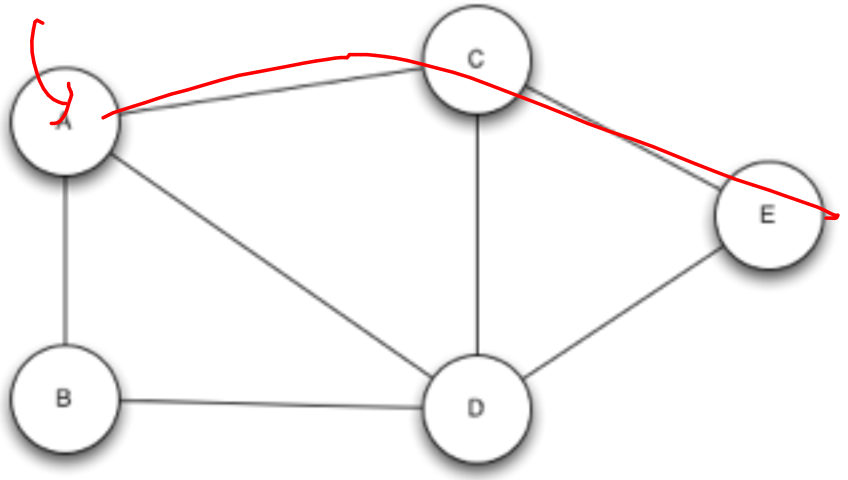
L18

4102

3.22.2016

abhi shelat

simple graph questions

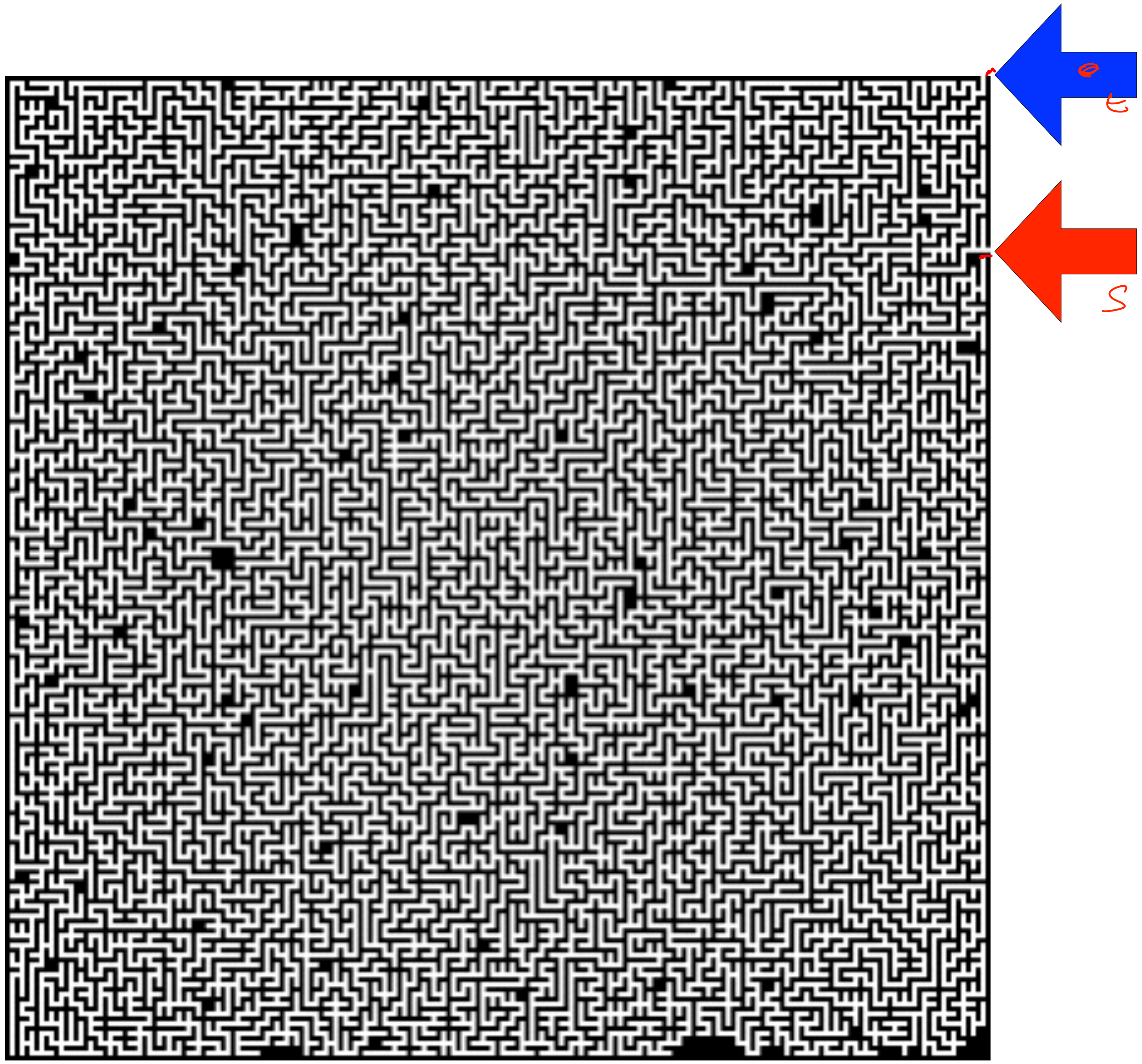


$G=(V,E)$, $w(E) \to \mathbb{N}$

all weights are ~~non-negative~~ positive for now.

what is the length of the path from a to e?

shortest


E
S

shortest path property

definition:

$\underline{\delta(s, v)}$

$\delta(s,v)$ = length of the shortest path from s to v in G.

shortest paths

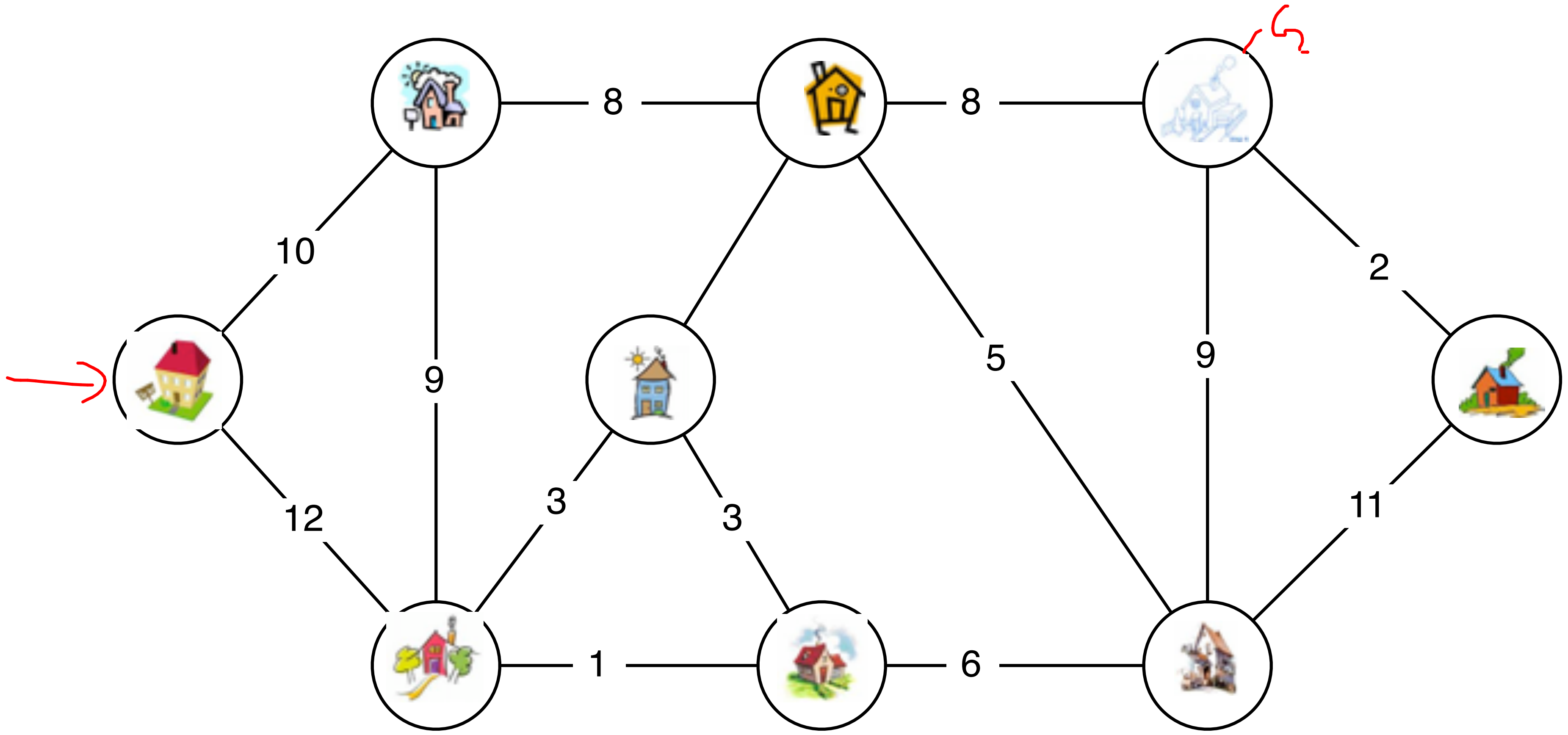
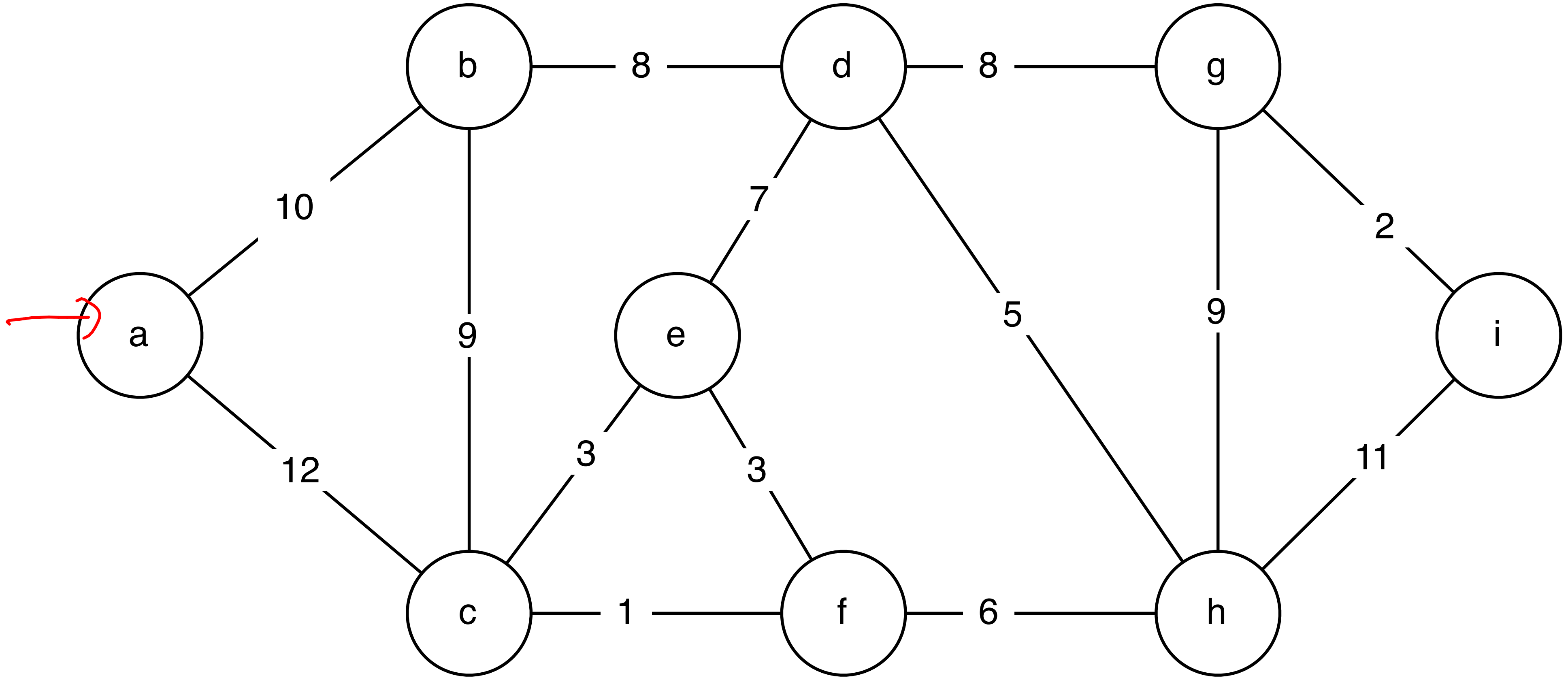
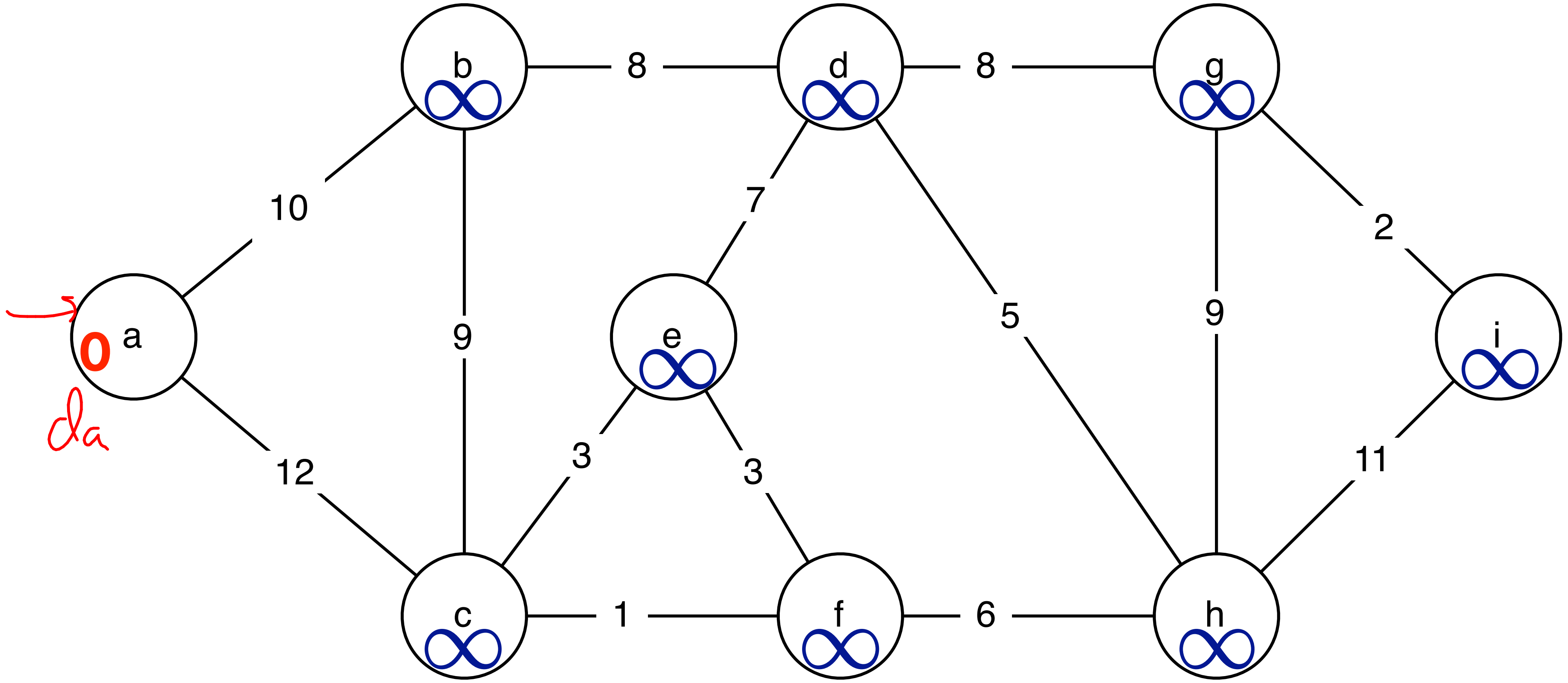


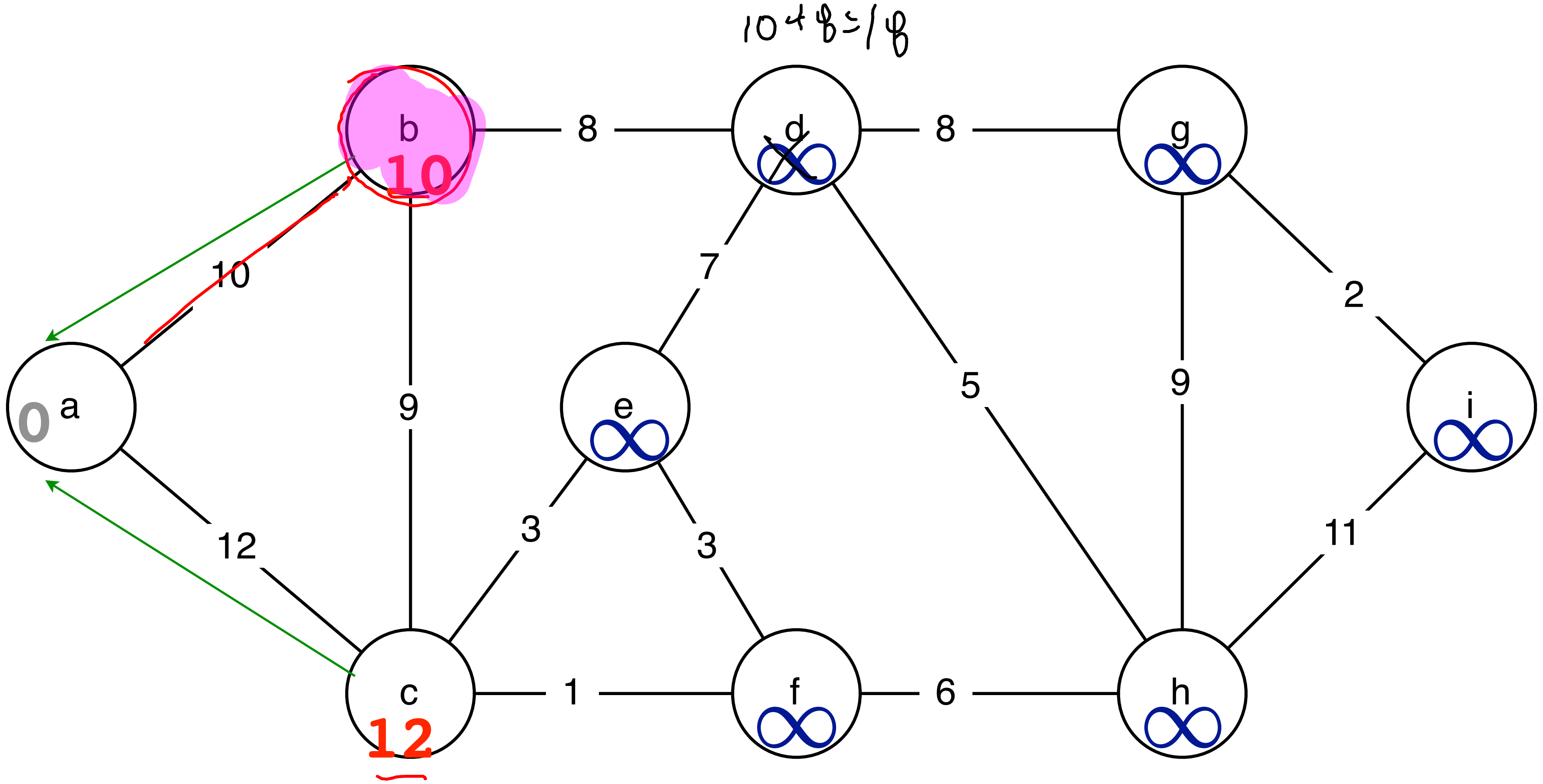
image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

