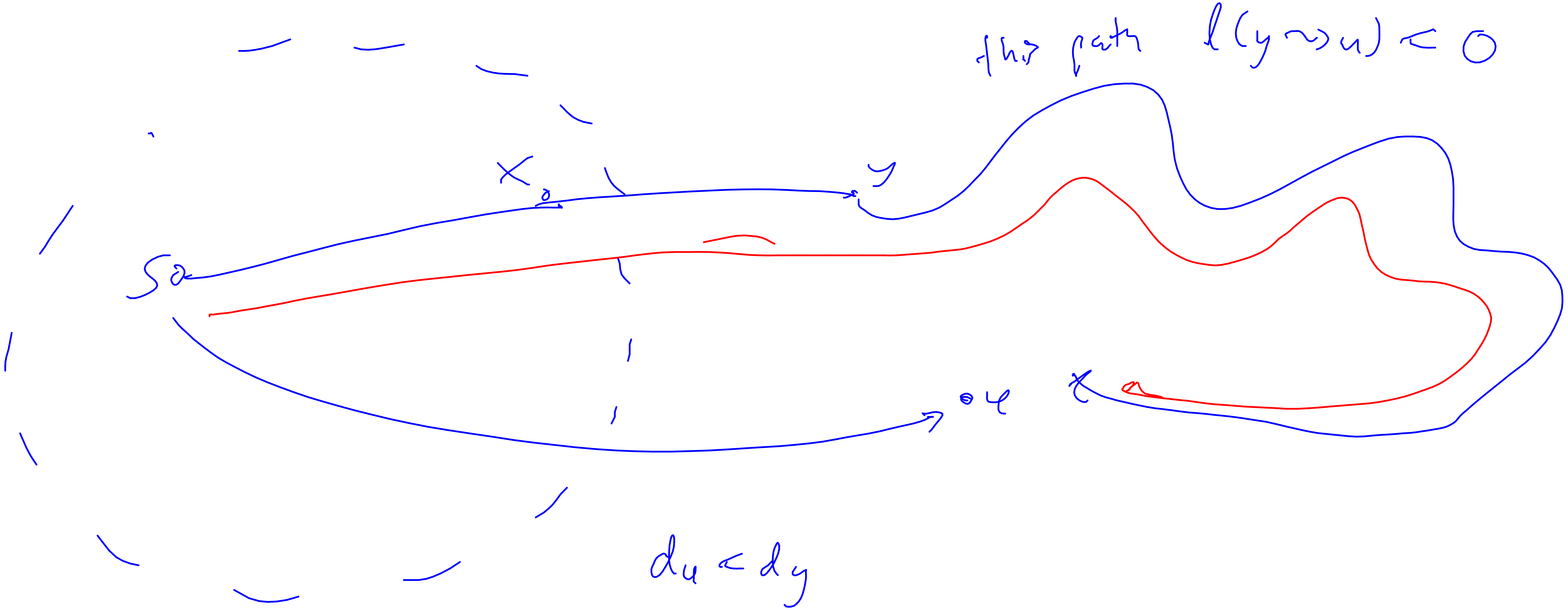
Mid-market rates: 2013-10-29 15:53 UTC

Click on a currency code to learn about it.

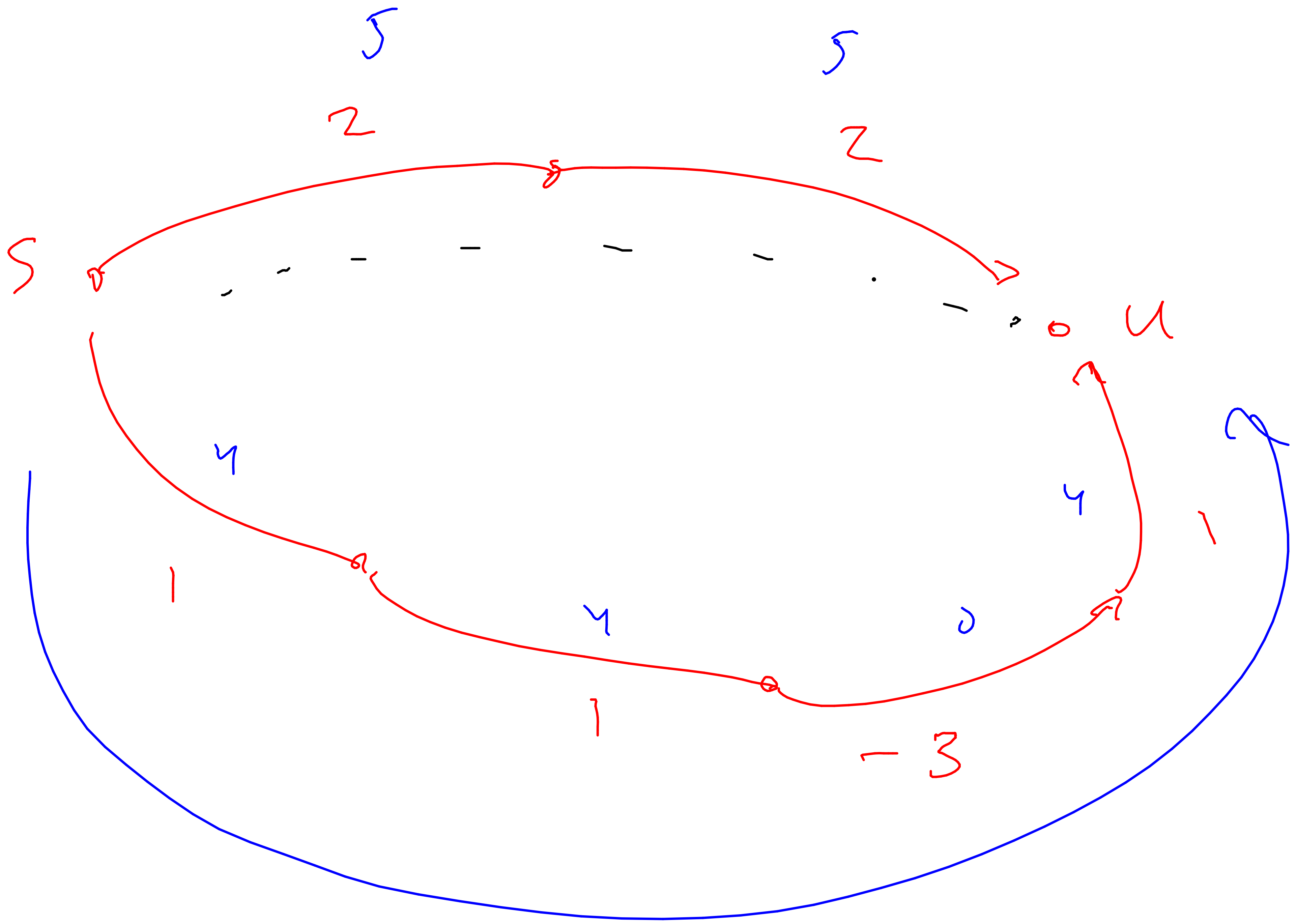
(1) USD 1\$ →(61.46) 61.46 →(0.01975) 1.213 NZD → 98.20 JPY → 0.9997 USD.



