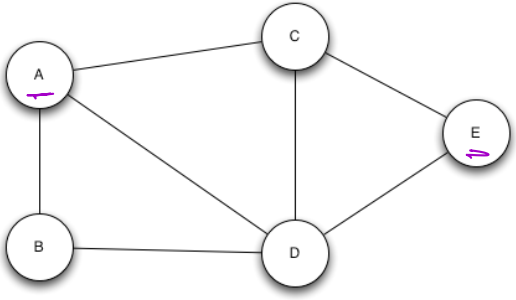
5800

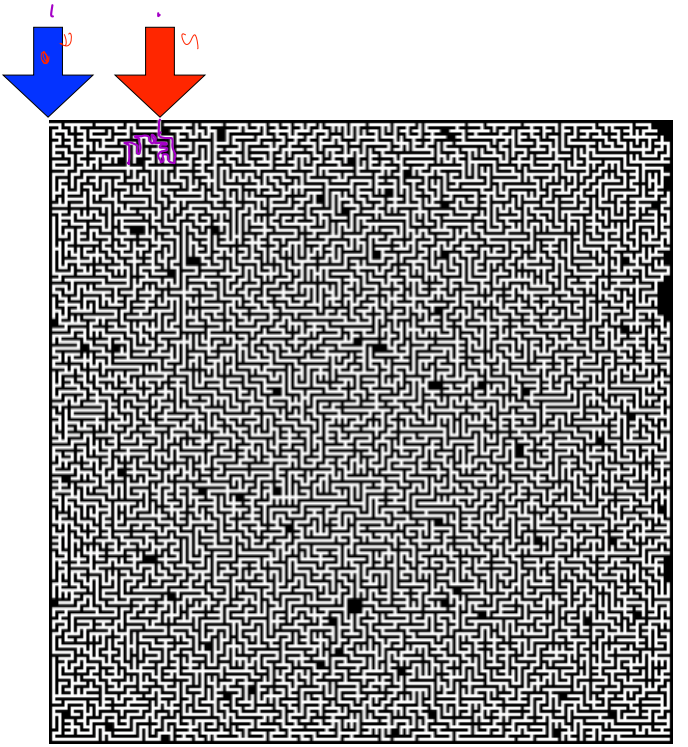
Shortest paths

mar 8/10 2022
shelat

simple graph questions



WHAT IS THE LENGTH OF THE PATH FROM A TO E?



shortest path property

DEFINITION:

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

delta

↑ sum the weights of the edges

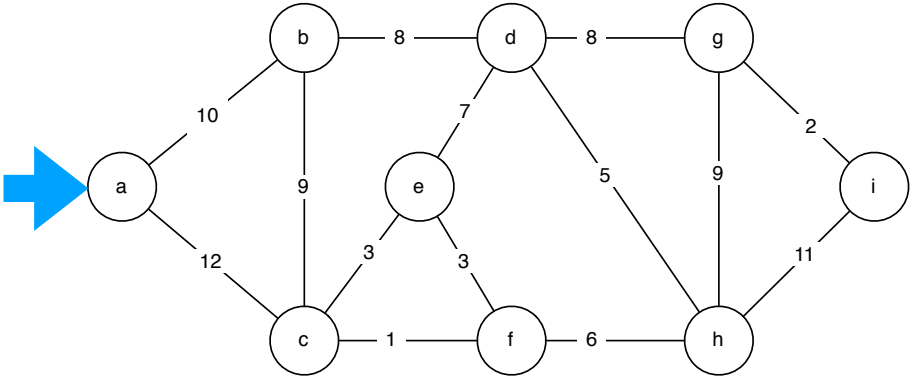
∞ if there is no path from $s \rightsquigarrow v$.