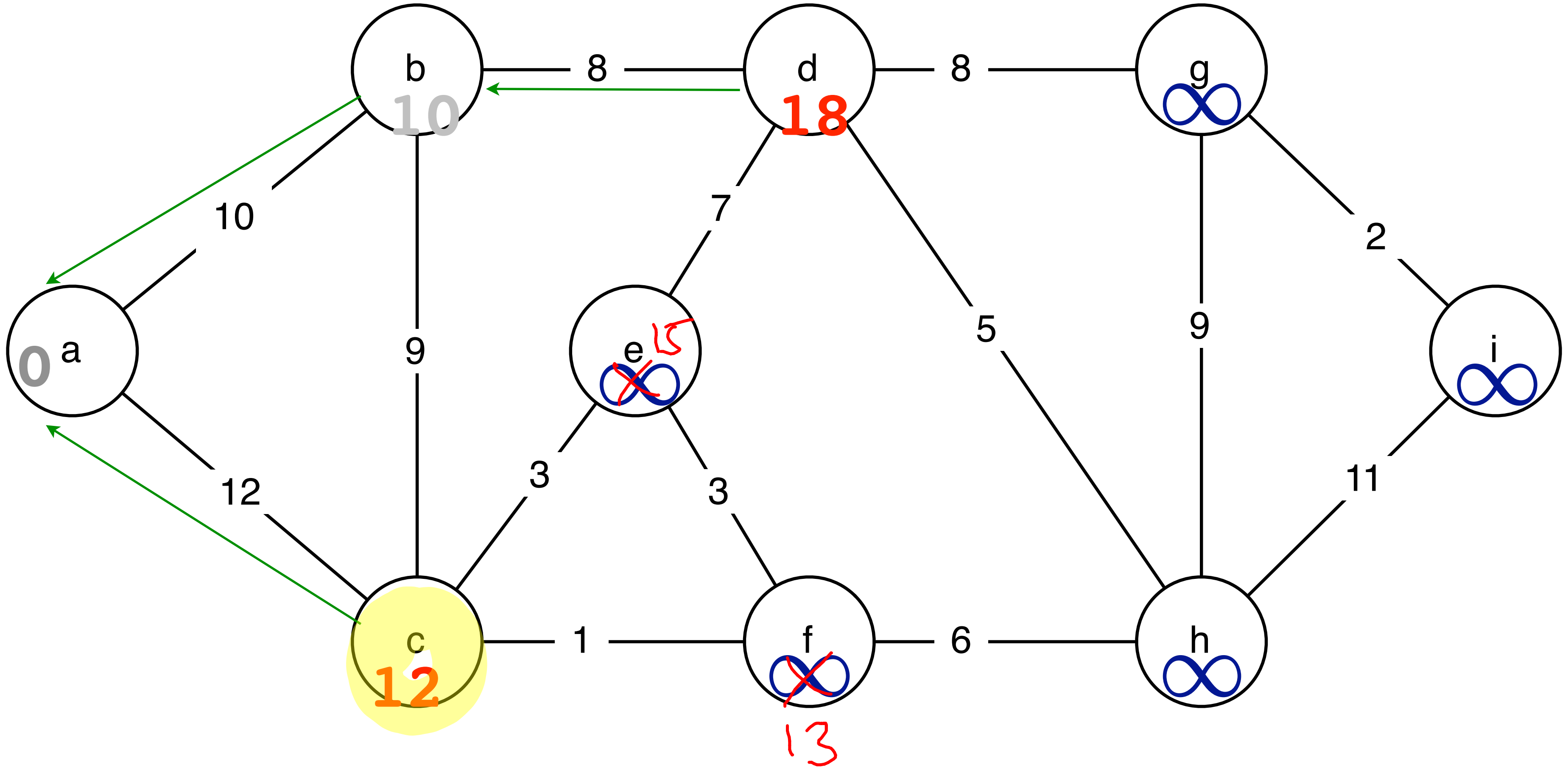
shortest paths

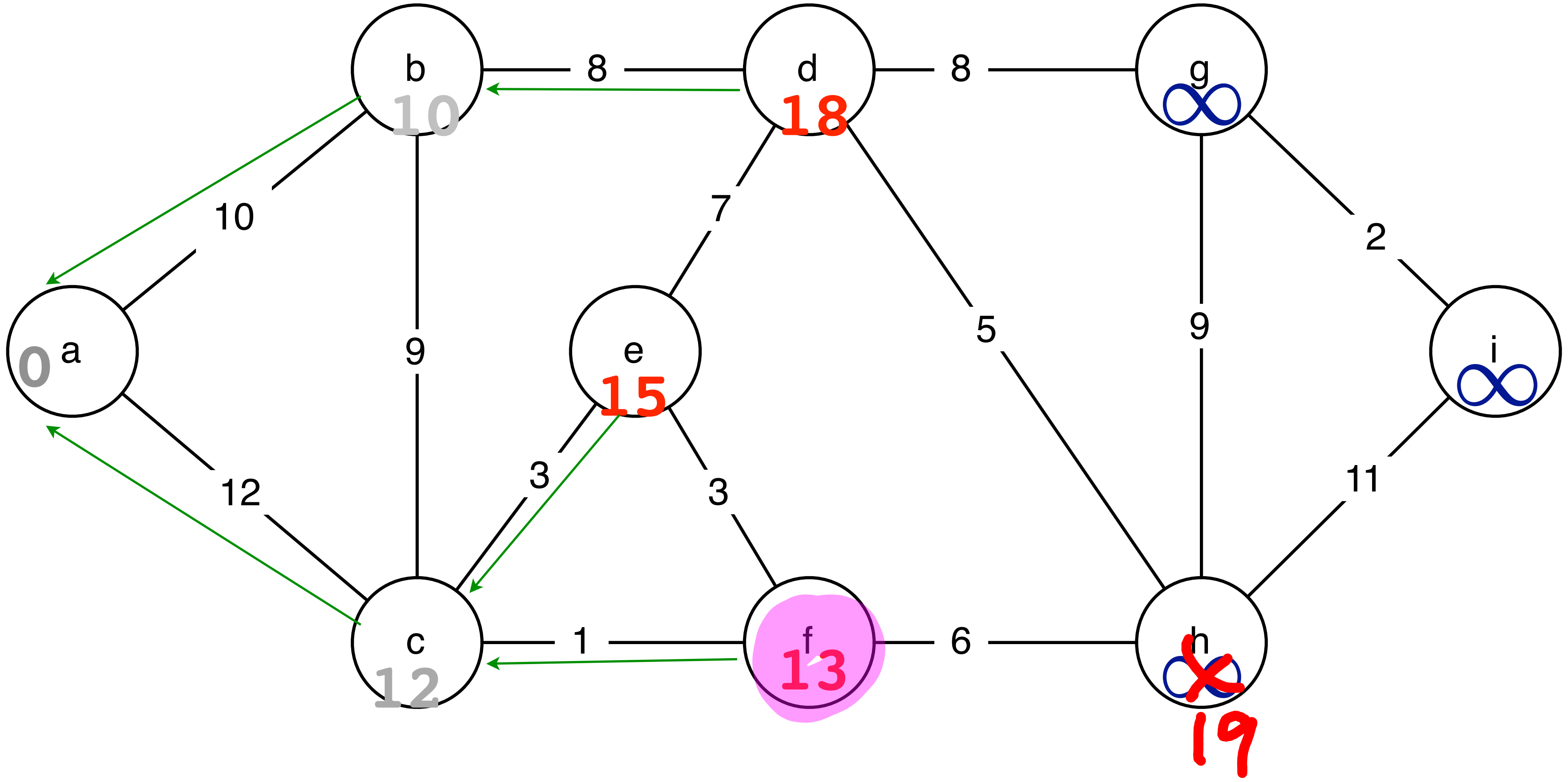
(a, g)