WHERE DOES OLD ARGUMENT BREAK DOWN

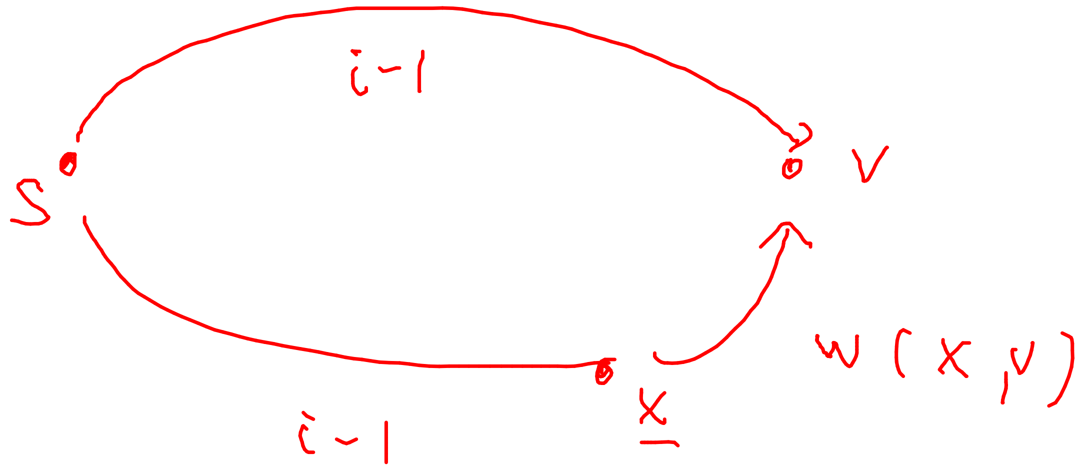


FIRST IDEAS:



SSSP(G,s)

$\text{SHORT}_{i,v}$ = length of the shortest path from $s \rightsquigarrow v$ that takes $\leq i$ hops.



$\text{Short}_{i,s} = 0$

$\text{Short}_{0,v} = \infty$

$$\text{Short}_{i,v} = \min \begin{cases} \text{SHORT}_{i-1,v} \\ \text{SHORT}_{i-1,x} + w(x,v) \quad \text{for all } x \in V. \end{cases}$$

SSSP(G,s)

$$\textsc{short}_{i,v} = \begin{cases} \infty & i = 0 \\ 0 & v = s \\ \min_{x \in V} & \left\{ \begin{array}{l} \textsc{short}_{i-1,v} \\ \textsc{short}_{i-1,x} + w(x,v) \end{array} \right\} \end{cases}$$

MAX LEN OF A SIMPLE PATH:

V-1

BELLMAN-FORD(G,s)

Short$_{V-1, v}$

BELLMAN-FORD(G, s)

$\text{SHORT}_{0,s} \leftarrow 0$

$\forall v \in V - \{s\}$, $\text{SHORT}_{0,v} \leftarrow \infty$

for $i = 1, \ldots, V - 1$

  do for each $v \in V - \{s\}$

    do $\text{SHORT}_{i,v} = \min_{x \in Adj(v)} \left\{ \begin{array}{l} \text{SHORT}_{i-1,v} \\ w(x,v) + \text{SHORT}_{i-1,x} \end{array} \right\}$

→ iterate over all the edges

BELLMAN-FORD(G, s)

$\text{SHORT}_{0,s} \leftarrow 0$

$\forall v \in V - \{s\}$, $\text{SHORT}_{0,v} \leftarrow \infty$

for $i = 1, \ldots, V - 1$

  do for each $e = (x, y) \in E$

    do $\text{SHORT}_{i,y} = \min \begin{cases} \text{SHORT}_{i-1,y} \\ \text{SHORT}_{i,y} \\ w(x,y) + \text{SHORT}_{i-1,x} \end{cases}$