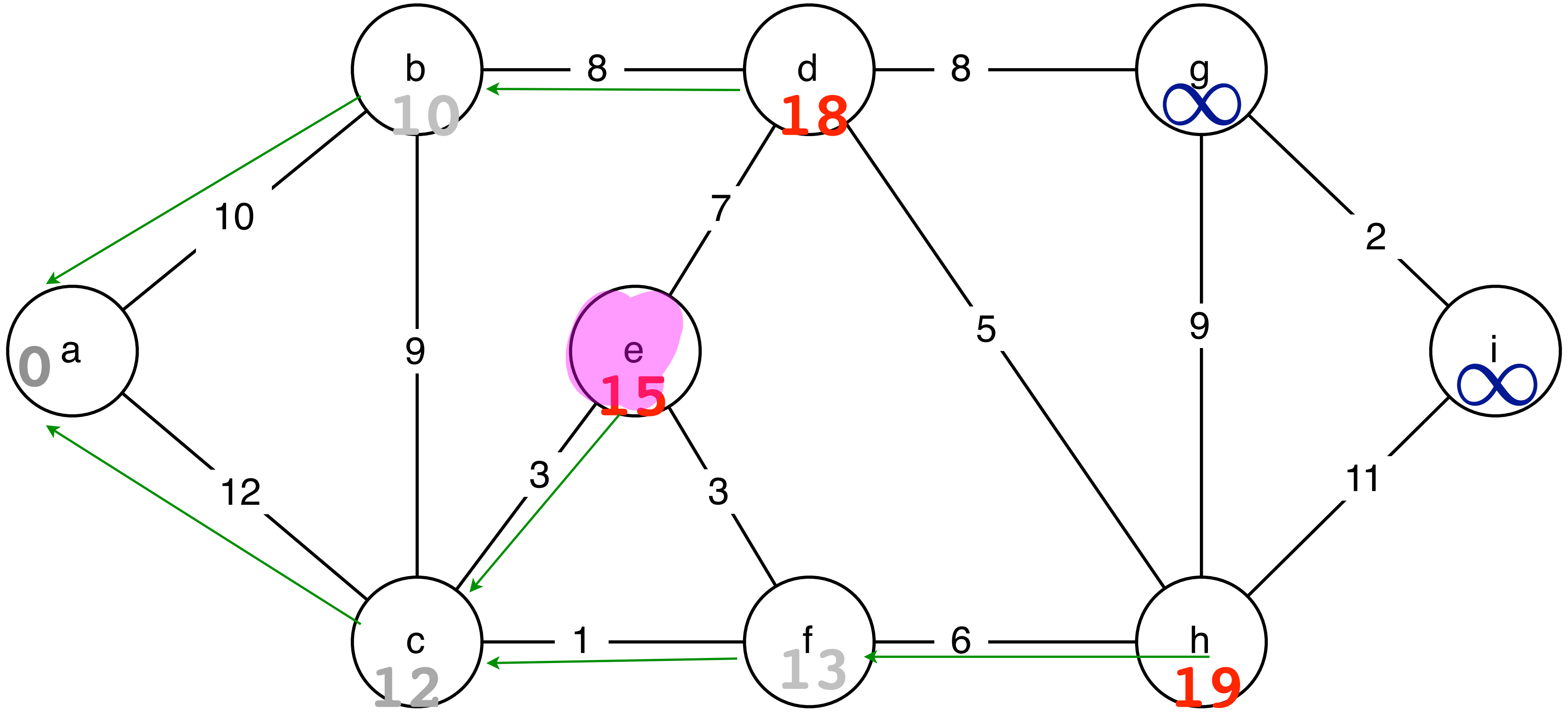


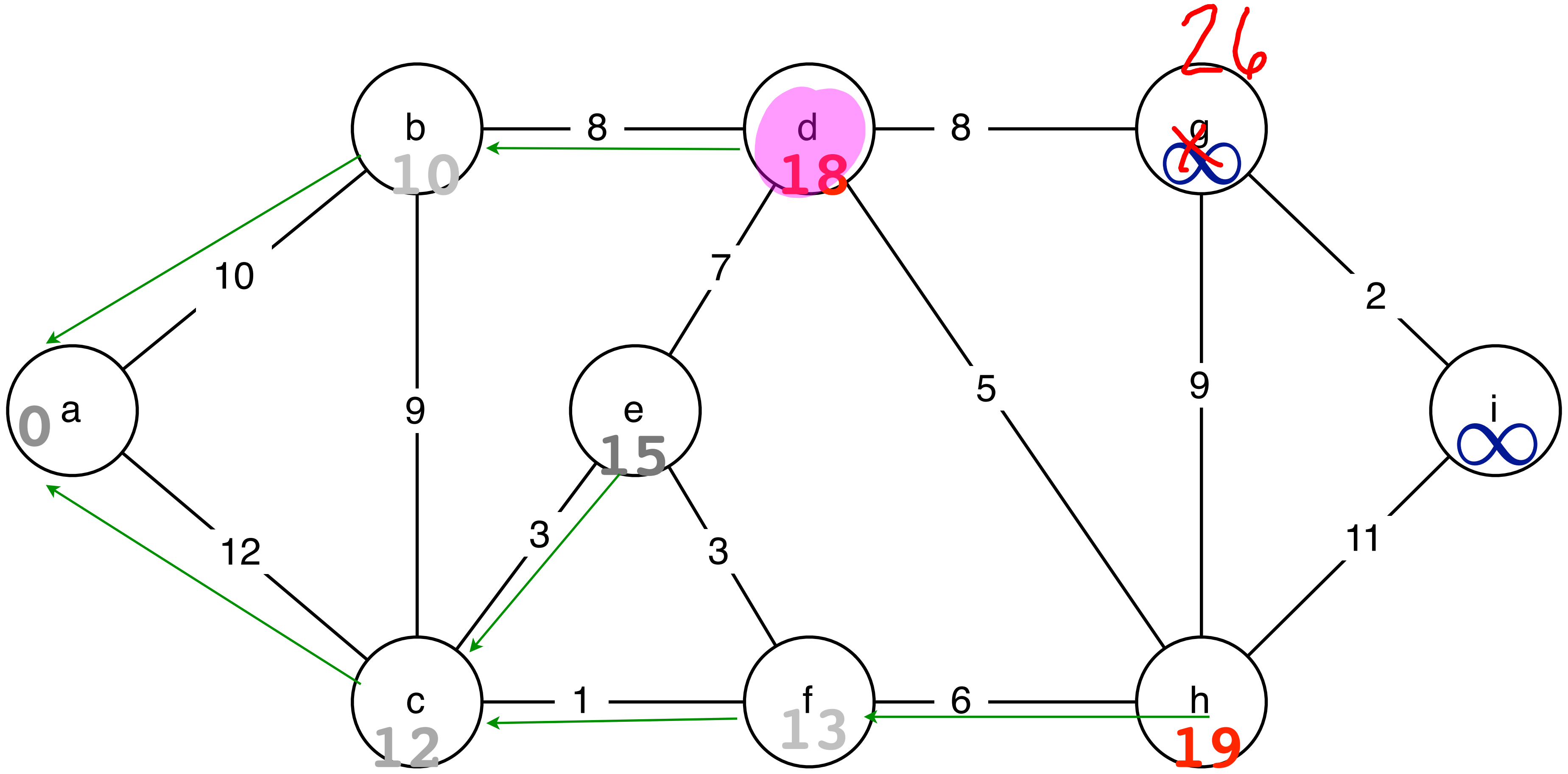

a
0
da
b
c
d
e
f
g
h
i
∞
10
12
8
9
7
3
3
1
5
6
8
9
2
11

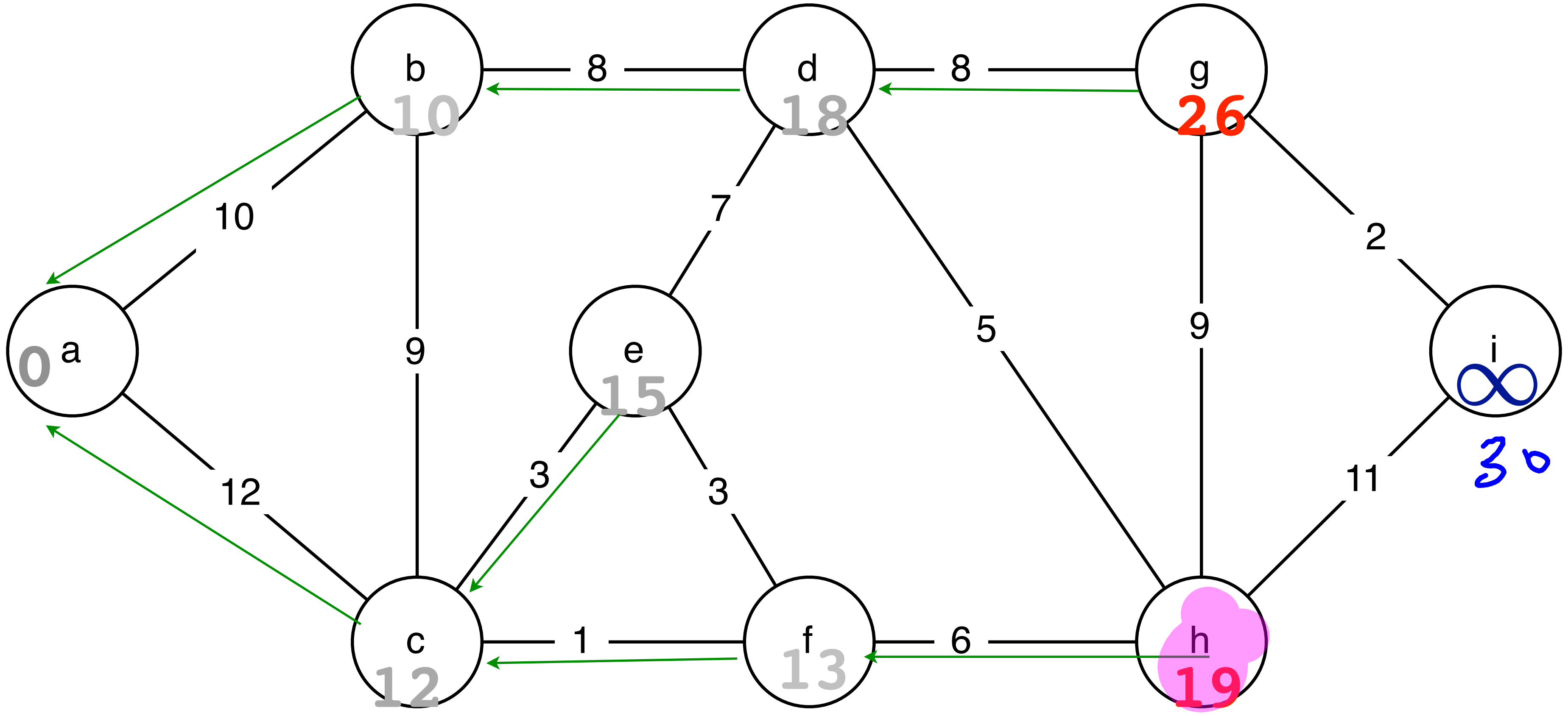

10+8=18
0 a
b 10
c 12
d ∞
e ∞
f ∞
g ∞
h ∞
i ∞
10
12
9
8
8
7
5
9
2
3
3
1
6
11

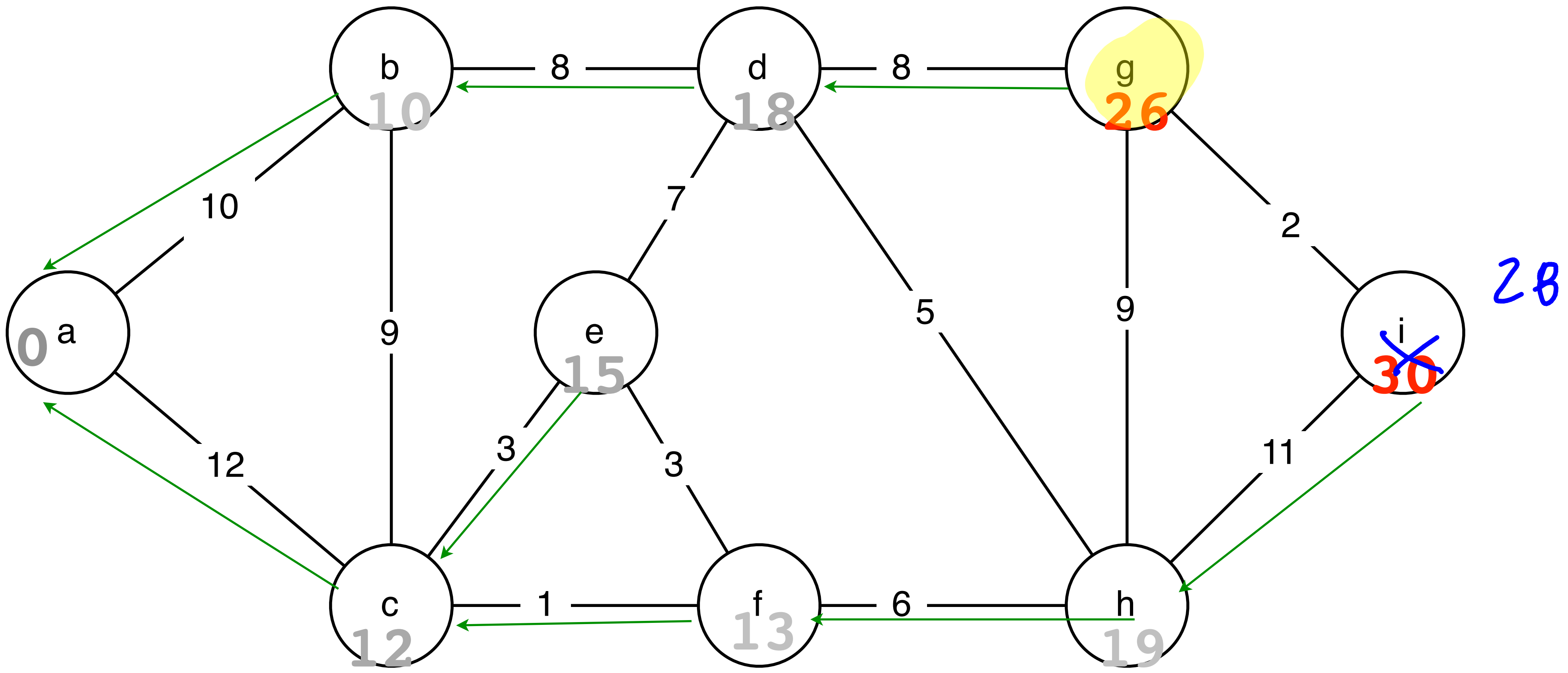

b
10
d
18
g
∞
a
0
e
∞
15
i
∞
c
12
f
∞
13
h
∞
8
8
10
9
7
5
9
2
3
3
12
1
6
11


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

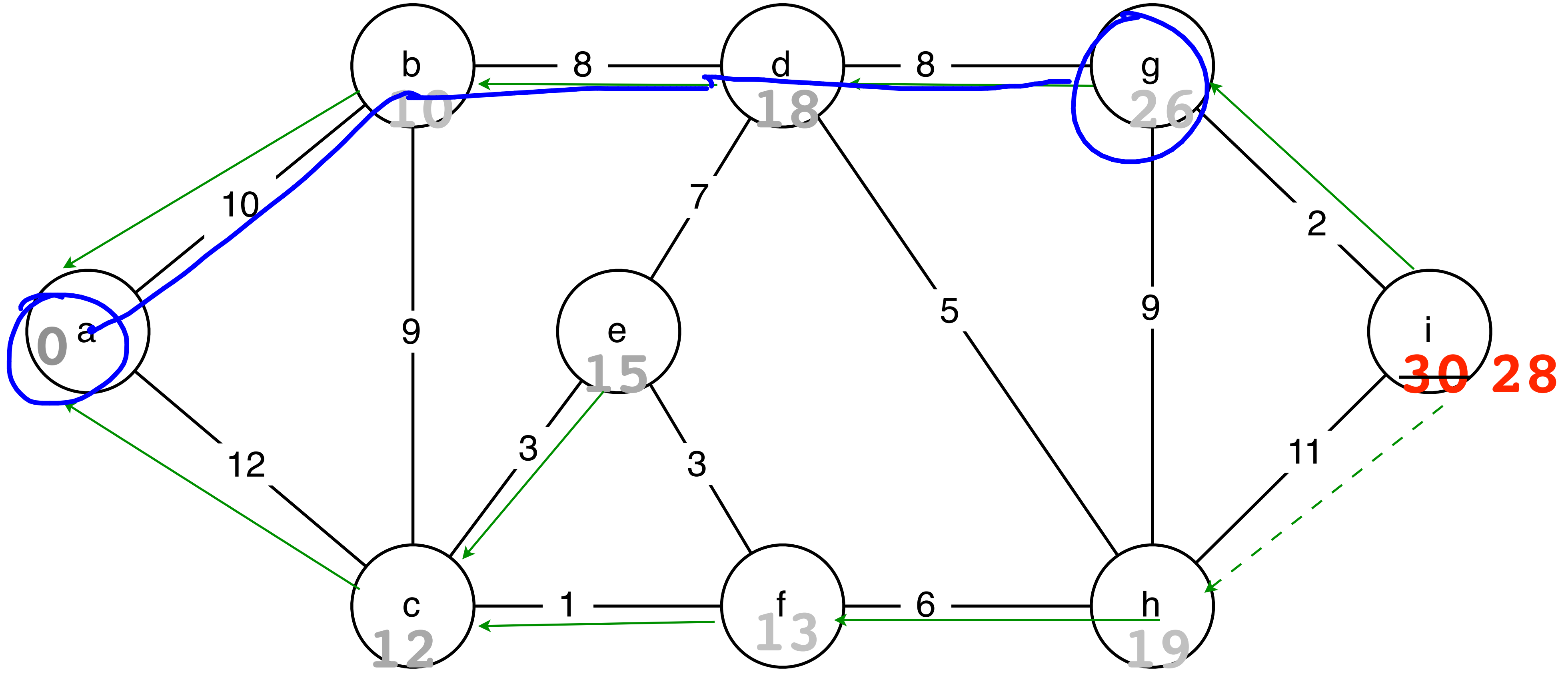

a
0
b
10
c
12
d
18
e
15
f
13
g
∞
h
19
i
∞
8
8
10
9
7
5
9
2
12
3
3
11
1
6


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


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


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

26



Algorithm

```
Dijkstra's (G, w, s) → s: source node, return, shortest paths to all nodes

① Intialize $d_v \leftarrow \infty$, $\pi_v \leftarrow$ nil
② Set $d_s \leftarrow 0$,
③ $Q \leftarrow$ make queue $(V, d_v)$
④ while Q is not empty
      $u \leftarrow$ extractmin(Q)
      for each neighbor v of u :
          if $d_v > d_u + w(u,v)$
              $\pi_v \leftarrow u$
              DecreaseKey$(Q, v, d_u + w(u,v))$
```

```
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($Q, v$)
```

runs $(V-1)$ times

runs E times

$\Theta(E \log V + V \log V)$ time or better.

```
$\text{DIJKSTRA}(G = (V, E), s)$
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($Q, v$)
```

```
PRIM($G = (V, E)$)
$Q \leftarrow \emptyset$    ▷ $Q$ is a Priority Queue
Initialize each $v \in V$ with key $k_v \leftarrow \infty$, $\pi_v \leftarrow$ NIL
Pick a starting node $r$ and set $k_r \leftarrow 0$
Insert all nodes into $Q$ with key $k_v$.
while $Q \neq \emptyset$
    do $u \leftarrow$ EXTRACT-MIN($Q$)
       for each $v \in Adj(u)$
           do if $v \in Q$ and $w(u, v) < k_v$
                 then $\pi_v \leftarrow u$
                      DECREASE-KEY($Q, v, w(u, v)$)    ▷ Sets $k_v \leftarrow w($
```

running time

```
DIJKSTRA($G = (V, E), s$)
for all $v \in V$
    do $\underline{d_u} \leftarrow \infty$
       $\pi_u \leftarrow$ NIL
$\underline{d_s} \leftarrow 0$
$Q \leftarrow$ 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 $d_v \leftarrow \underline{d_u + w(u, v)}$
                      $\pi_v \leftarrow u$
                      DECREASEKEY($Q, v$)
```

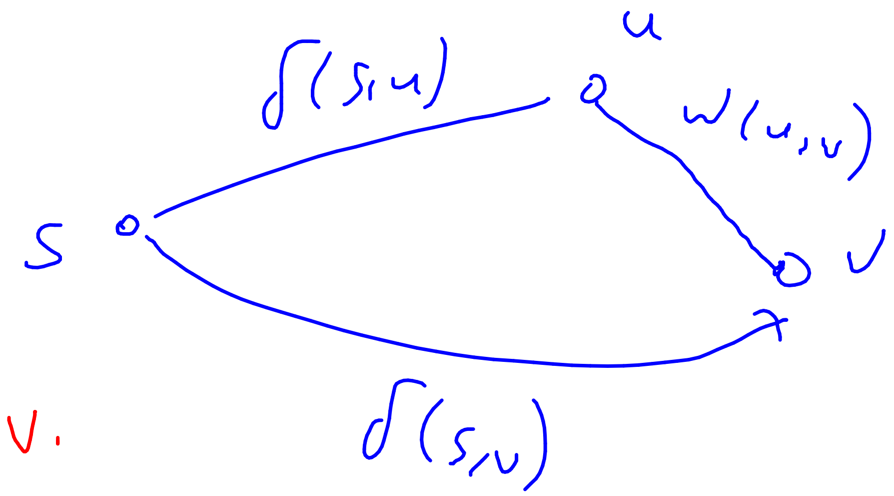
why does Dijkstra work?

triangle inequality:

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

This follows by definition of shortest paths, $\delta(s,u)$

$s \rightsquigarrow u \rightsquigarrow v$ is one path from s to v.



upper bound: $d_v \geq \delta(s,v)$

① $d_v \geq \delta(s,v)$. Follows because we initialize $d_v \leftarrow \infty$, and we only update d_v by using

$$d_v \leftarrow d_u + w(u,v)$$

Proof: Let S be all the nodes not in Q. At line 5, S is empty.

Property 1: For all $v \in S$, $d_v = \delta(s,v)$.

proof: By Induction. (P1) holds @ the start of the loop. Suppose it holds at iteration i of the loop. Consider iteration $i+1$. Line 7 we extract a node u. Only u is added to S. Now, lets argue that $d_u = \delta(s,u)$.

Spse for the sake of reaching a contradiction, $d_u \neq \delta(s,u)$ (*)

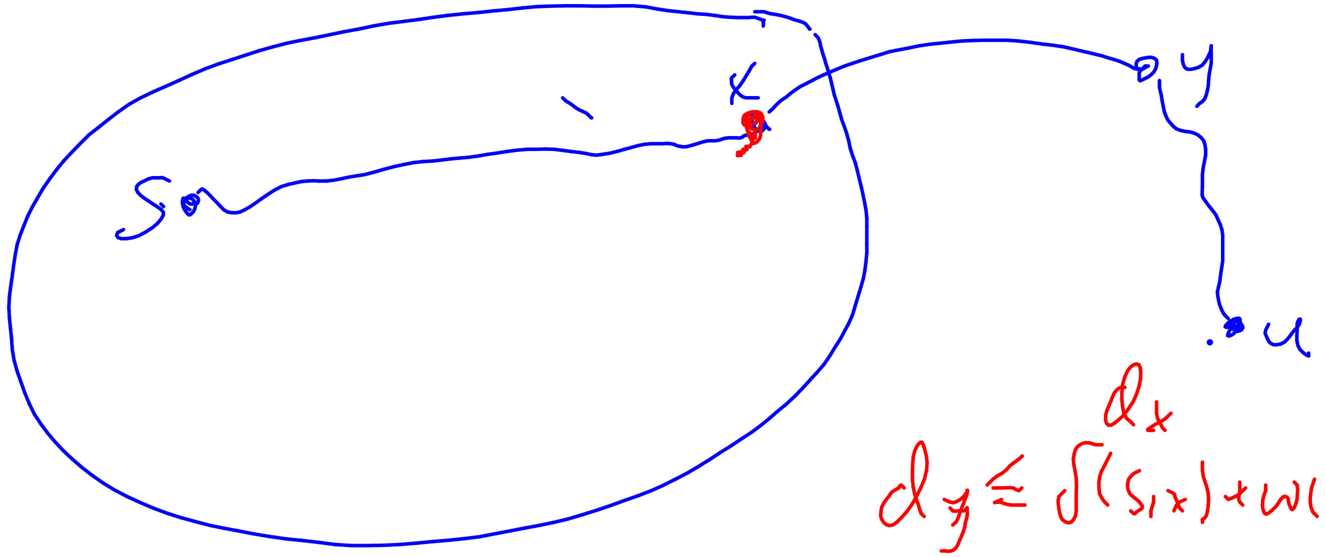
① $d_u \geq \delta(s,u)$. $\Rightarrow$ $\exists$ some path from s to u.

If there was no path, $\delta(s,u) = \infty$.

By *, we know that $d_u \neq \delta(s,u)$ but also that $d_u \geq \delta(s,u)$ $\Rightarrow$ $\delta(s,u)$ cannot be ∞.

Consider any path from s to u.

Let x be the very last node in set S along path p. (x, y) is first edge to cross the cut (S, V−S) where S is the set right before u is extracted.



d_x

$d_y \le \delta(s,x) + w(x,y)$

$\ell(p^*) \ge d_u$

$$\underline{\ell(p)} = \underline{\overset{d_x}{\delta(s,x)} + w(x,y)} + \ell(y \rightsquigarrow u)$$

$$\ge d_y + \ell(y \rightsquigarrow u)$$ why??

① x is in S @ this time. when line 10 of the algorithm was run for x, y was a neighbor of x and so $d_y \leftarrow \delta(s,x) + w(x,y)$

$$\ge d_u + \ell(y \rightsquigarrow u)$$ why?

Because in this iteration, u was chosen as extractmin, so $d_u \le d_y$

$$\underline{\ge d_u}$$ why??

No negative weights.

We have shown that the length of $\underline{\text{any}}$ path from $s \rightsquigarrow u$, $\ell(s \rightsquigarrow u) \geq d_u$.

We also know that $d_u \geq \delta(s,u)$

length of any path from s to u

$$\Rightarrow \underset{\overset{\uparrow}{\underline{p^*}}}{\underline{\ell(s \rightsquigarrow u)}} \geq \underline{\underline{d_u}} \geq \underset{\overset{\uparrow}{\ell(p^*)}}{\underline{\delta(s,u)}} \Rightarrow \underline{\underline{\underline{d_u} = \delta(s,u)}}.$$

☒

breadth first search

input: $\underline{G = (V, E), s}$

output:

$w(e) = 1$

$\delta(s,v)$ for all v where $w(e) = 1$ $e \in E$.

breadth first search

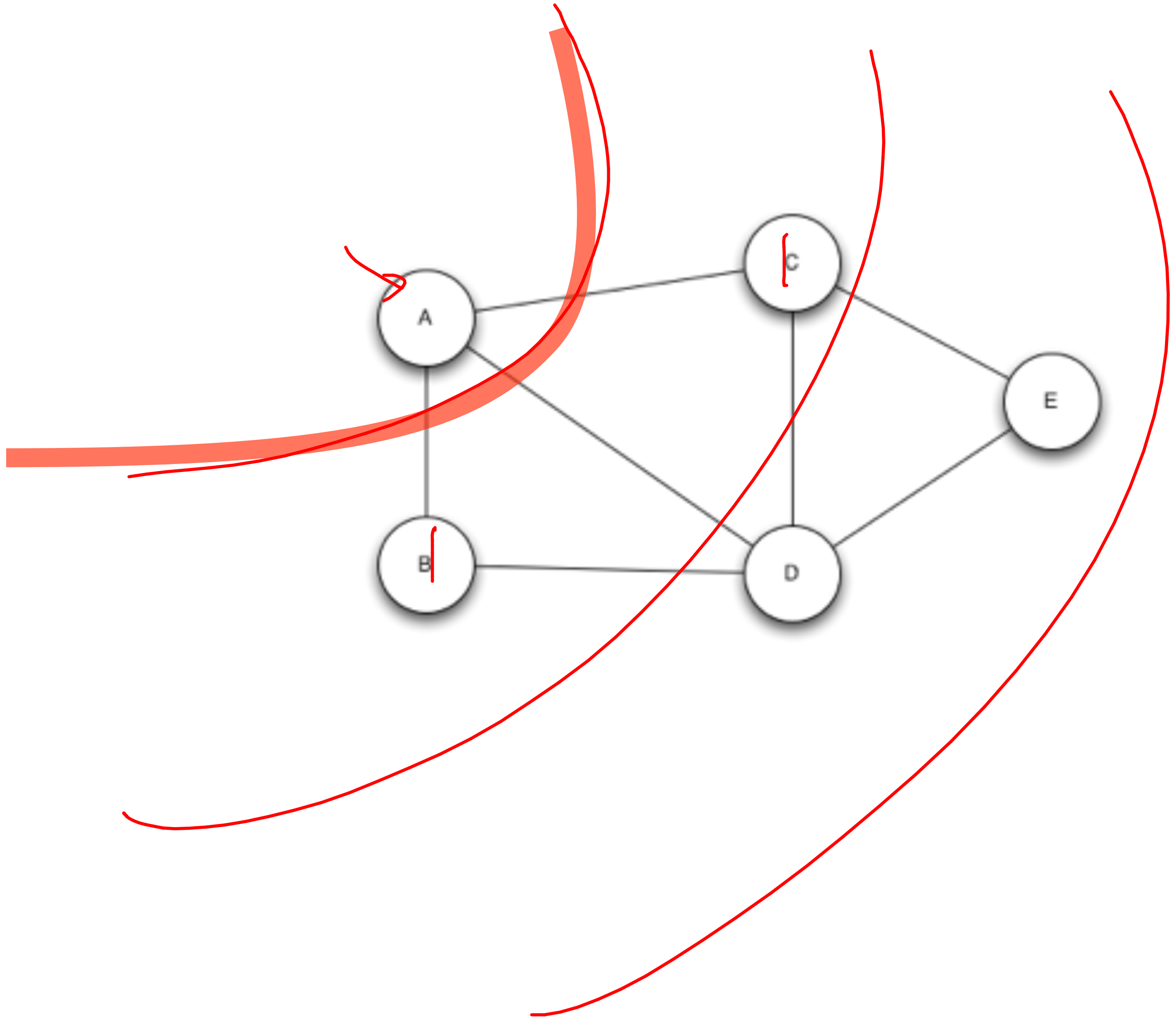
input: $G = (V, E), s$

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

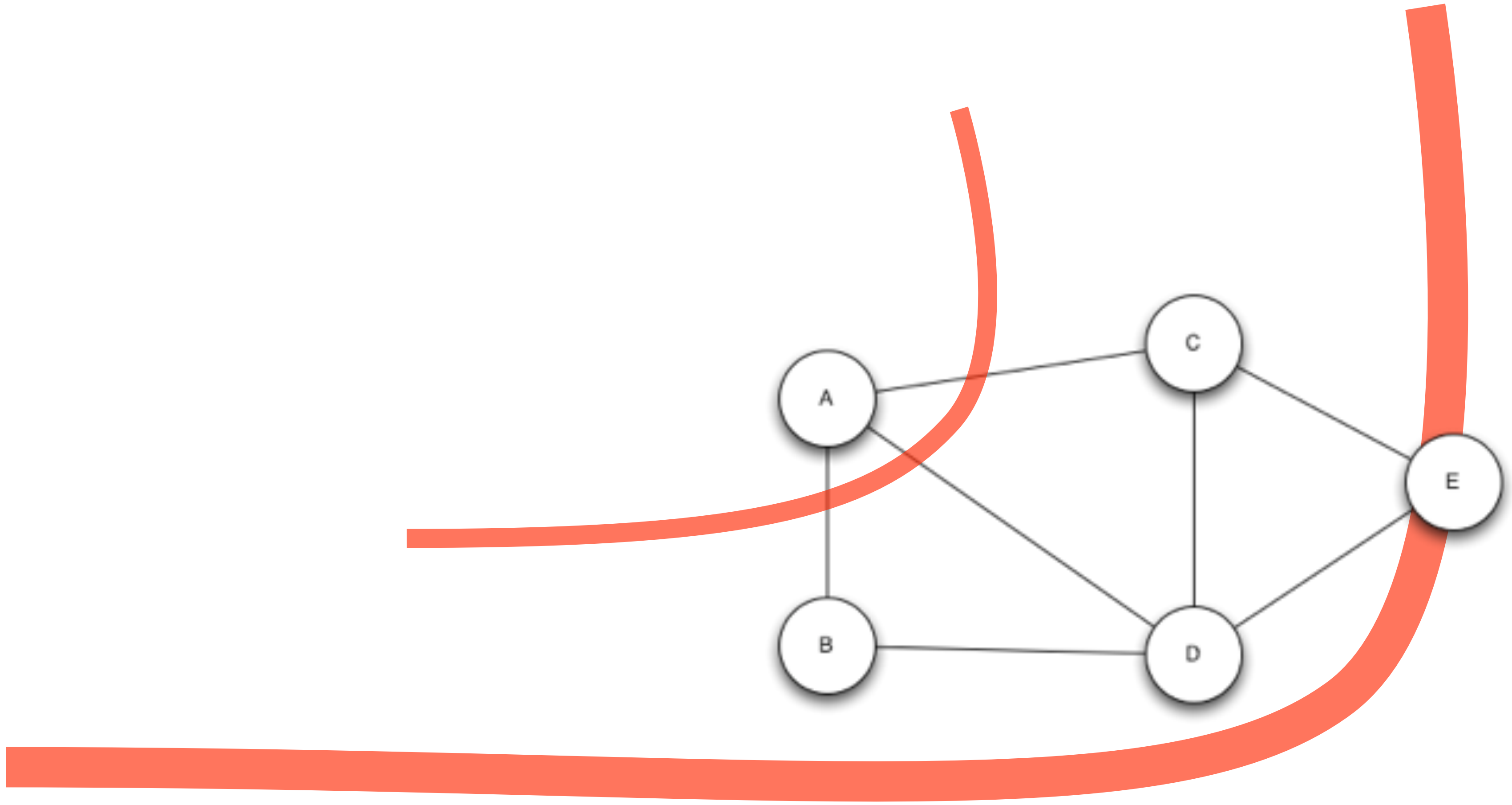
smallest # of edges from s to v

$w(e) = 1$!!

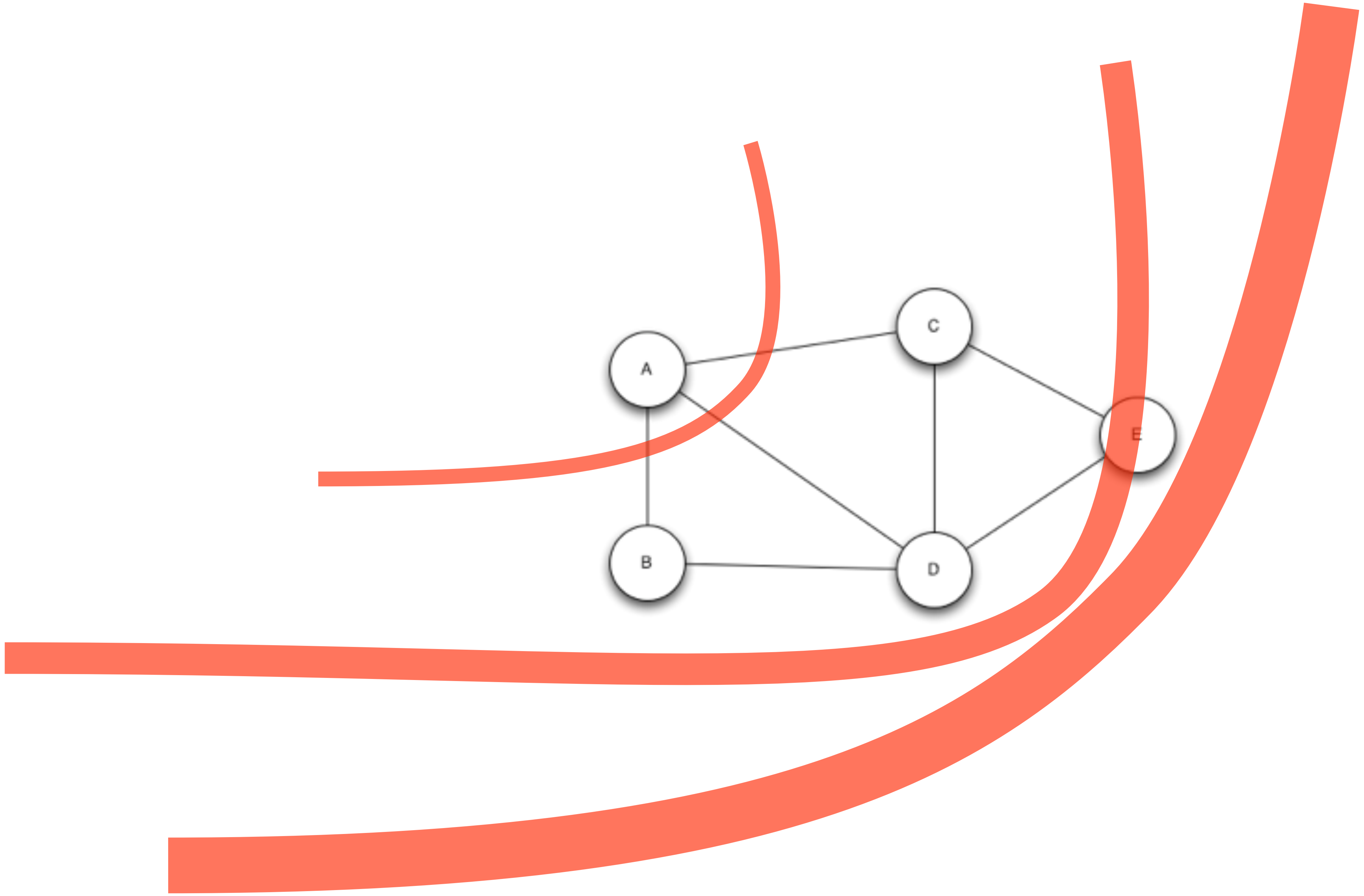
breadth-first search



breadth-first search



breadth-first search

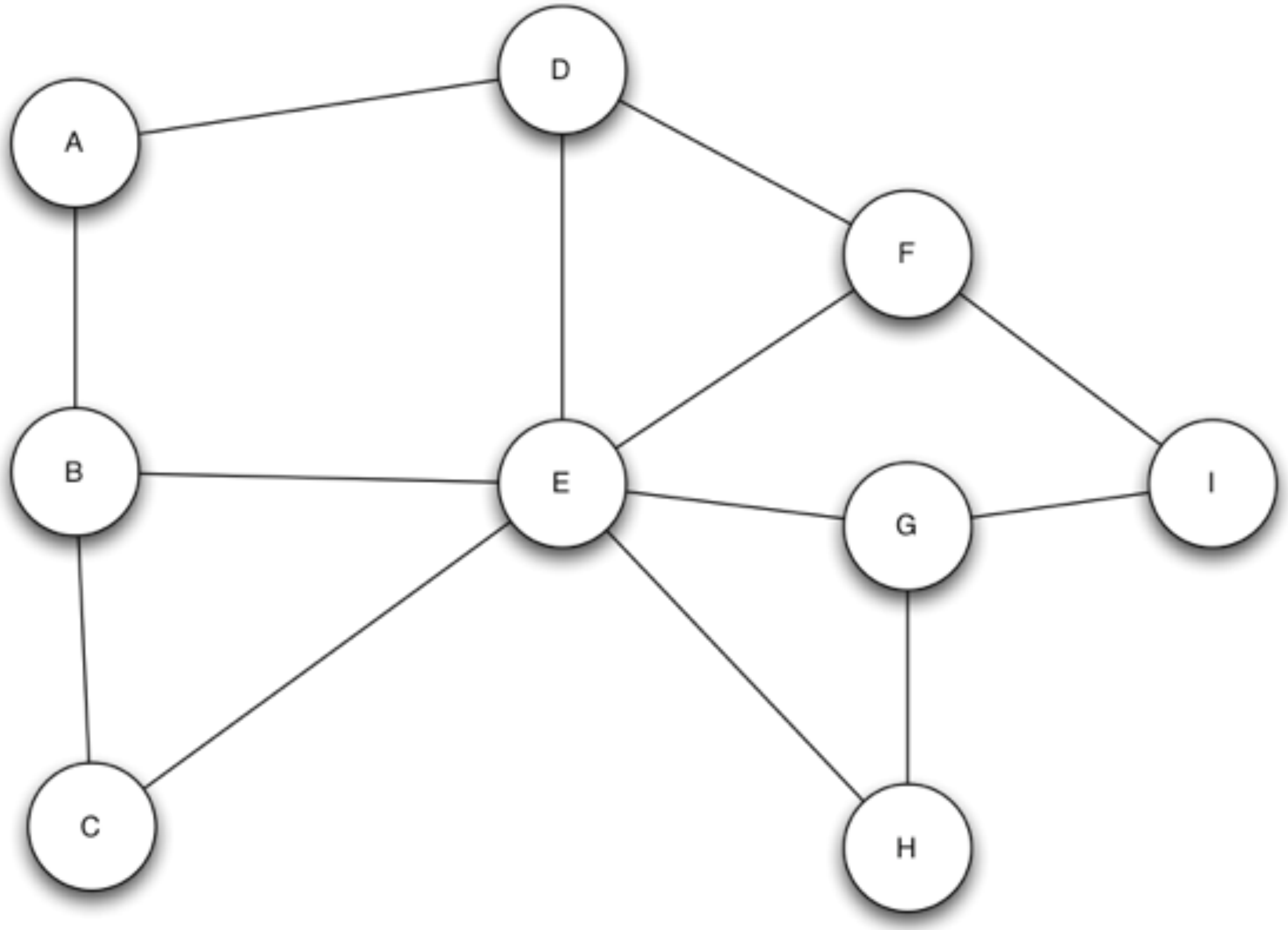


breadth first search

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

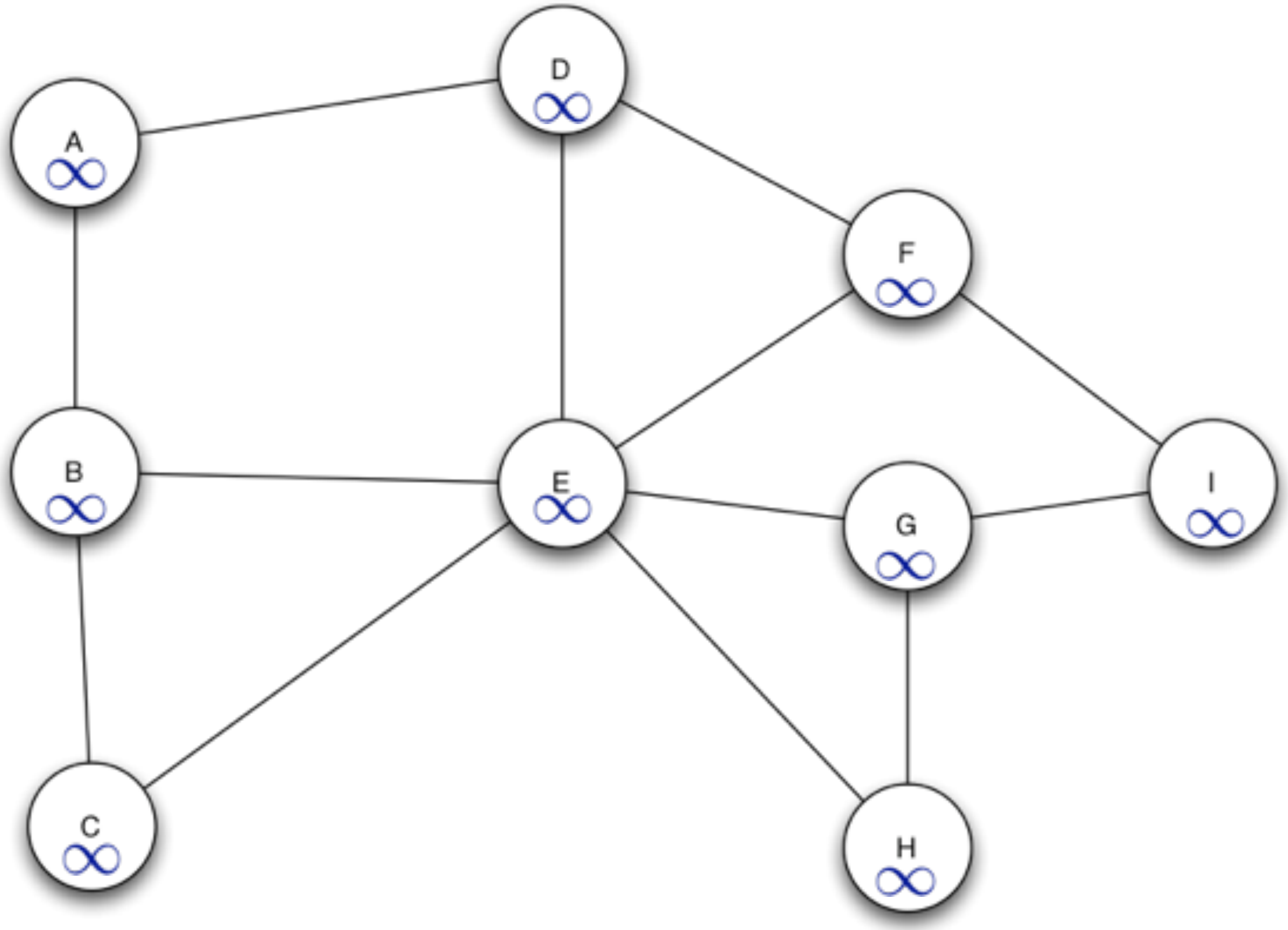
output: smallest # of edges from s to v $d_v \quad \forall v \in V$

bfs(G, a)



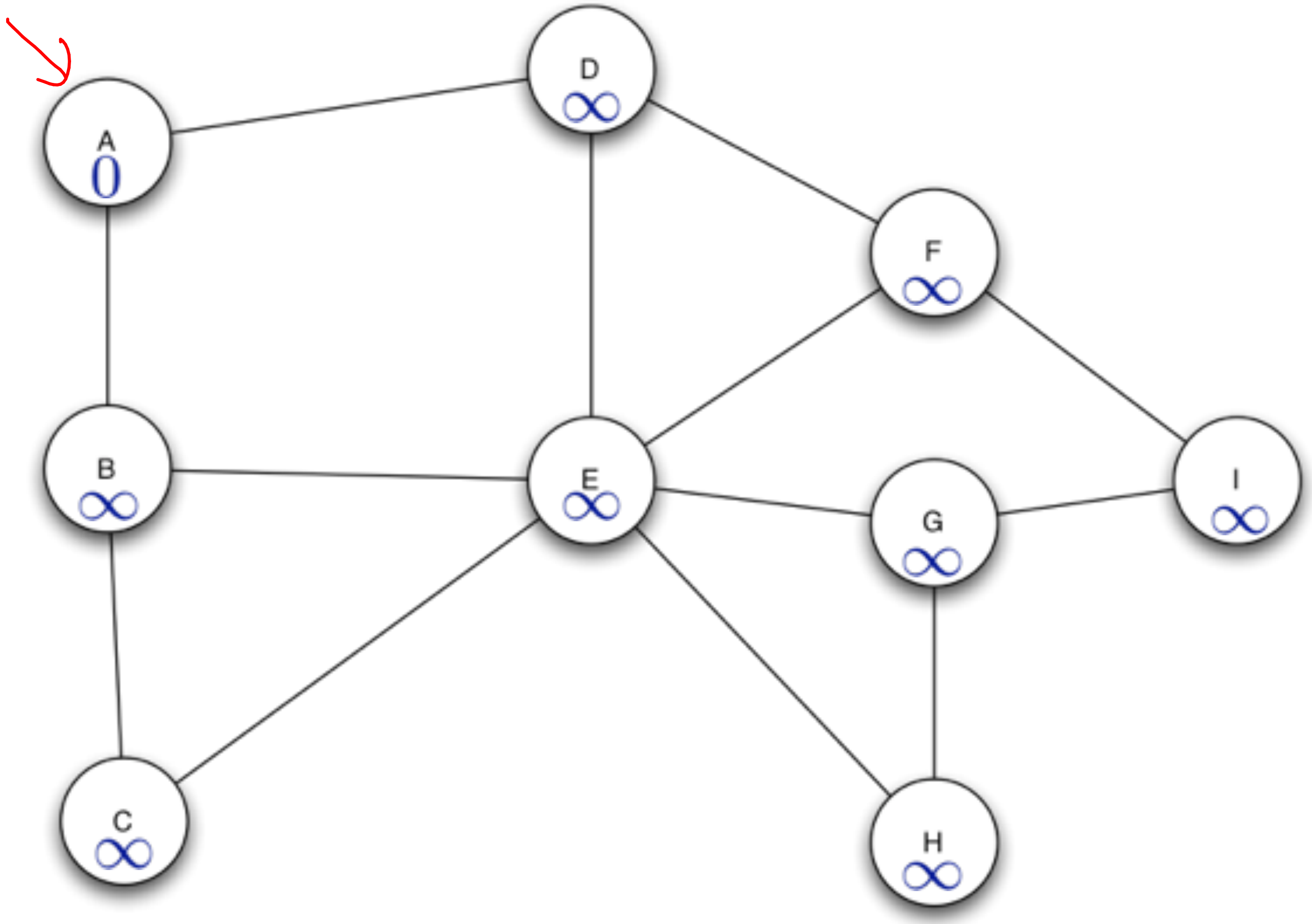
$\underline{Q}$

bfs(G, a)