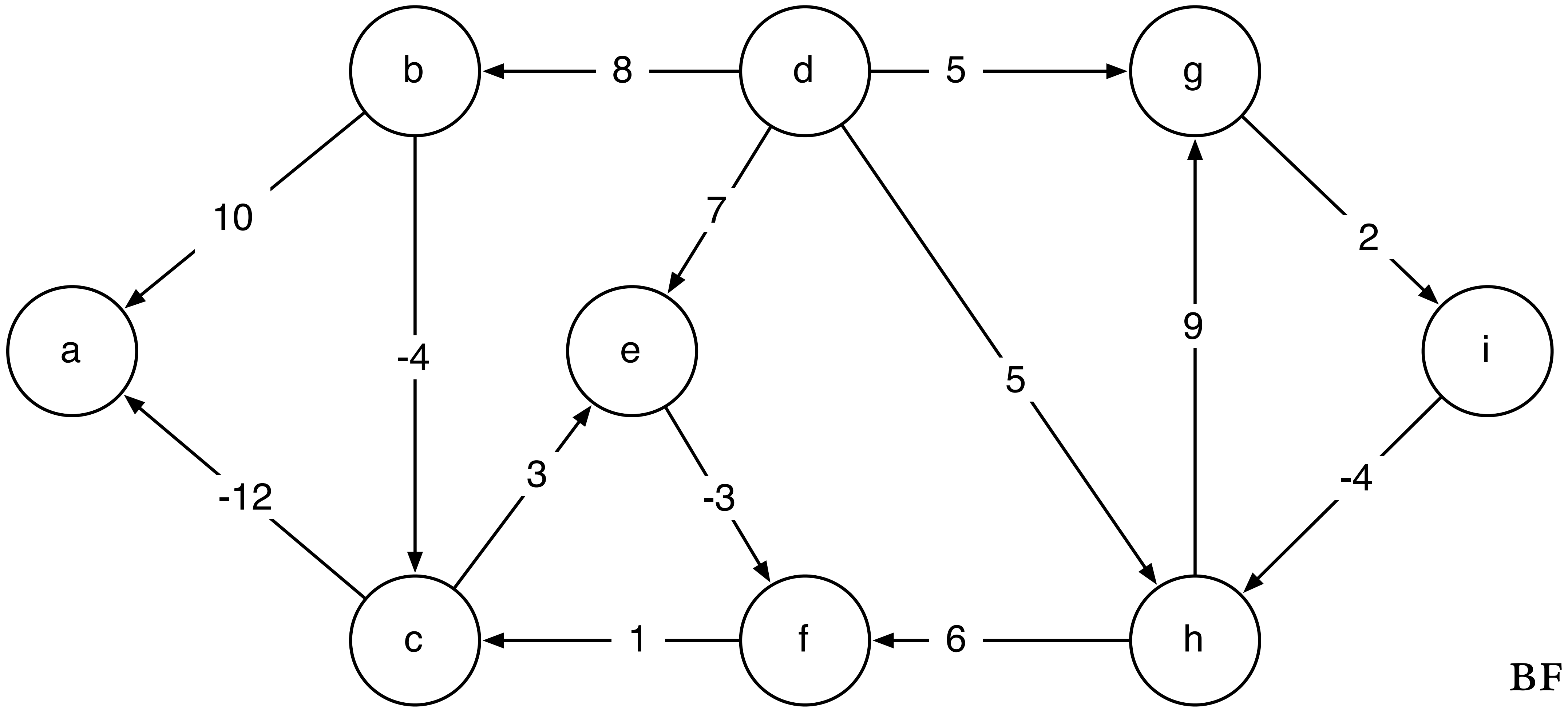
$\rightarrow \Theta(VE)$

$\downarrow$

$\Theta(E \log V)$

$\downarrow$

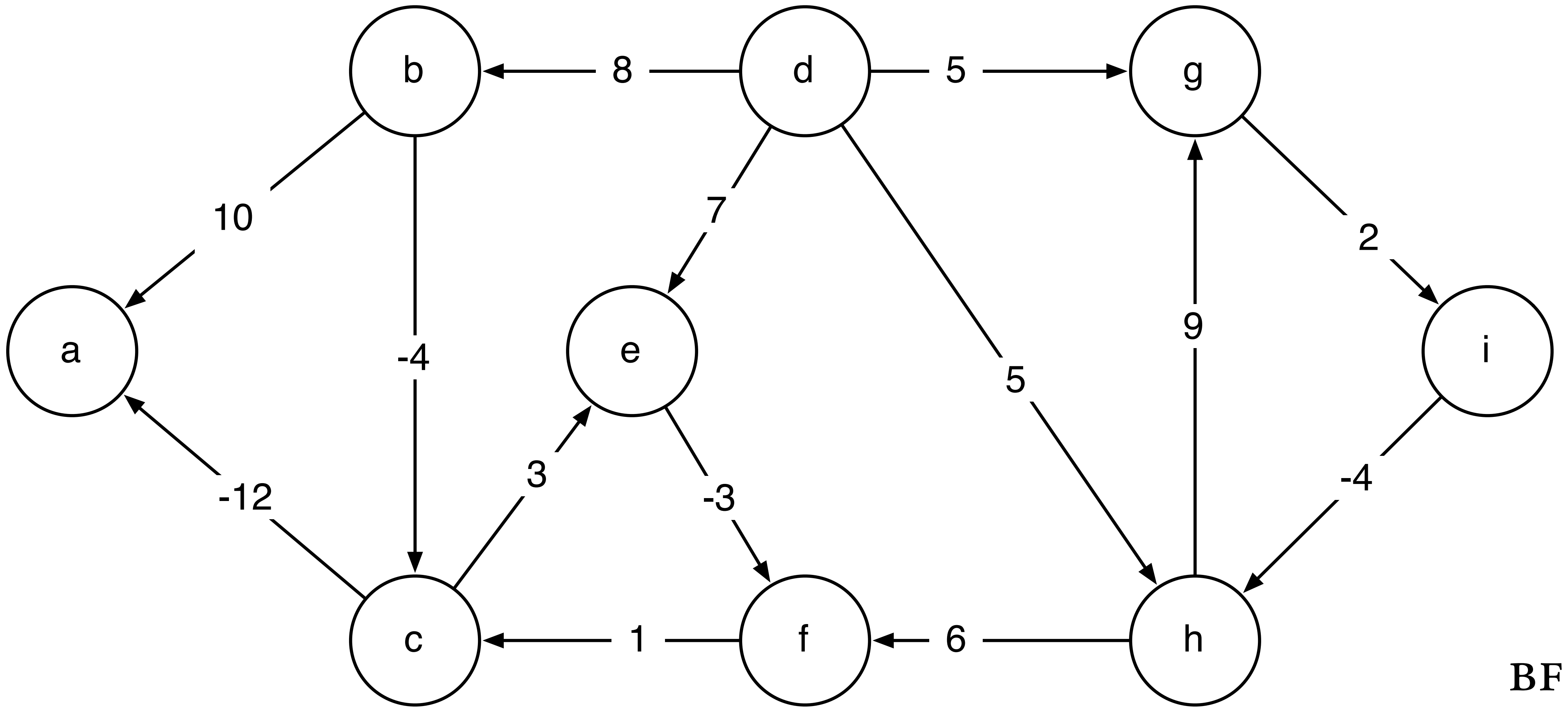
$\Theta(E)$



$$\text{SHORT}_{i,v} = \begin{cases} \infty & i = 0 \\ 0 & v = s \\ \min_{x \in V} & \left\{ \begin{array}{l} \text{SHORT}_{i-1,v} \\ \text{SHORT}_{i-1,x} + w(x,v) \end{array} \right\} \end{cases}$$

BF(G,d)