Q

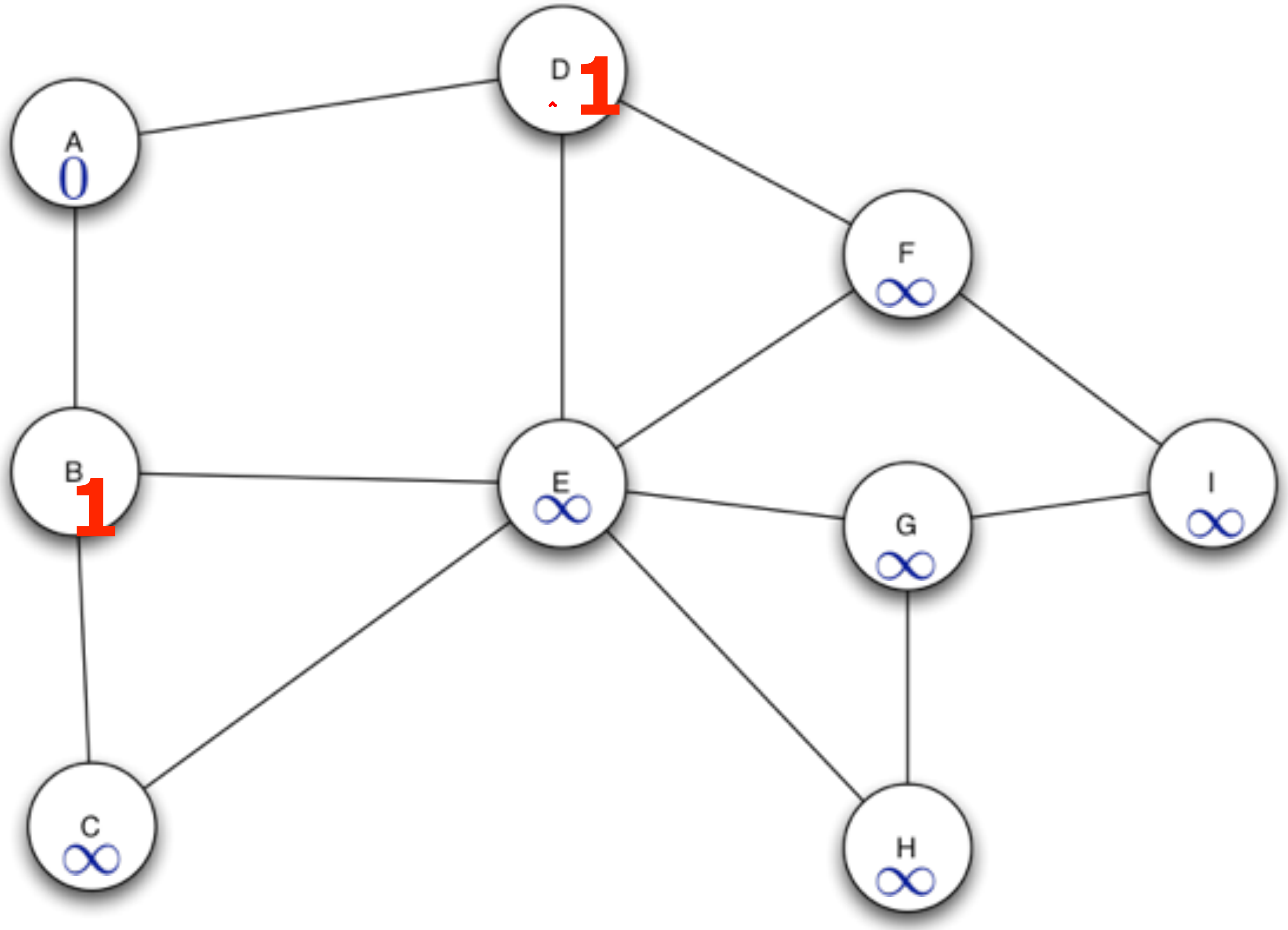
bfs(G, a)



Q

a

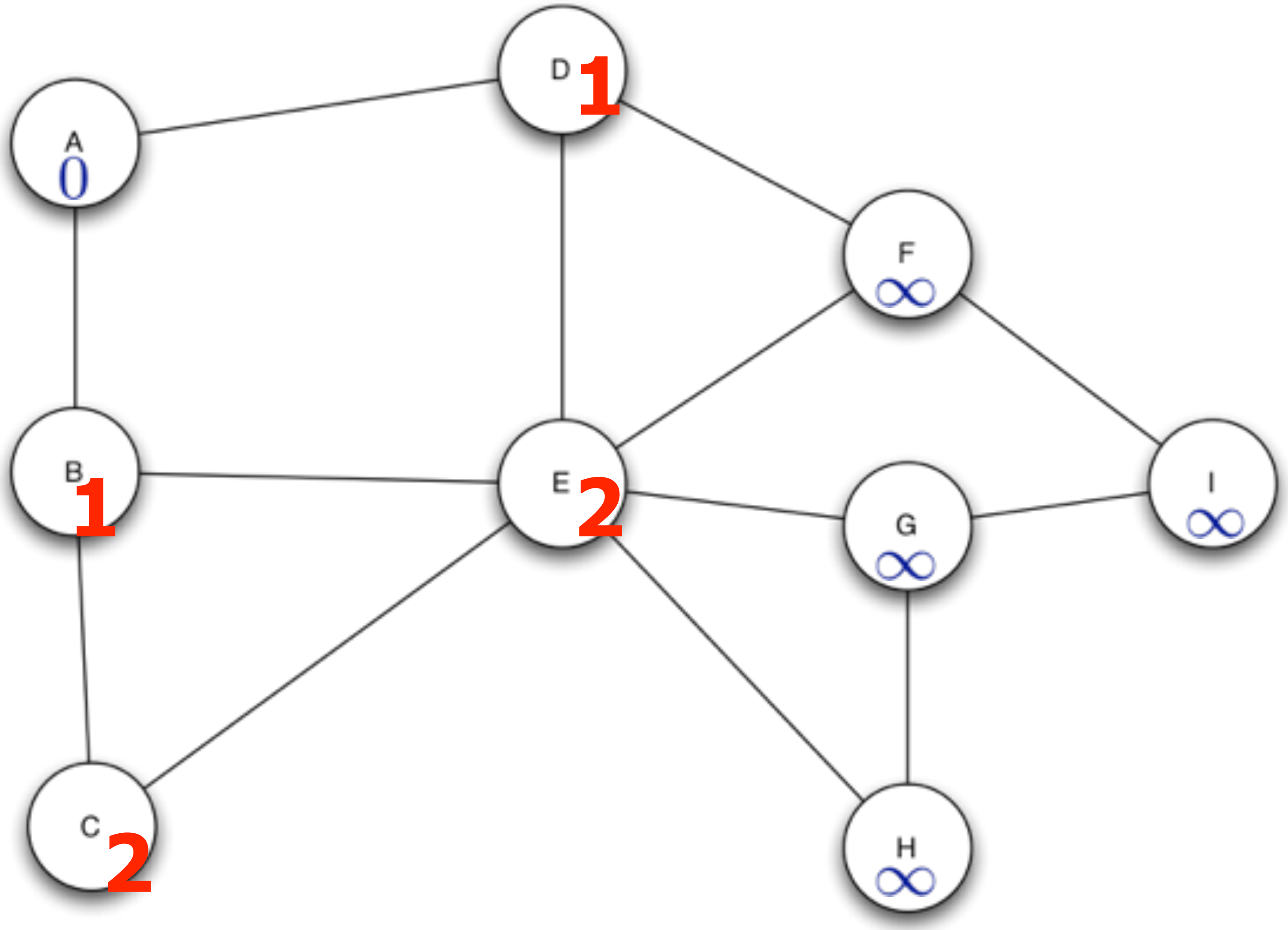
bfs(G, a)



Q

~~a~~
b
d

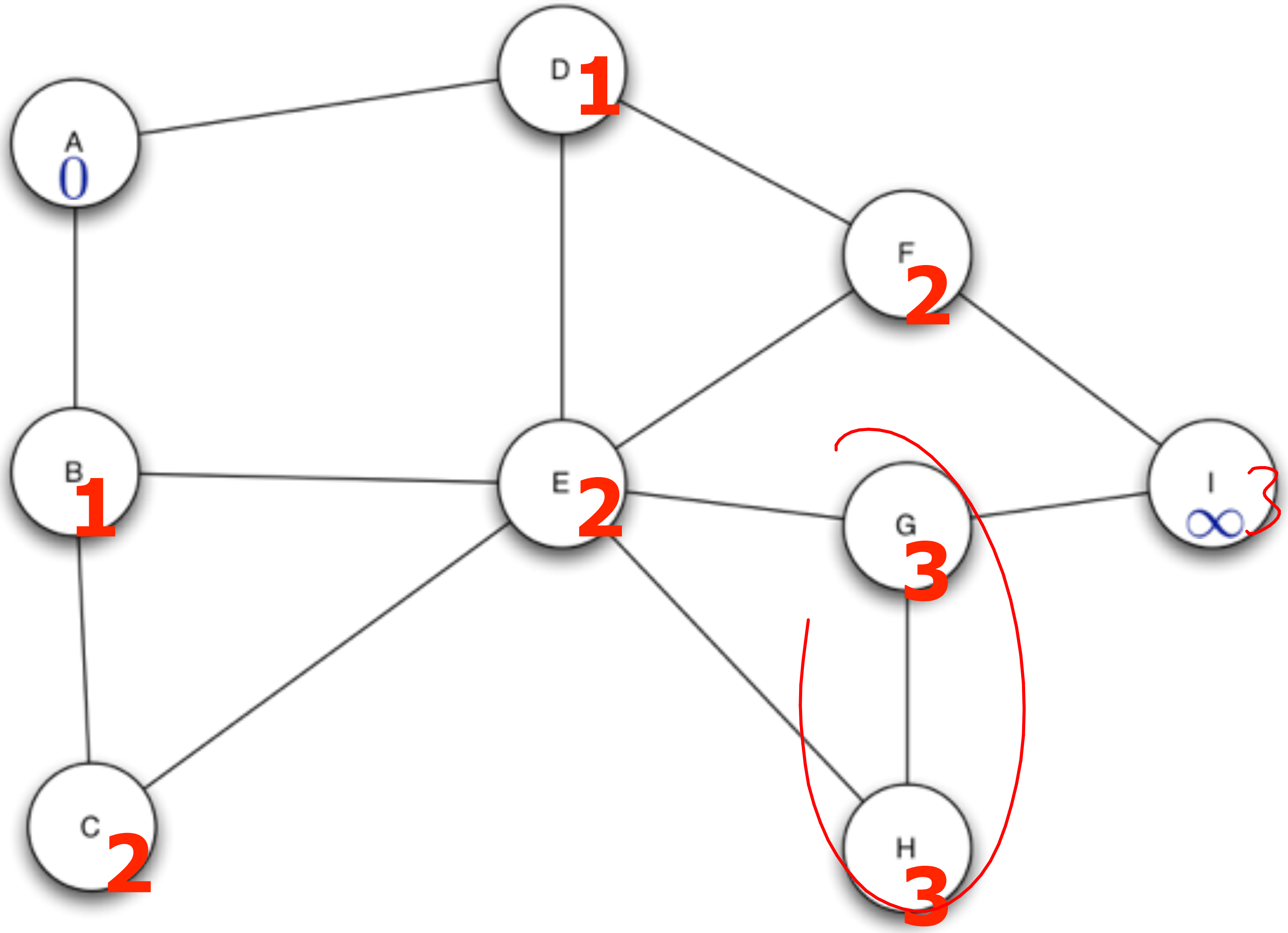
bfs(G, a)



Q

a
~~b~~
d
c
e

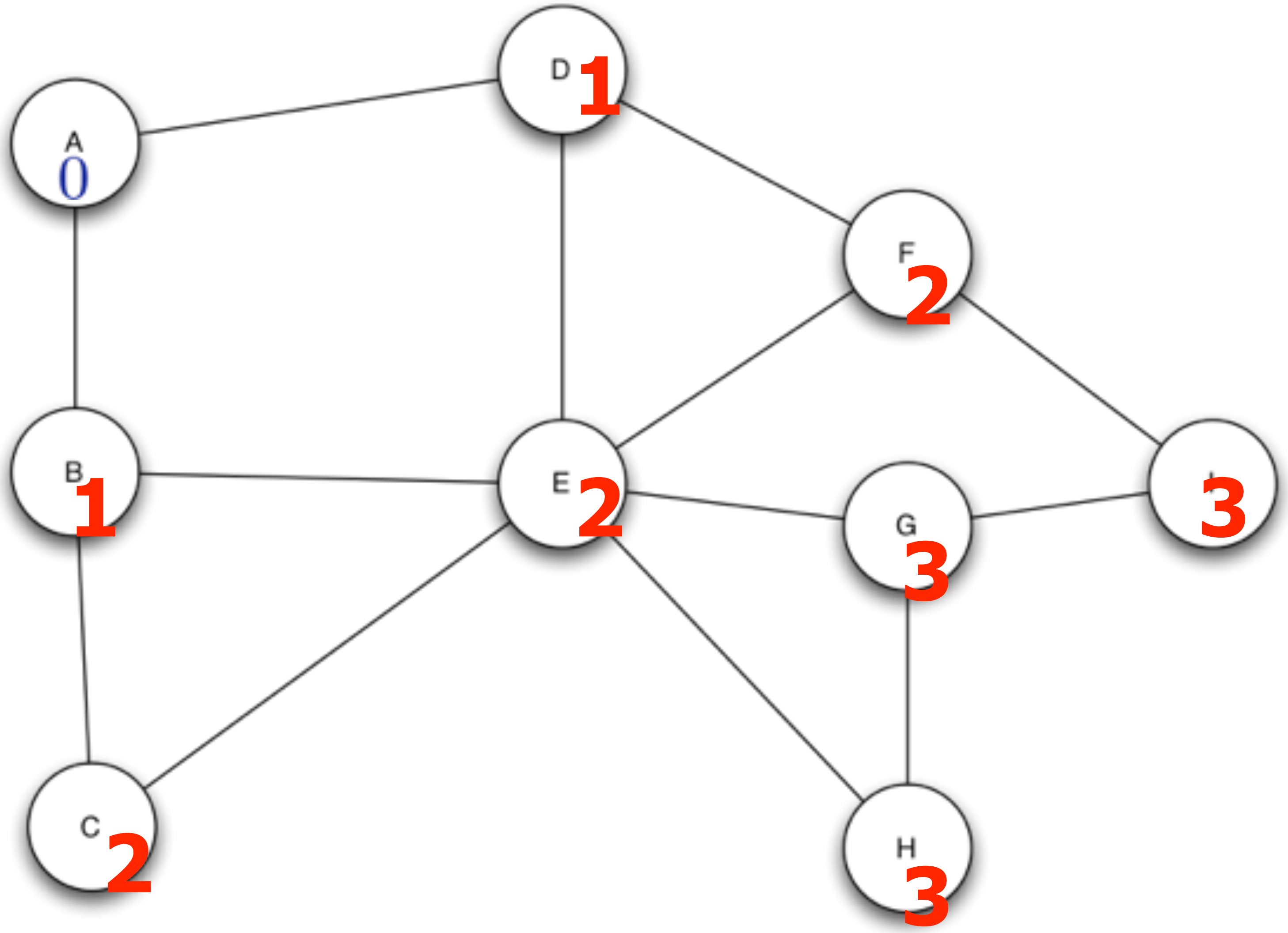
bfs(G, a)



Q

a
b
~~d~~
~~c~~
e
f
g
h

bfs(G, a)



Q

a
b
d
c
e

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($Q$)
        for each $v \in Adj[u]$
            do if $d[v] = \infty$
                then $d[v] \leftarrow d[u] + 1$
                    ENQUEUE($Q, v$)
```

Q

① instead of priority Queue.

b/c $w(e) = 1$,

equivalent to extractmin.

decreasekey

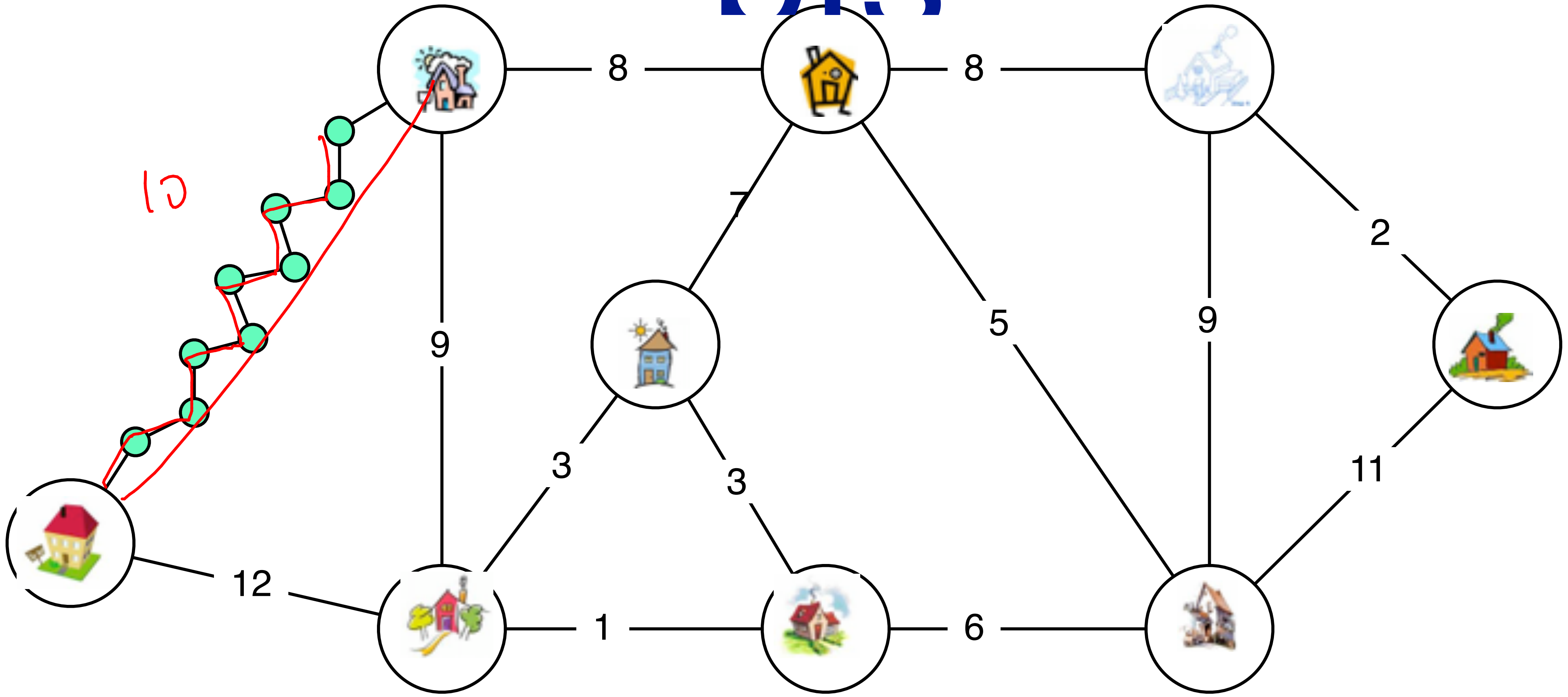
$\Theta(V+E)$

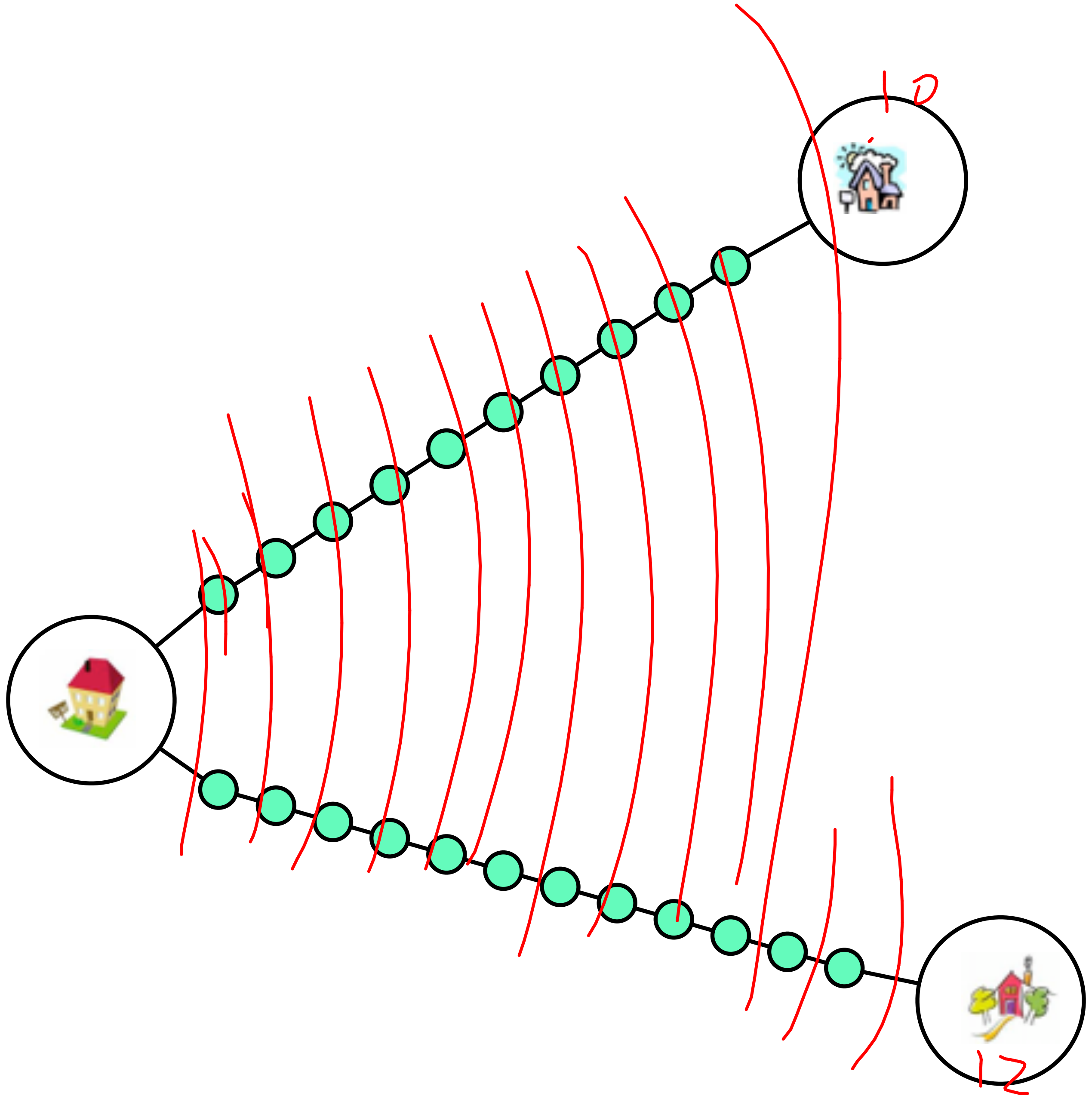
BFS theorem

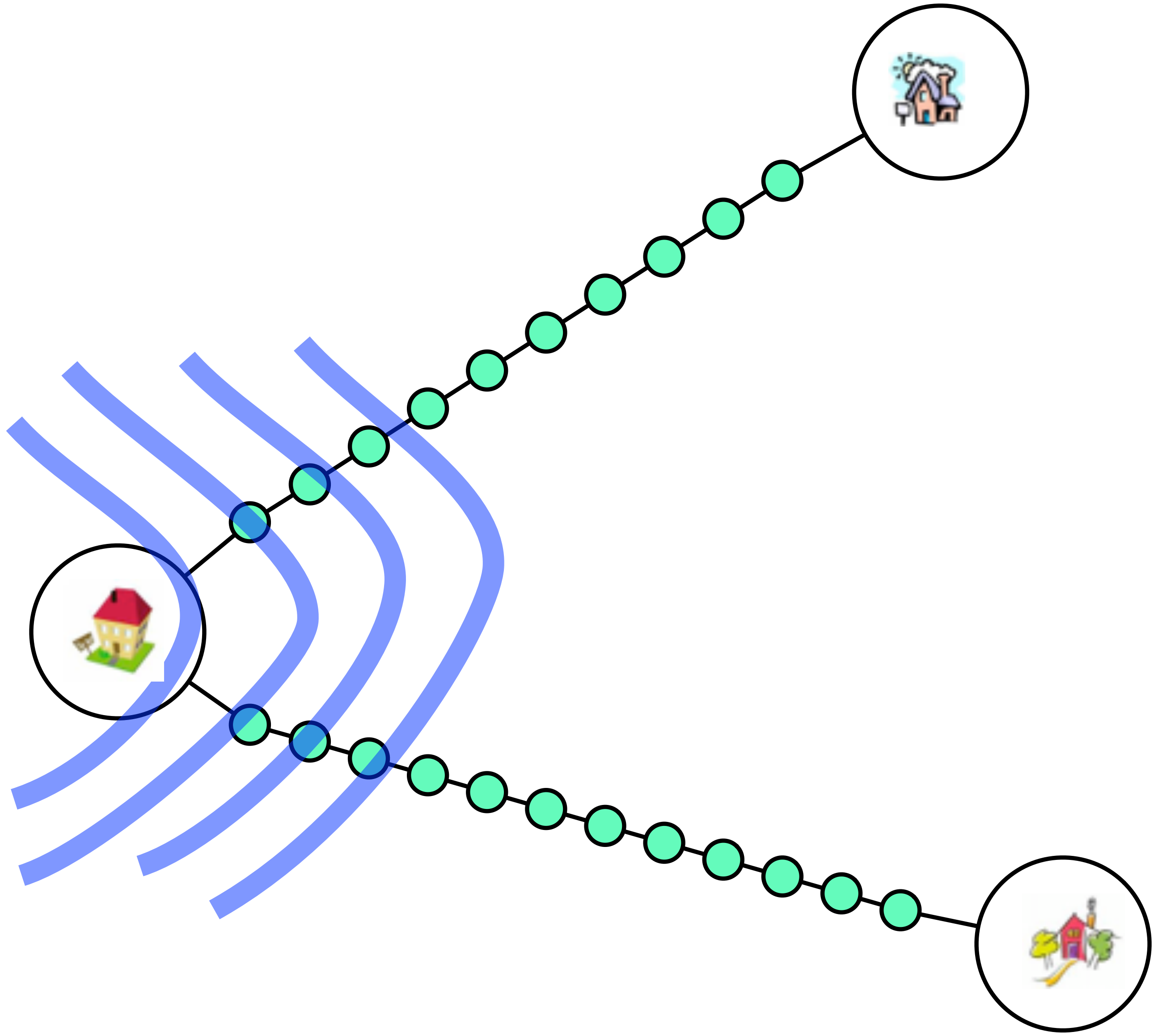
```
BFS$(V, E, s)$
for each $u \in V - \{s\}$
    do $d[u] \leftarrow \infty$
$d[s] \leftarrow 0$
$Q \leftarrow \emptyset$
ENQUEUE$(Q, s)$
while $Q \neq \emptyset$
    do $u \leftarrow$ DEQUEUE$(Q)$
        for each $v \in Adj[u]$
            do if $d[v] = \infty$
                then $d[v] \leftarrow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

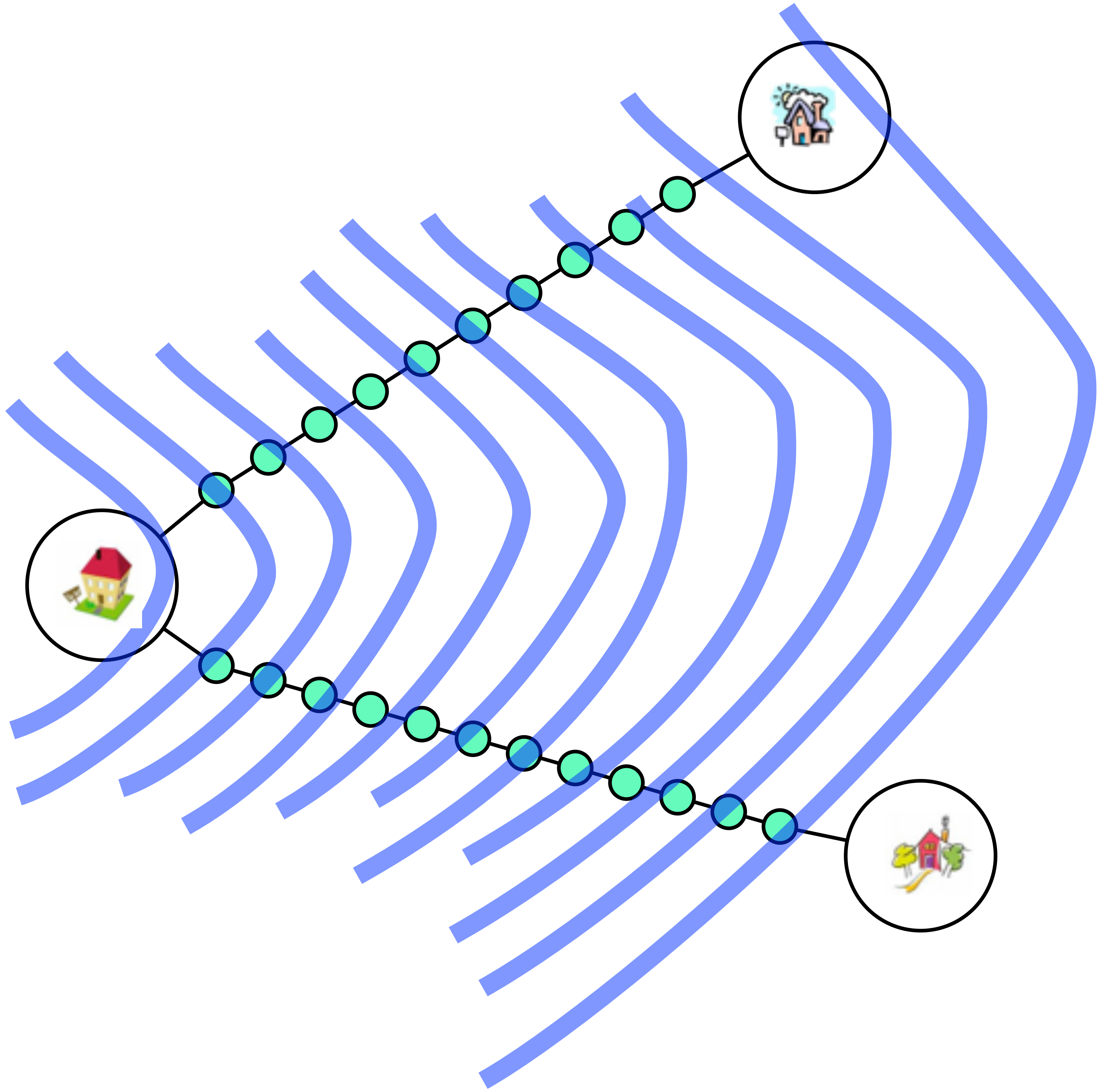
```
DIJKSTRA$(G = (V, E), s)$
for all $v \in V$
    do $d_u \leftarrow \infty$
       $\pi_u \leftarrow$ NIL
$d_s \leftarrow 0$
$Q \leftarrow$ MAKEQUEUE$(V)$      ▷ use $d_u$ as key
while $Q \neq \emptyset$
    do $u \leftarrow$ EXTRACTMIN$(Q)$
       for each $v \in Adj(u)$
           do if $d_v > d_u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

bfs




10
12





shortest paths

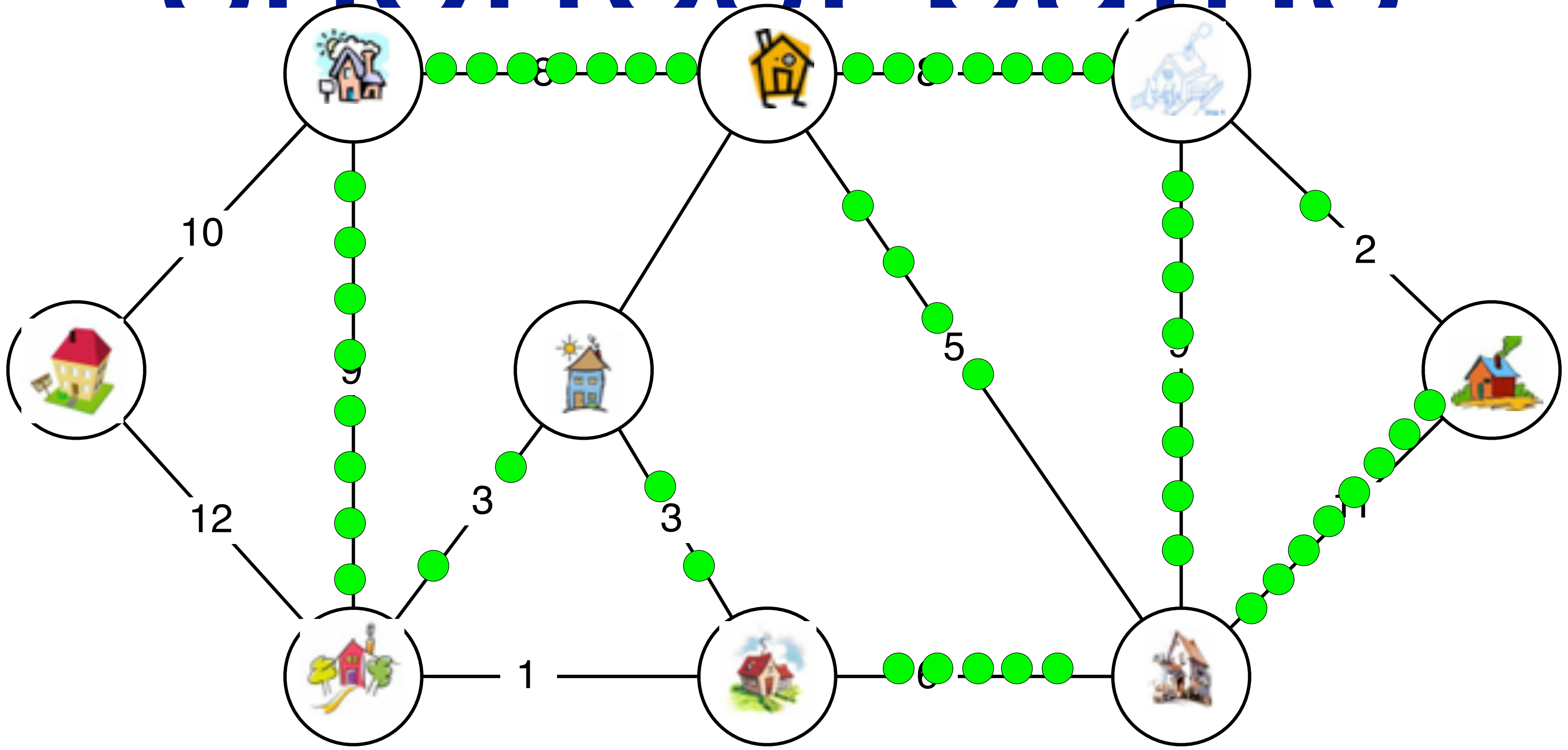
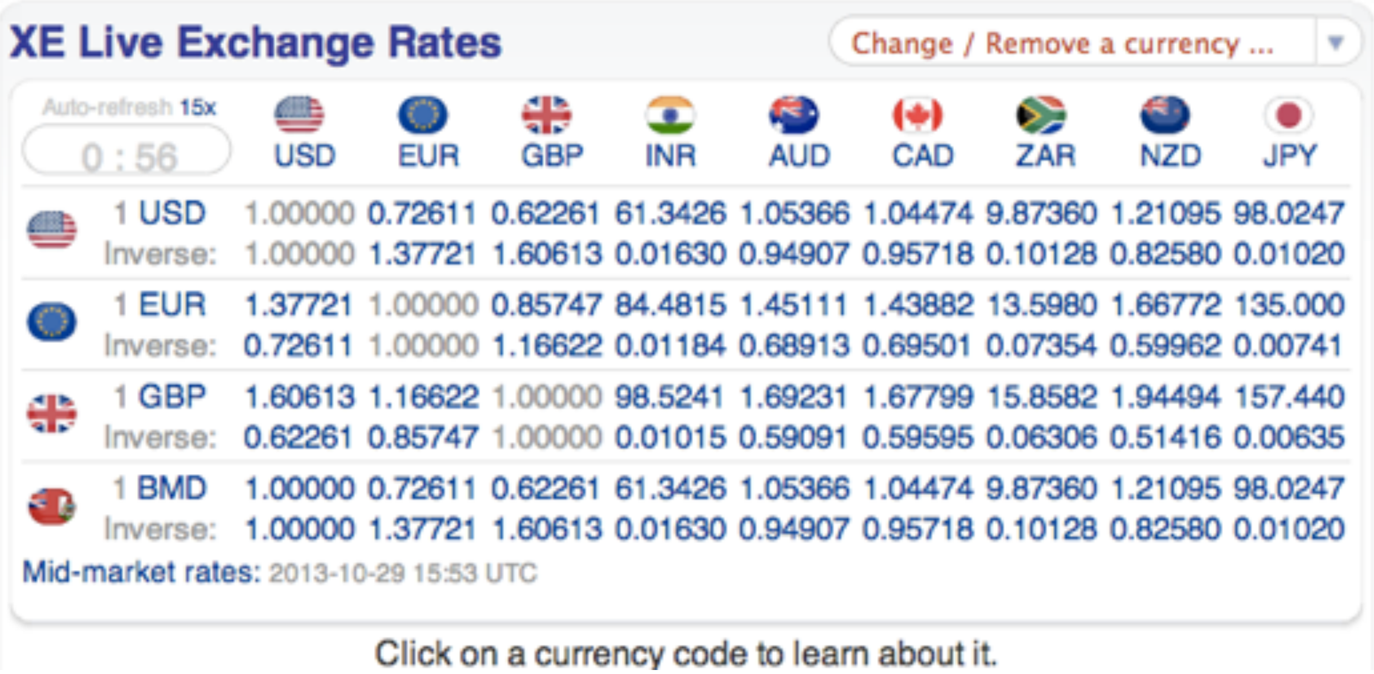


image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

What about Negative edge weights?



XE Live Exchange Rates

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.

where does old argument break down

first ideas: Add to each edge

sssp(G,s)

$$\text{SHORT}_{i,v} =$$

sssp(G,s)

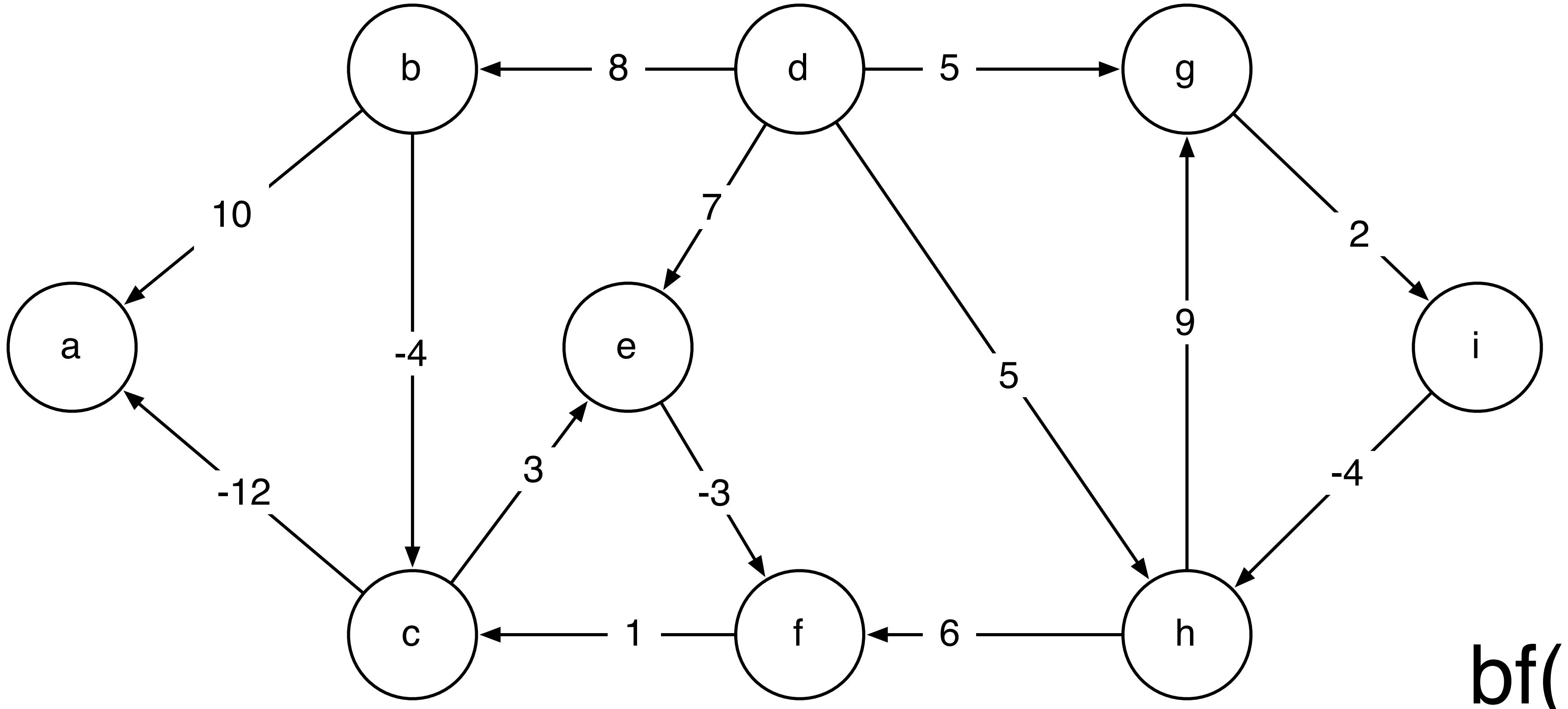
$$\textsc{short}_{i,v} = \begin{cases} \infty & i = 0 \\ 0 & v = s \\ \min_{x \in V} & \left\{ \begin{array}{l} \textsc{short}_{i-1,v} \\ \textsc{short}_{i-1,x} + w(x,v) \end{array} \right\} \end{cases}$$

max len of a simple path:

bellman-ford(G,s)