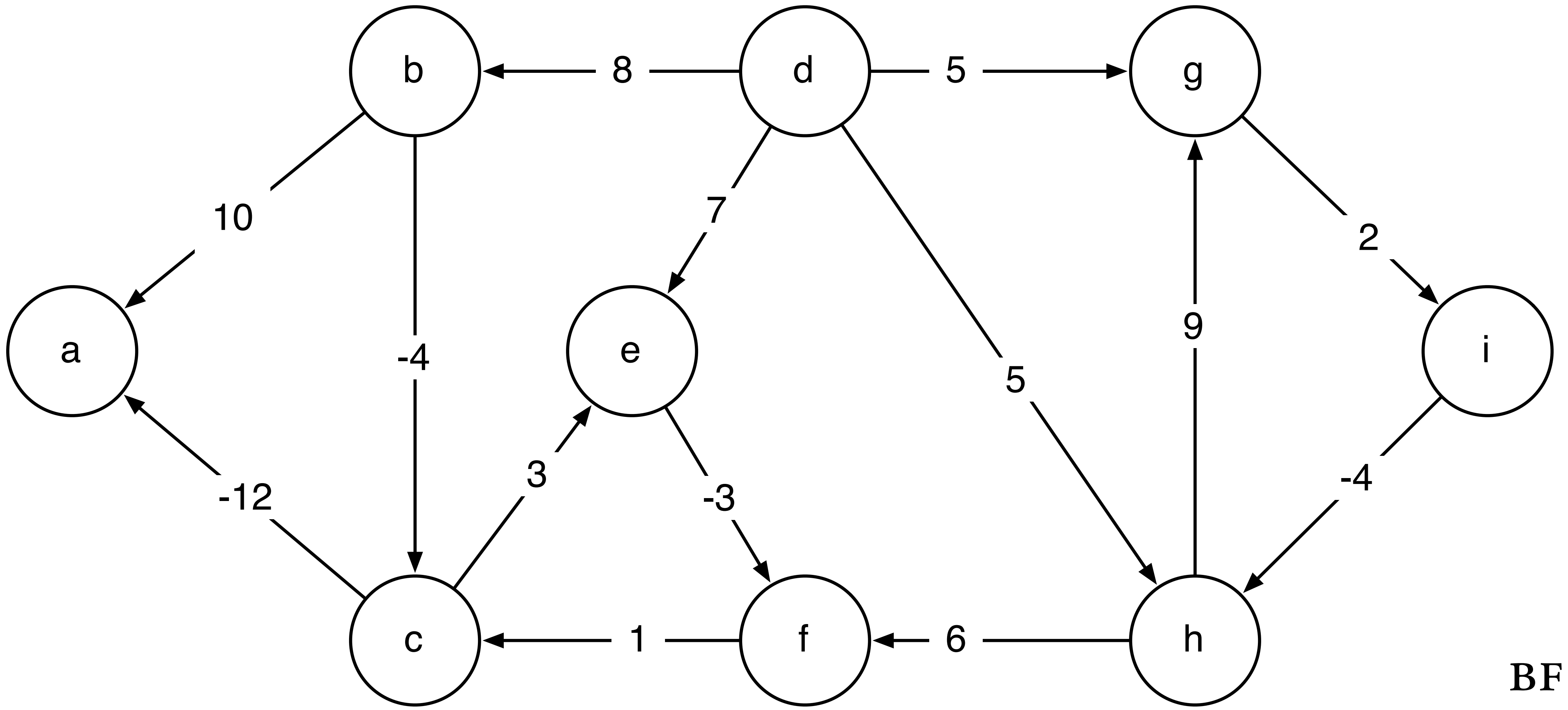
	0	1	2	3	4	5	6	7
A								
B								
C								
D	0							
E								
F								
G								
H								
I								



$$\text{SHORT}_{i,v} = \begin{cases} \infty & i = 0 \\ 0 & v = s \\ \min_{x \in V} & \left\{ \begin{array}{l} \text{SHORT}_{i-1,v} \\ \text{SHORT}_{i-1,x} + w(x,v) \end{array} \right\} \end{cases}$$

BF(G,d)

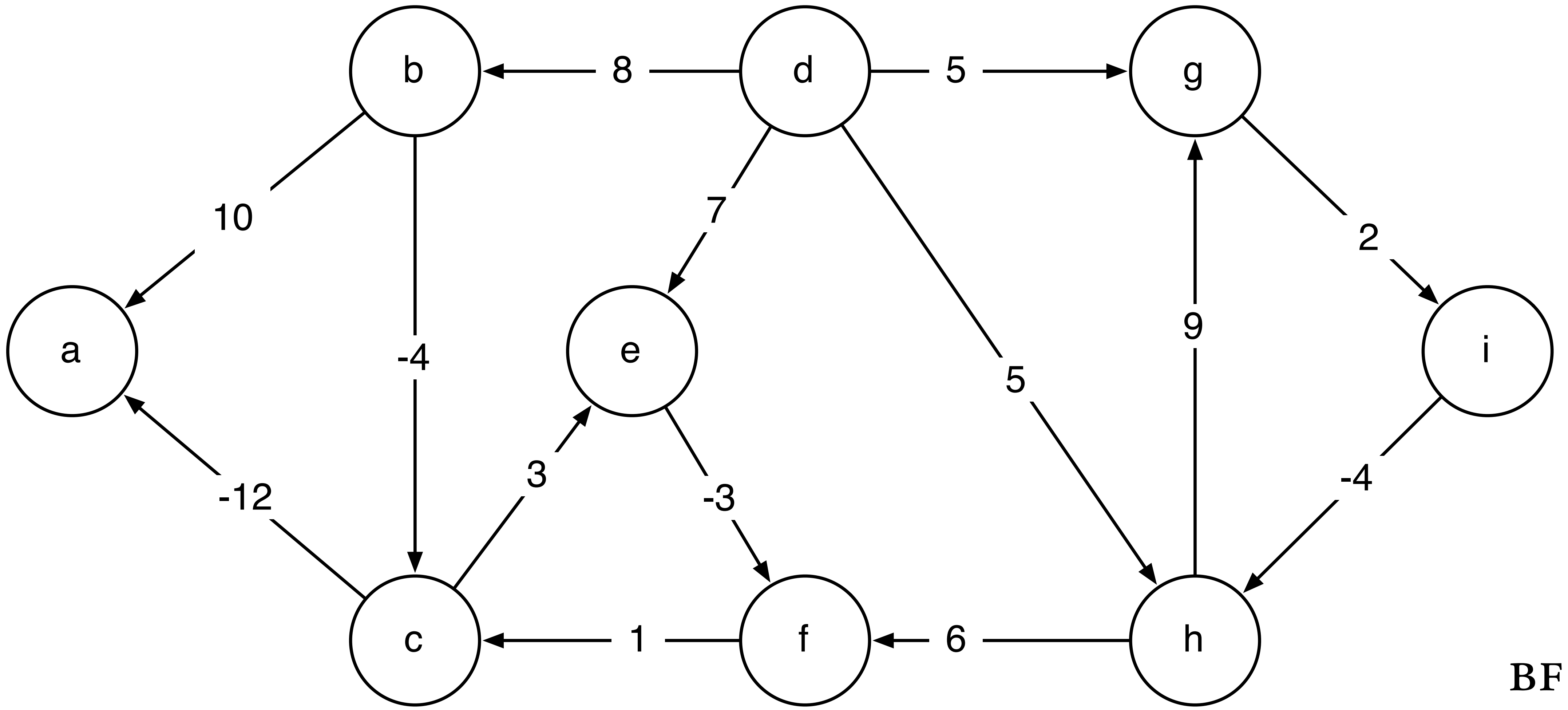
	0	1	2	3	4	5	6	7
A	∞							
B	∞							
C	∞							
D	0							
E	∞							
F	∞							
G	∞							
H	∞							
I	∞							



$$\text{SHORT}_{i,v} = \begin{cases} \infty & i = 0 \\ 0 & v = s \\ \min_{x \in V} & \left\{ \begin{array}{l} \text{SHORT}_{i-1,v} \\ \text{SHORT}_{i-1,x} + w(x,v) \end{array} \right\} \end{cases}$$

BF(G,d)