```
BELLMAN-FORD$(G, s)$
SHORT$_{0,s} \leftarrow 0$
$\forall v \in V - \{s\}$, SHORT$_{0,v} \leftarrow \infty$
for $i = 1, \ldots, V - 1$
    do for each $v \in V - \{s\}$
        do SHORT$_{i,v} = \min_{x \in Adj(v)} \left\{ \begin{array}{l} \text{SHORT}_{i-1,v} \\ w(x,v) + \text{SHORT}_{i-1,x} \end{array} \right\}$
```

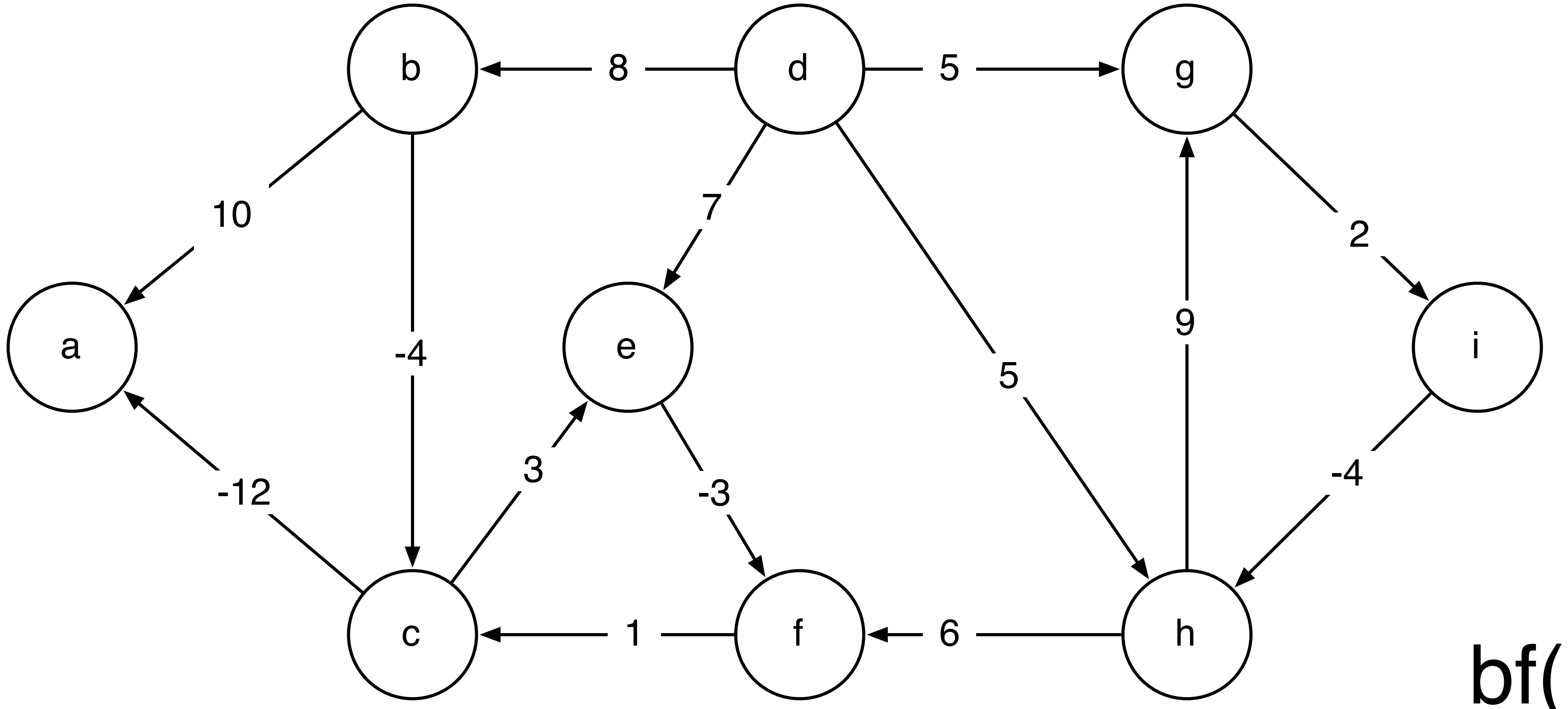
```
BELLMAN-FORD$(G, s)$
$\textsc{short}_{0,s} \leftarrow 0$
$\forall v \in V - \{s\}$, $\textsc{short}_{0,v} \leftarrow \infty$
for $i = 1, \ldots, V - 1$
    do for each $e = (x, y) \in E$
        do $\textsc{short}_{i,y} = \min \left\{ \begin{array}{l} \textsc{short}_{i-1,y} \\ \textsc{short}_{i,y} \\ w(x,y) + \textsc{short}_{i-1,x} \end{array} \right\}$
```



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

bf(G,d)

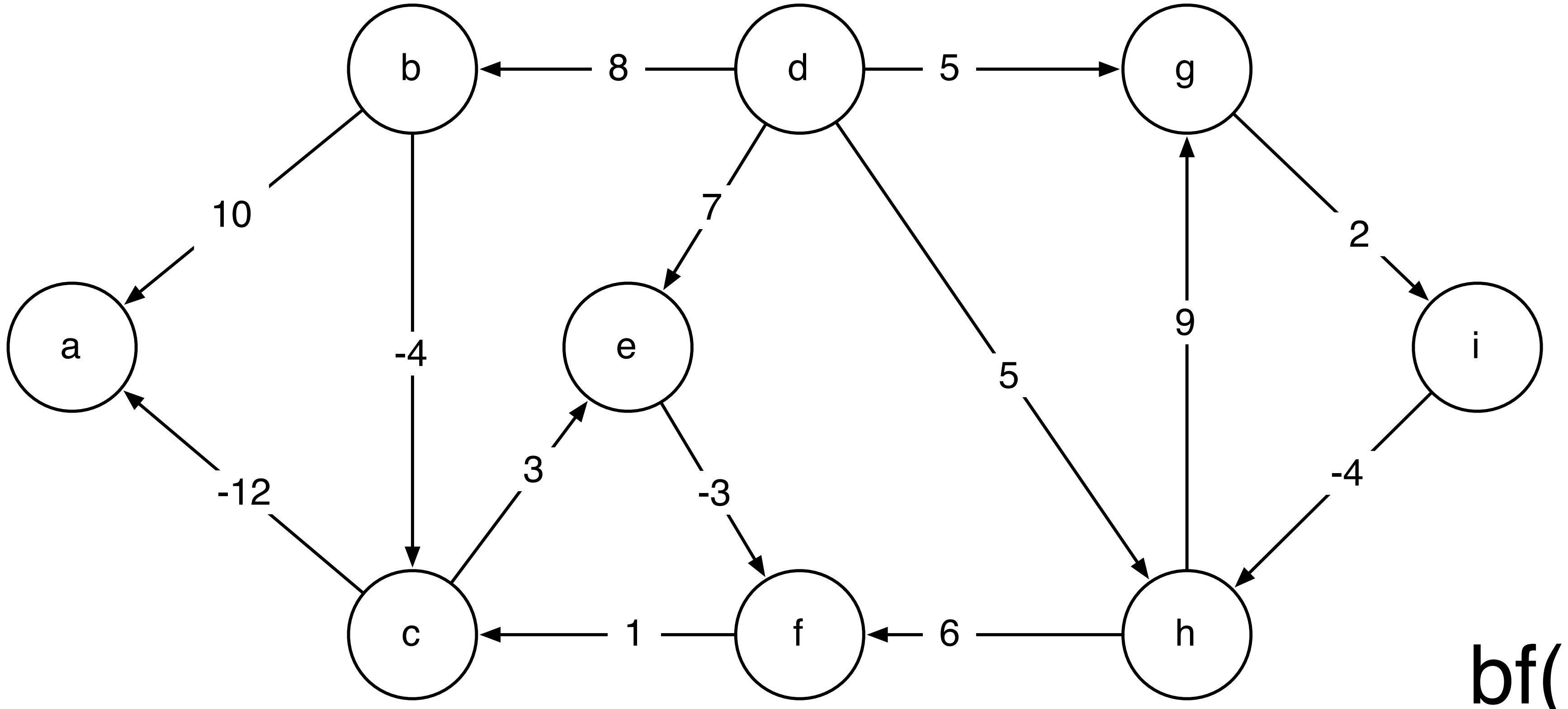
	0	1	2	3	4	5	6	7
a								
b								
c								
d	0							
e								
f								
g								
h								
i								



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

bf(G,d)

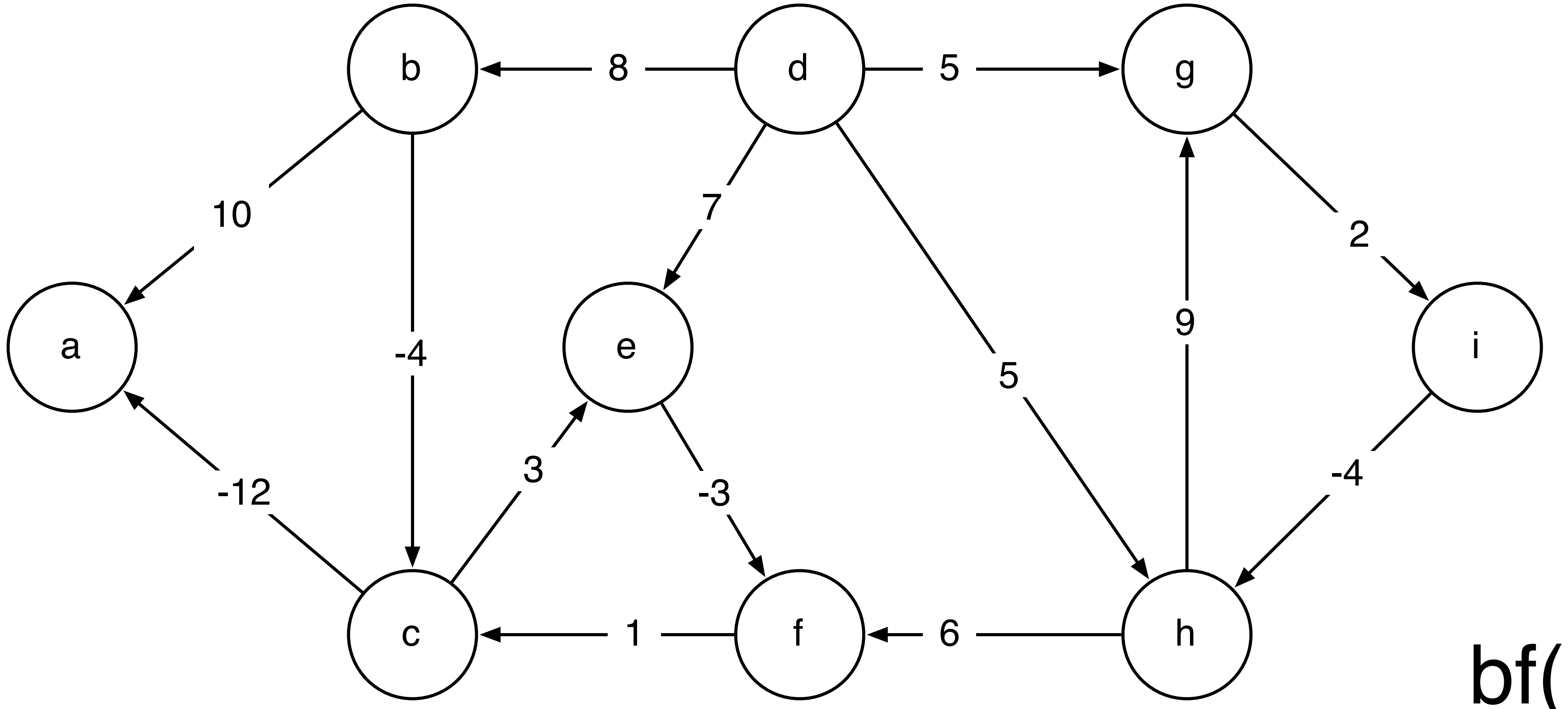
	0	1	2	3	4	5	6	7
a	∞							
b	∞							
c	∞							
d	0							
e	∞							
f	∞							
g	∞							
h	∞							
i	∞							



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

bf(G,d)

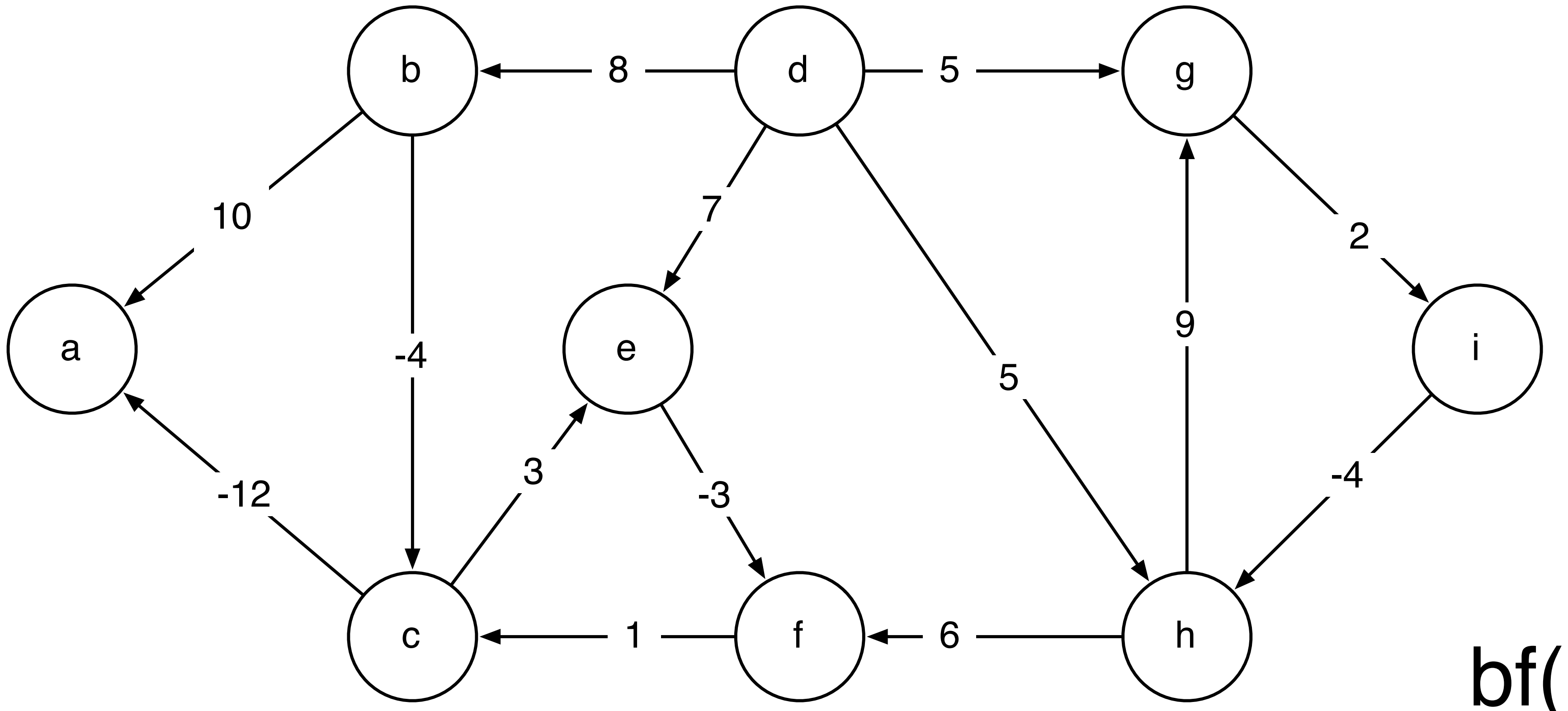
	0	1	2	3	4	5	6	7
a	∞							
b	∞	8						
c	∞							
d	0	0						
e	∞	7						
f	∞							
g	∞	5						
h	∞	5						
i	∞							



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

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

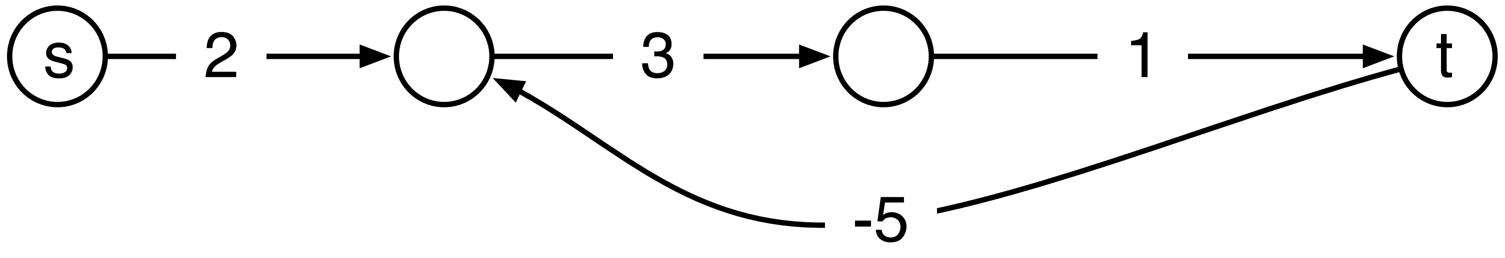
```
BELLMAN-FORD$(G, s)$
$\text{SHORT}_{0,s} \leftarrow 0$
$\forall v \in V - \{s\}$, $\text{SHORT}_{0,v} \leftarrow \infty$
for $i = 1, \ldots, V - 1$
    do for each $e = (x, y) \in E$
        do $\text{SHORT}_{i,y} = \min \left\{ \begin{array}{l} \text{SHORT}_{i-1,y} \\ \text{SHORT}_{i,y} \\ w(x,y) + \text{SHORT}_{i-1,x} \end{array} \right\}$
```