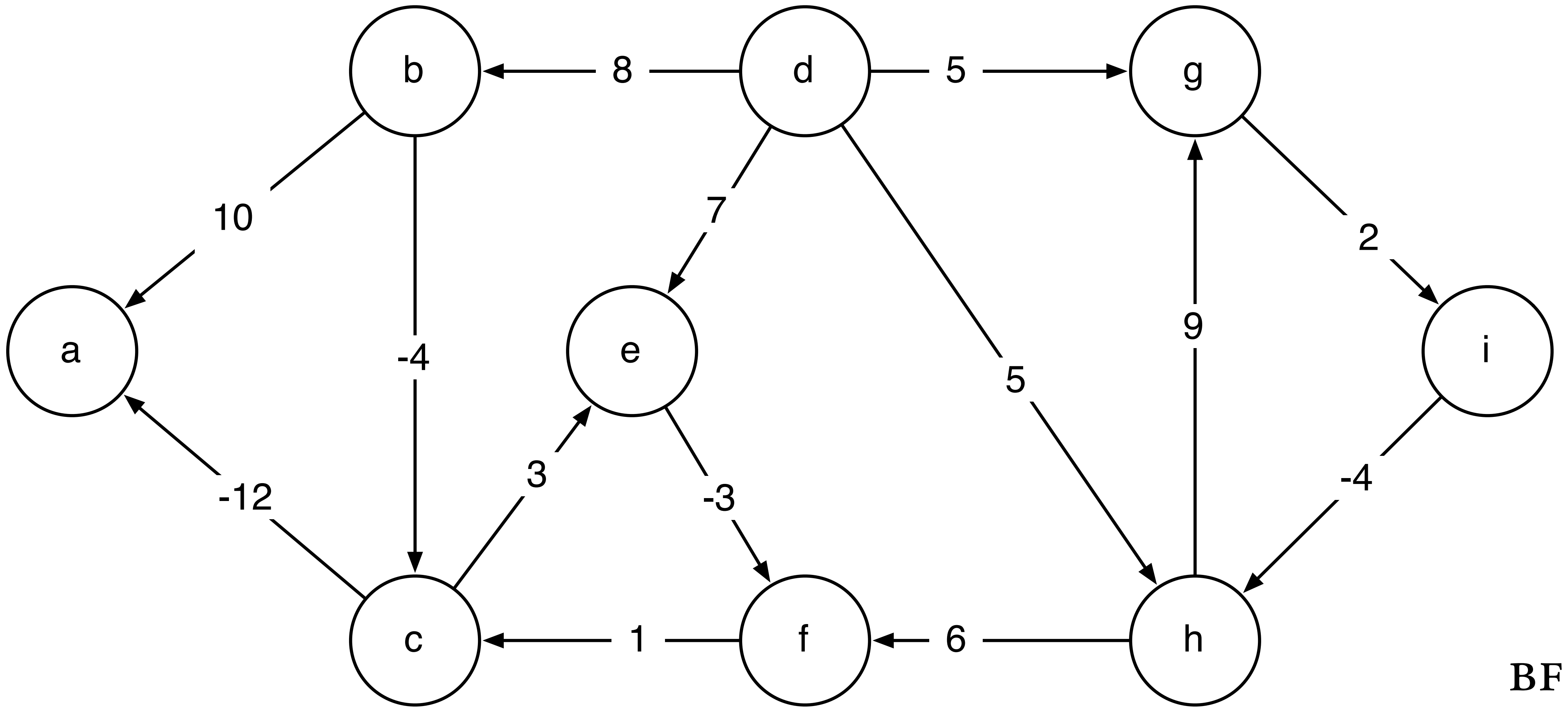
	0	1	2	3	4	5	6	7
A	∞							
B	∞	8						
C	∞							
D	0	0						
E	∞	7						
F	∞							
G	∞	5						
H	∞	5						
I	∞							



$$\text{SHORT}_{i,v} = \begin{cases} \infty & i = 0 \\ 0 & v = s \\ \min_{x \in V} & \left\{ \begin{array}{l} \text{SHORT}_{i-1,v} \\ \text{SHORT}_{i-1,x} + w(x,v) \end{array} \right\} \end{cases}$$

BF(G,d)

	0	1	2	3	4	5	6	7
A	∞		18					
B	∞	8	8					
C	∞		4					
D	0	0	0					
E	∞	7	7					
F	∞		4					
G	∞	5	5					
H	∞	5	5					
I	∞		7					



$$
\text{SHORT}_{i,v} = \begin{cases} \infty & i = 0 \\ 0 & v = s \\ \min_{x \in V} & \left\{ \begin{array}{l} \text{SHORT}_{i-1,v} \\ \text{SHORT}_{i-1,x} + w(x,v) \end{array} \right\} \end{cases}
$$

BF(G,d)

	0	1	2	3	4	5	6	7
A	∞		18	-8				
B	∞	8	8	8				
C	∞		4	4				
D	0	0	0	0				
E	∞	7	7	7				
F	∞		4	4				
G	∞	5	5	5				
H	∞	5	5	3				
I	∞		7	7				

OPTIMIZATION

BELLMAN-FORD(G, s)

$\text{SHORT}_{0,s} \leftarrow 0$

$\forall v \in V - \{s\}$, $\text{SHORT}_{0,v} \leftarrow \infty$

for $i = 1, \ldots, V - 1$

  do for each $e = (x, y) \in E$

    do $\text{SHORT}_{i,y} = \min \left\{ \begin{array}{l} \text{SHORT}_{i-1,y} \\ \text{SHORT}_{i,y} \\ w(x,y) + \text{SHORT}_{i-1,x} \end{array} \right\}$

BELLMAN-FORD(G, s)

$d_s \leftarrow 0$

$\forall v \in V - \{s\}$, $d_v \leftarrow \infty$

for $i = 1, \ldots, V - 1$

  do for each $e = (x, y) \in E$

    do $d_y \leftarrow \min \{ \; d_y, w(x,y) + d_x \; \}$

RUNNING TIME

BELLMAN-FORD(G, s)

$d_s \leftarrow 0$

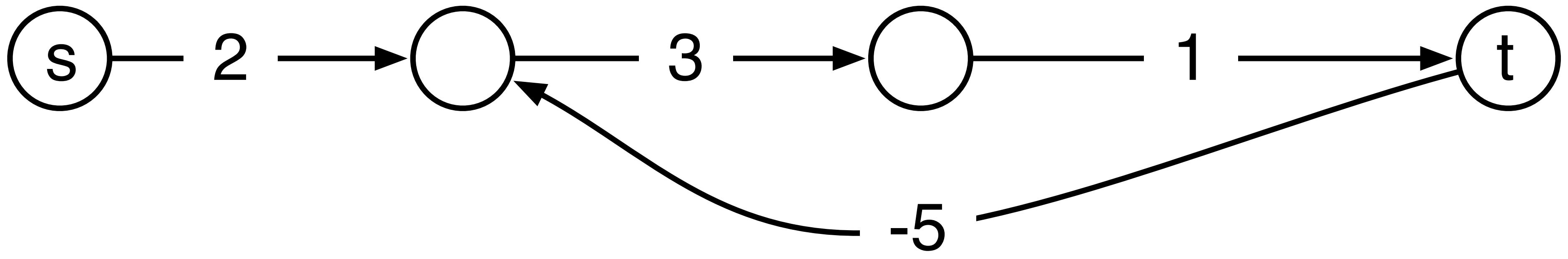
$\forall v \in V - \{s\},\ d_v \leftarrow \infty$

for $i = 1, \ldots, V - 1$

  do for each $e = (x, y) \in E$

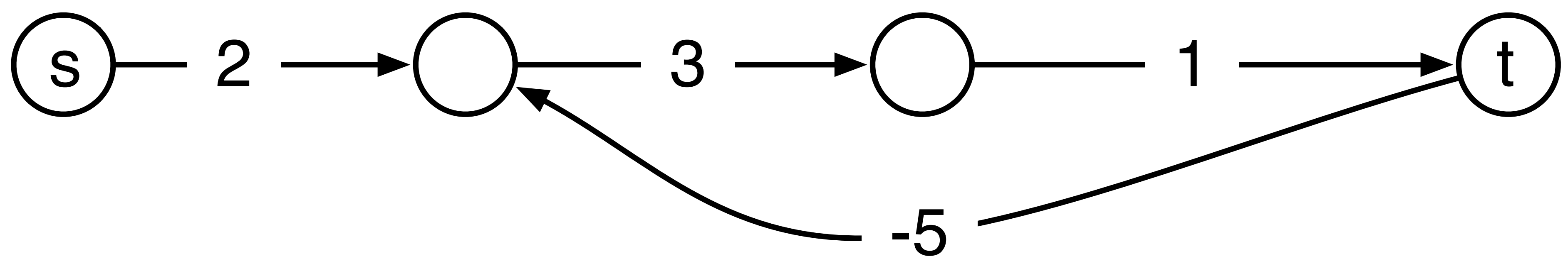
    do $d_y \leftarrow \min\left\{\ d_y, w(x, y) + d_x\ \right\}$

NEGATIVE CYCLES?



S	0			
A				
B				
T				

NEGATIVE CYCLES?



S	0	0	0	0
A	2	2	2	1
B		5	5	5
T			6	6

APPLICATIONS OF BF

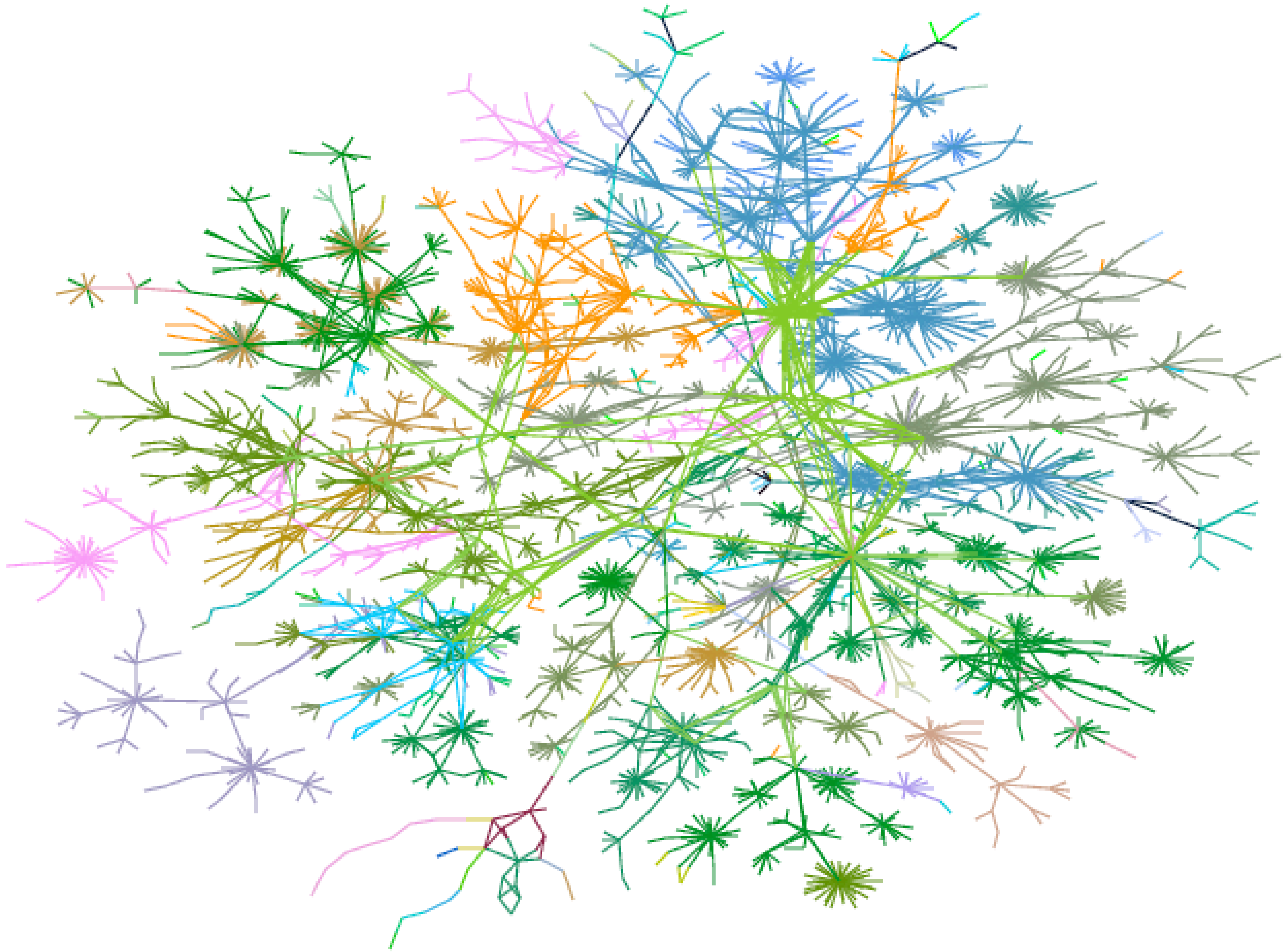
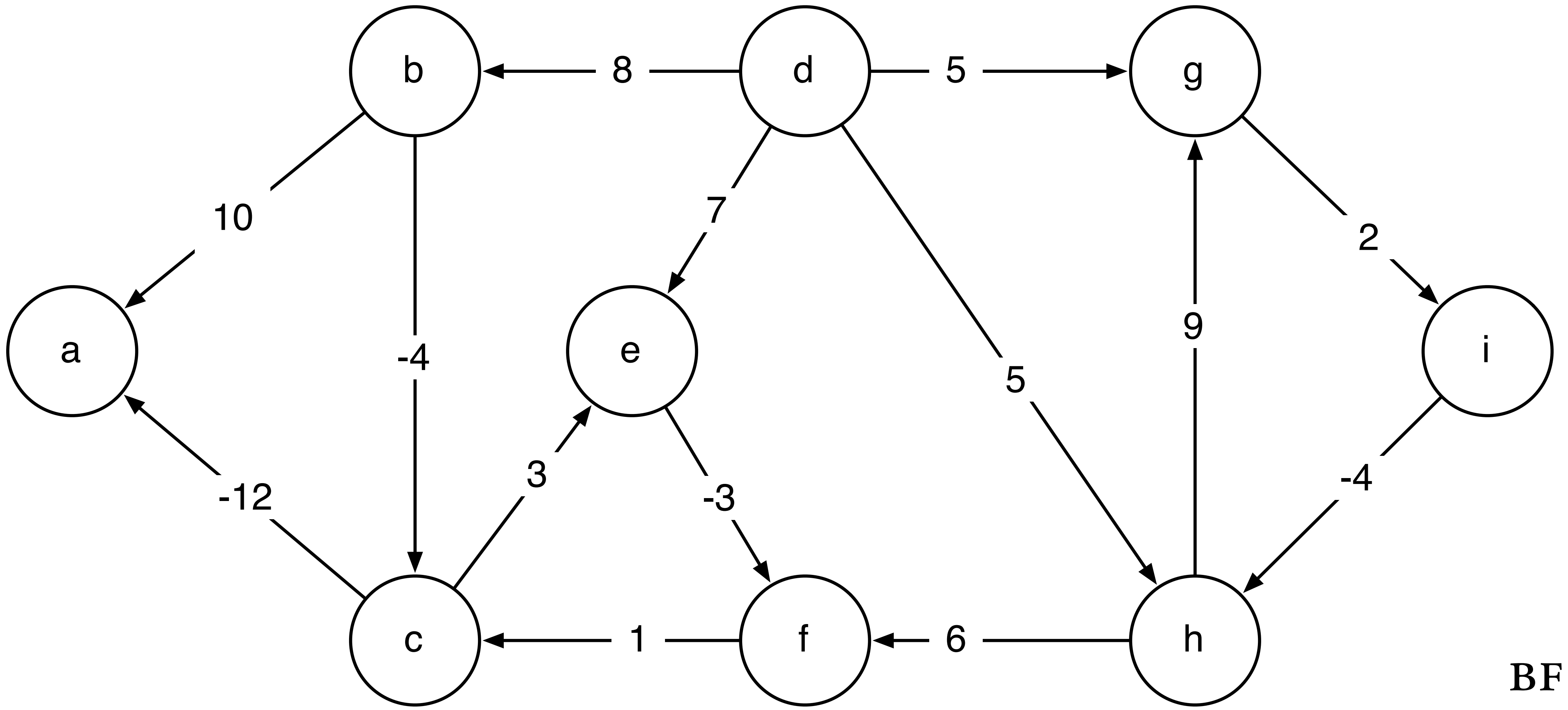