```
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

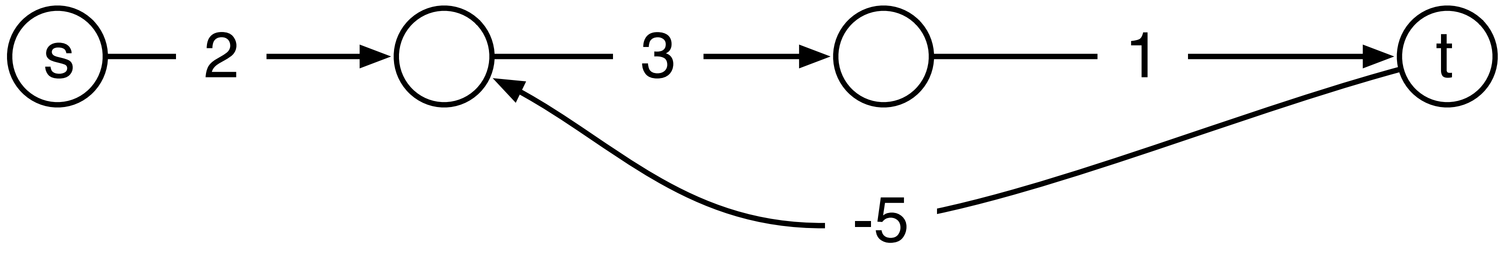
```
BELLMAN-FORD$(G, s)$
$d_s \leftarrow 0$
$\forall v \in V - \{s\},\ d_v \leftarrow \infty$
for $i = 1, \ldots, V - 1$
        do for each $e = (x, y) \in E$
                do $d_y \leftarrow \min\{\ d_y, w(x, y) + d_x\ \}$
```

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

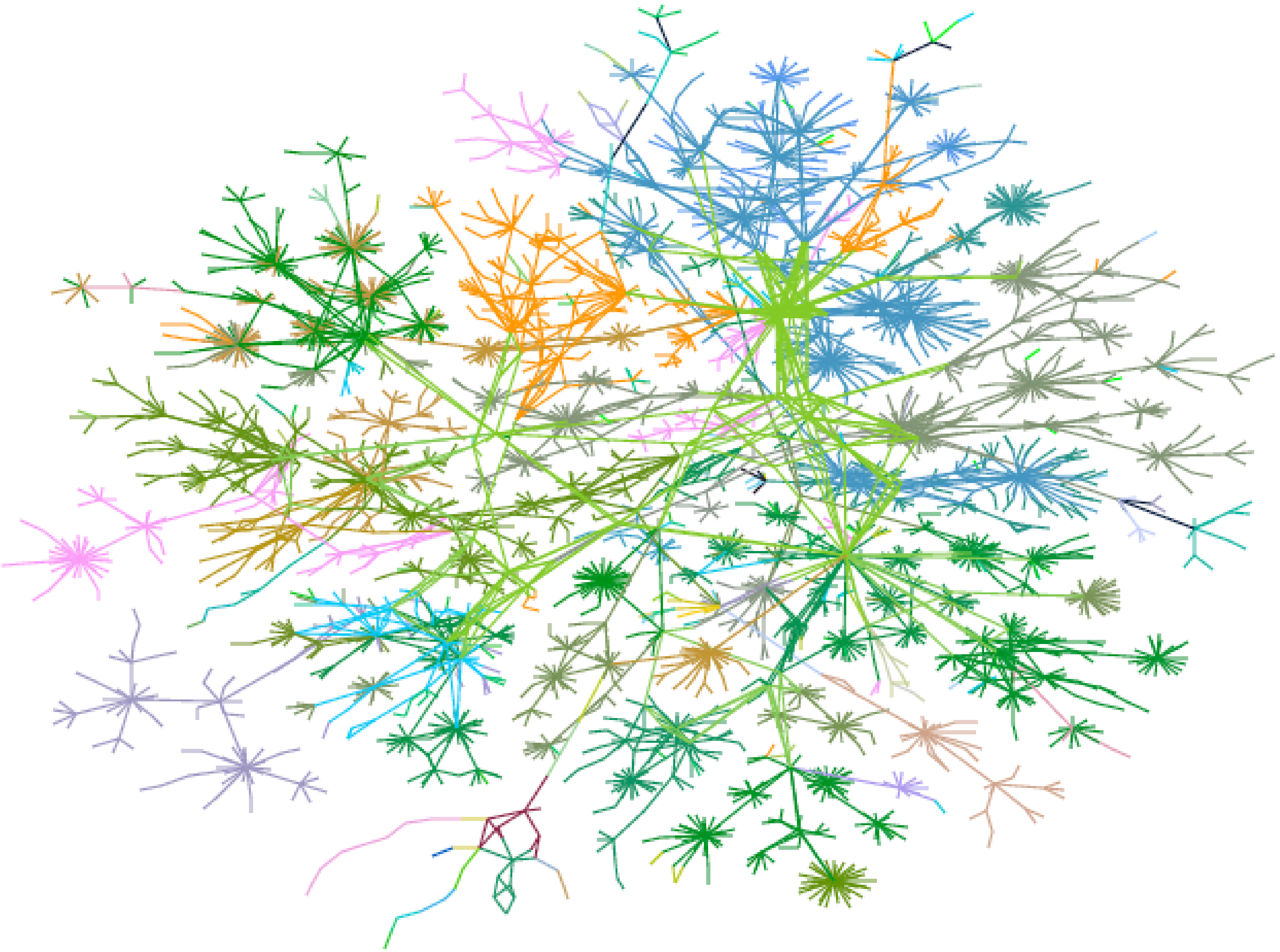
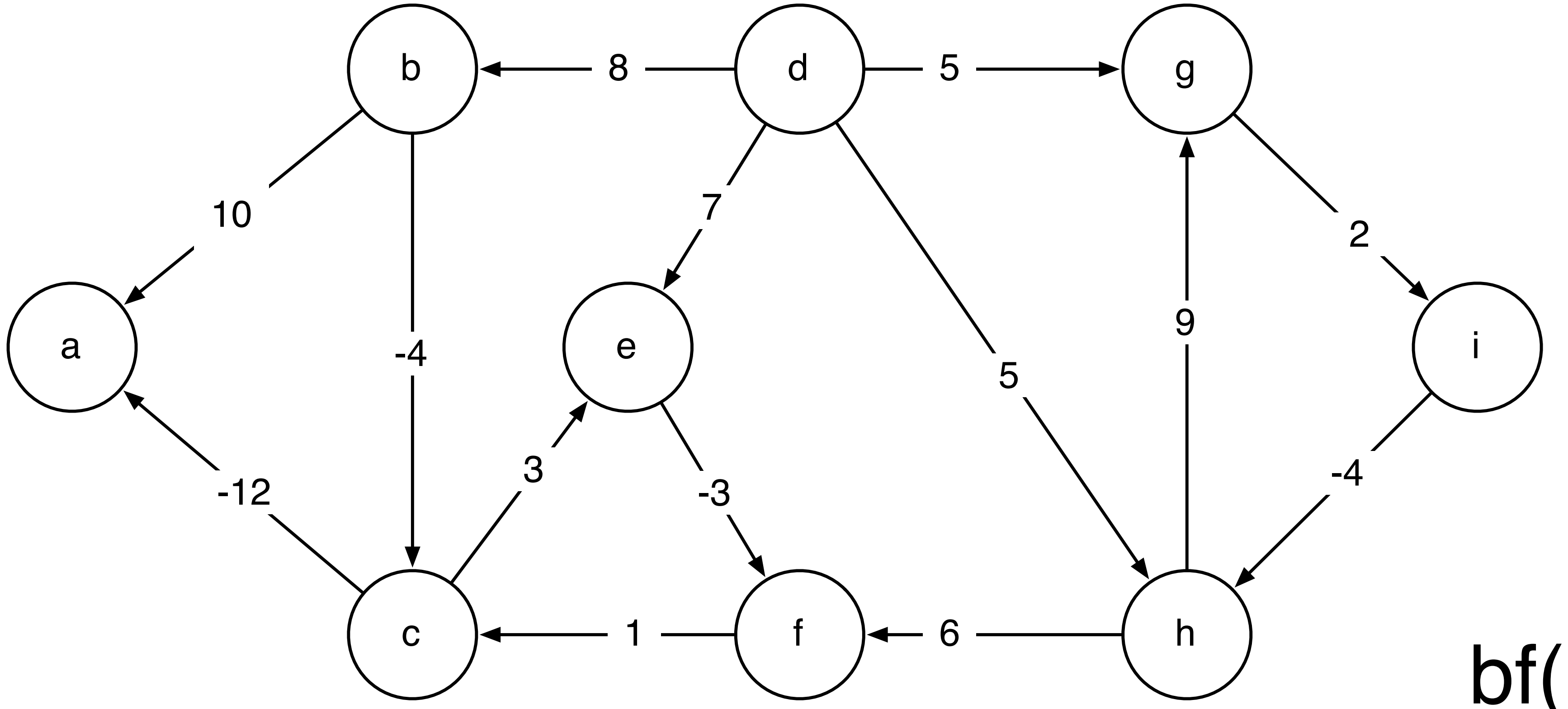


image: cheswick et al

Figure 3: Lucent's intranet as of 1 October 1999.

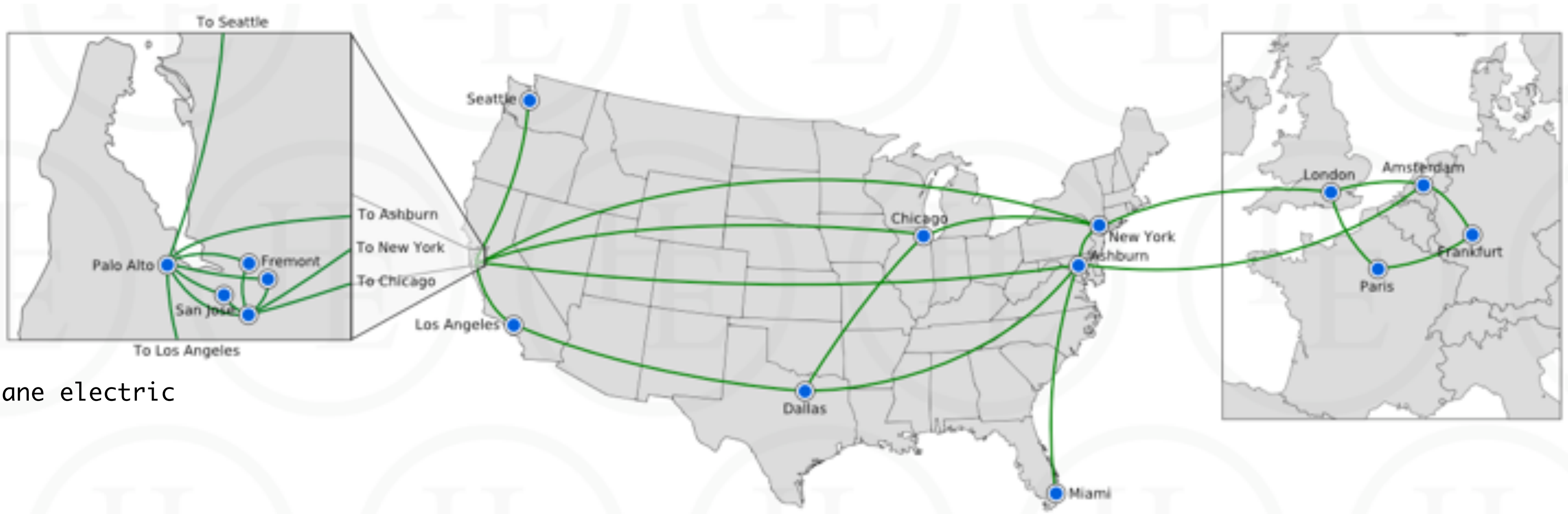


bf(G,d)

	0	1	2	3	4	5	6	7
a	∞							
b	∞	8						
c	∞							
d	0	0						
e	∞	7						
f	∞							
g	∞	5						
h	∞	5						
i	∞							

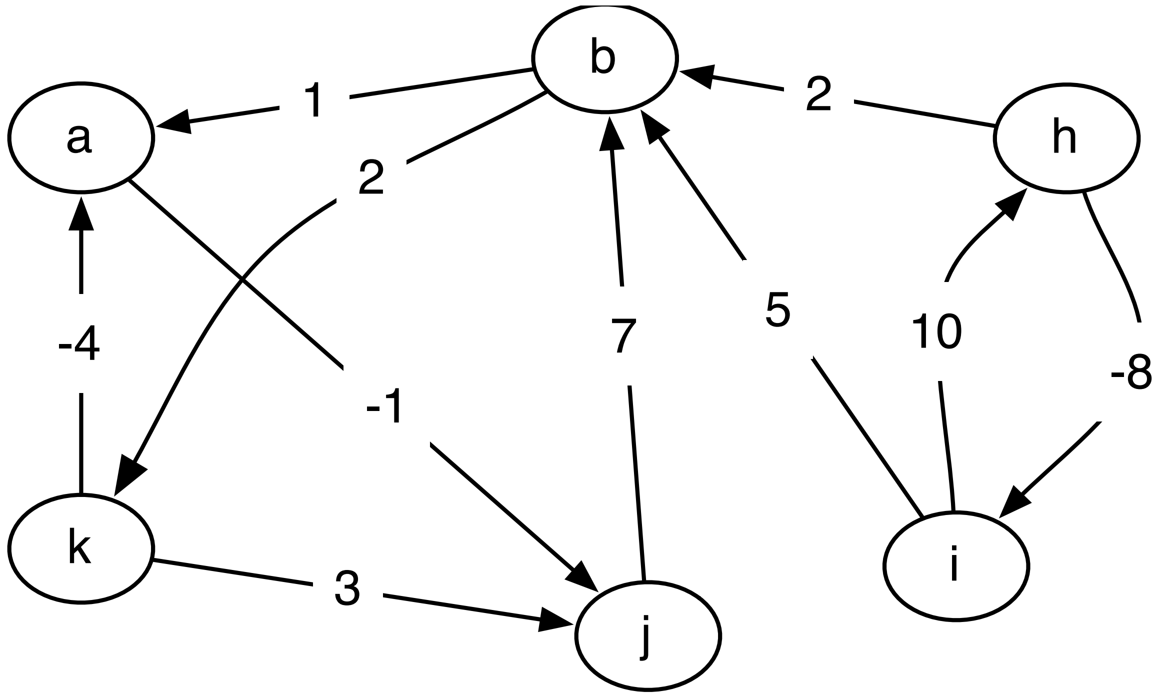
what happens when B changes...

distance vector



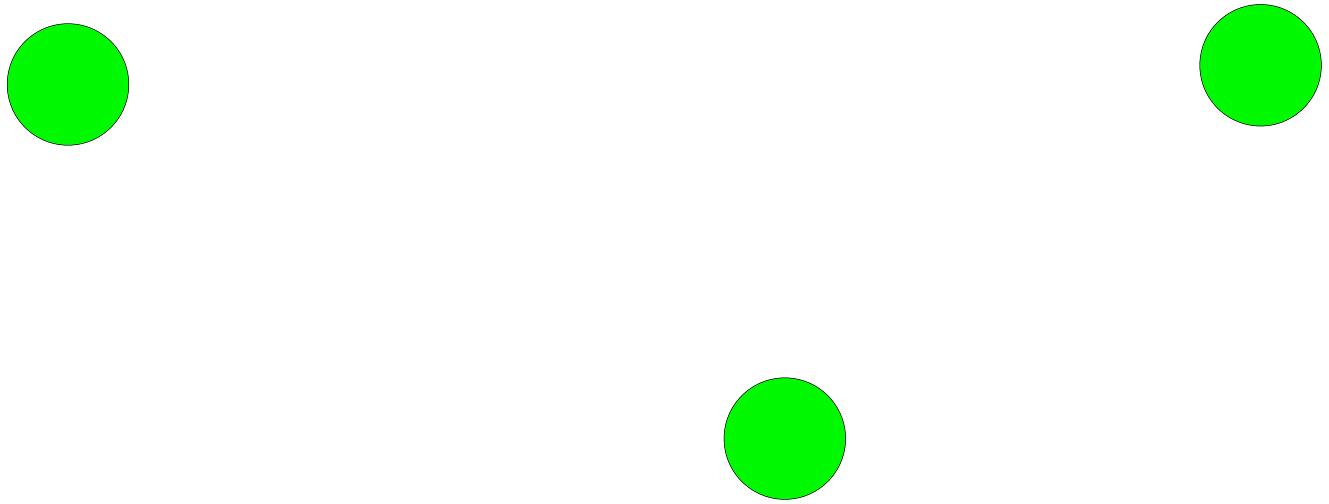
`image: hurricane electric`

all-pairs shortest path



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

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



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

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

floyd-warshall(G,W)