image: cheswick et al
Figure 3: Lucent's intranet as of 1 October 1999.



BF(G,d)

	0	1	2	3	4	5	6	7
A	∞							
B	∞	8						
C	∞							
D	0	0						
E	∞	7						
F	∞							
G	∞	5						
H	∞	5						
I	∞							

WHAT HAPPENS WHEN

B CHANGES...

DISTANCE VECTOR

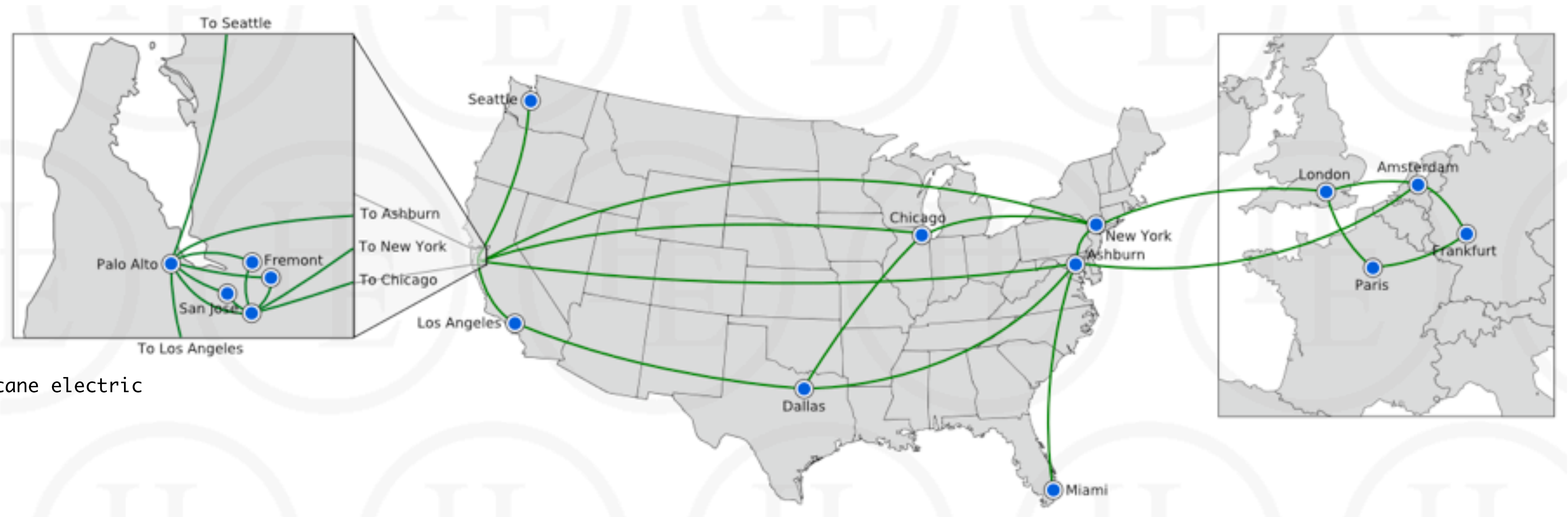
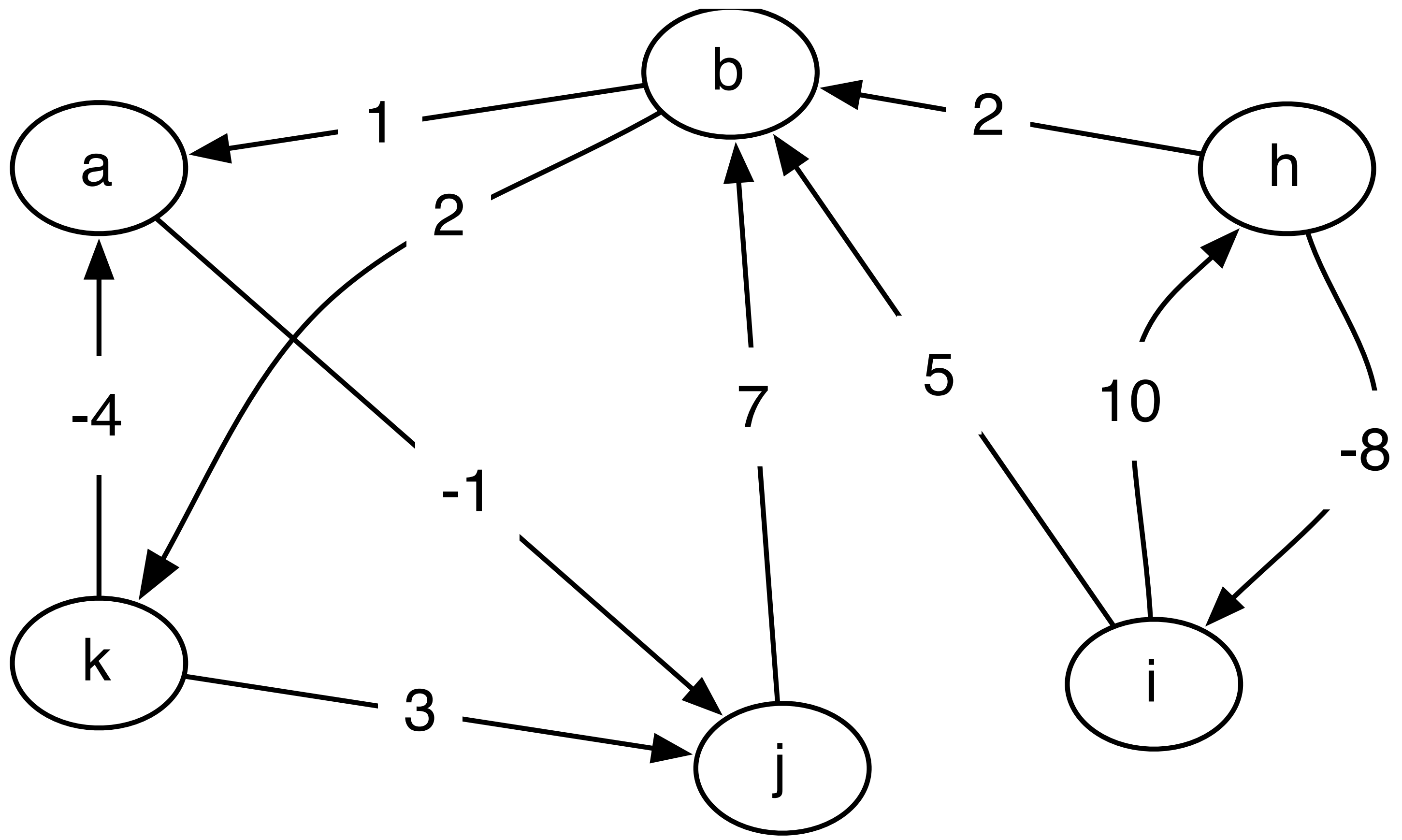


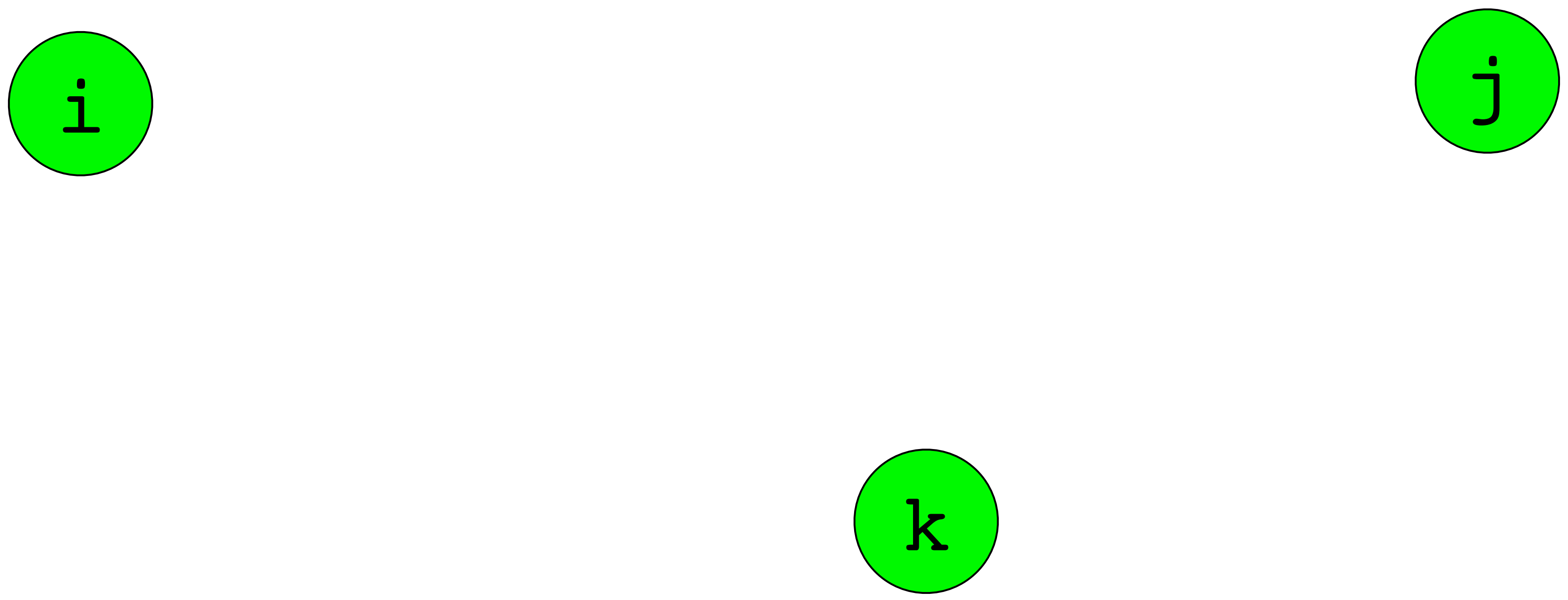
image: hurricane electric

ALL-PAIRS SHORTEST PATH



$$\mathrm{ASHORT}_{\mathtt{i,j,k}} =$$

$\mathrm{ASHORT}_{\mathtt{i},\mathtt{j},\mathtt{k}} =$



$$\mathrm{ASHORT}_{i,j,k} =$$

$$\mathrm{ASHORT}_{i,j,k} = \left\{ \begin{array}{ll} w_{i,j} & k = 0 \\ \min \left\{ \begin{array}{l} \mathrm{ASHORT}_{i,j,k-1} \\ \mathrm{ASHORT}_{i,k,k-1} + \mathrm{ASHORT}_{k,j,k-1} \end{array} \right. & k \geq 1 \end{array} \right\}$$

FLOYD-WARSHALL(G, W)