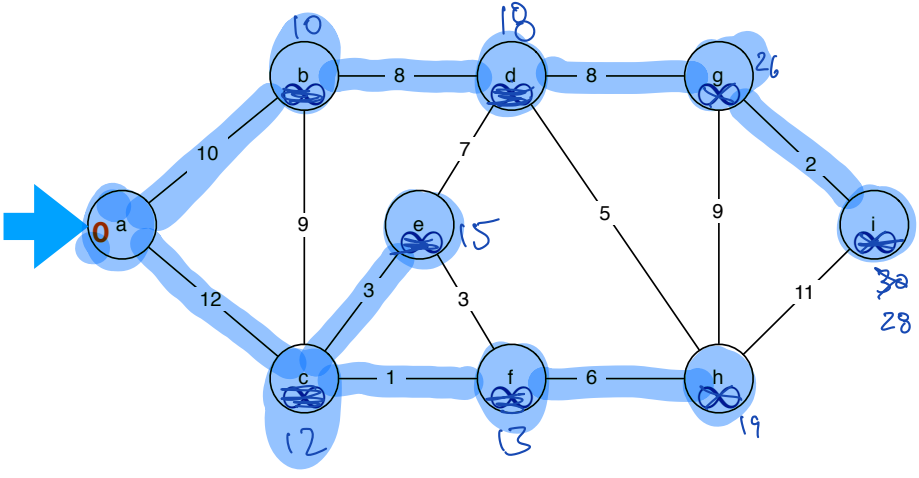
shortest path property

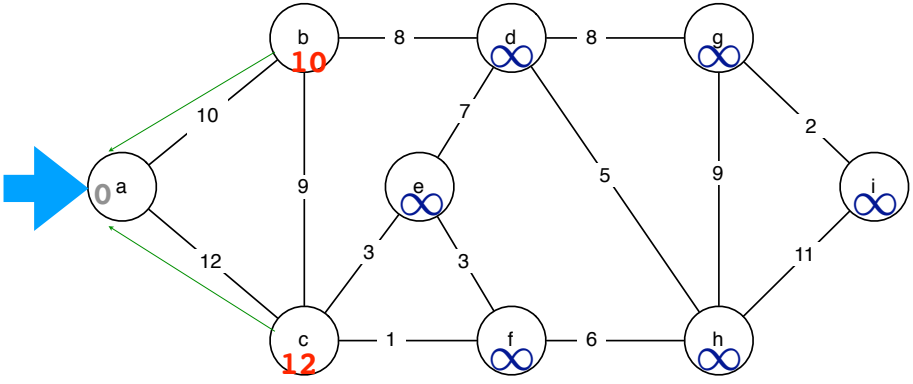
DEFINITION:

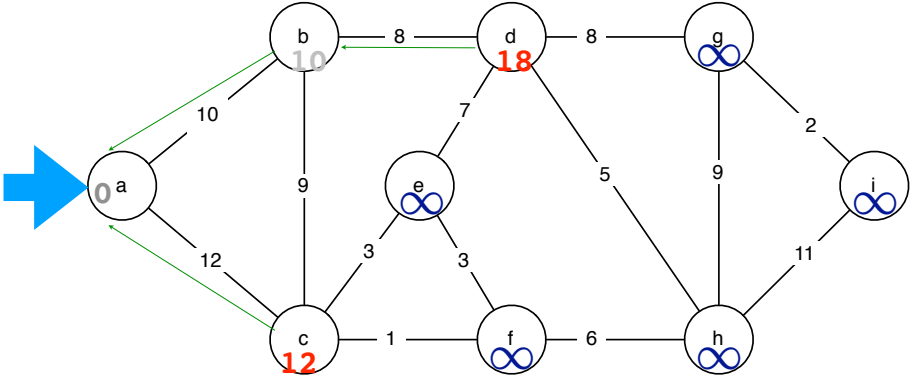
$\delta(s, v)$ Length of the shortest path from s to v, set to ∞ if there is no path from s to v

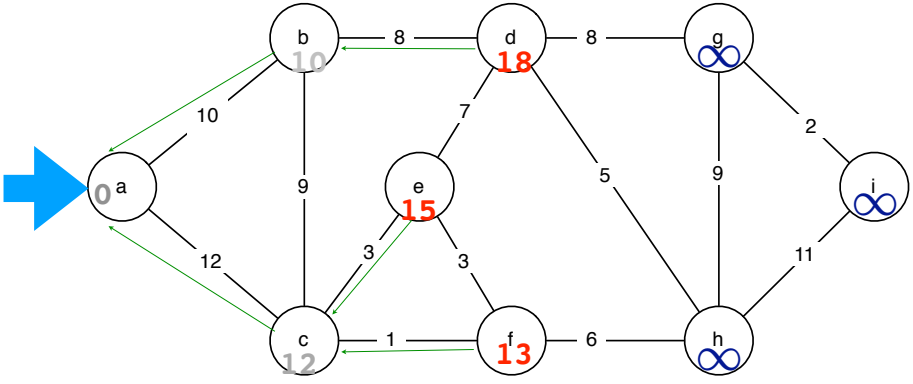
shortest paths from a

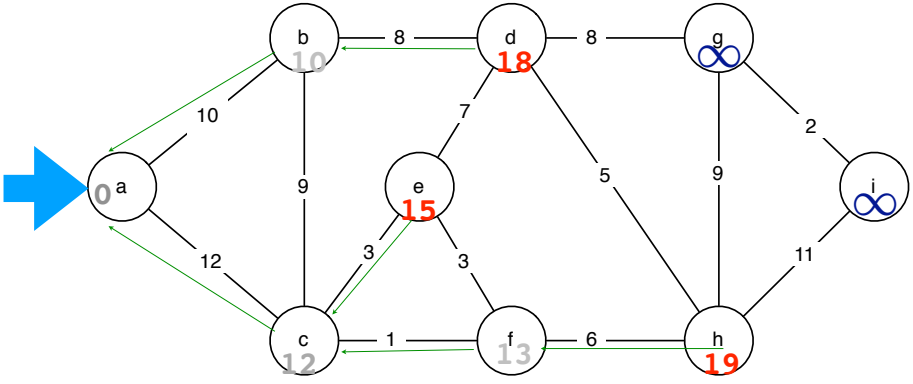


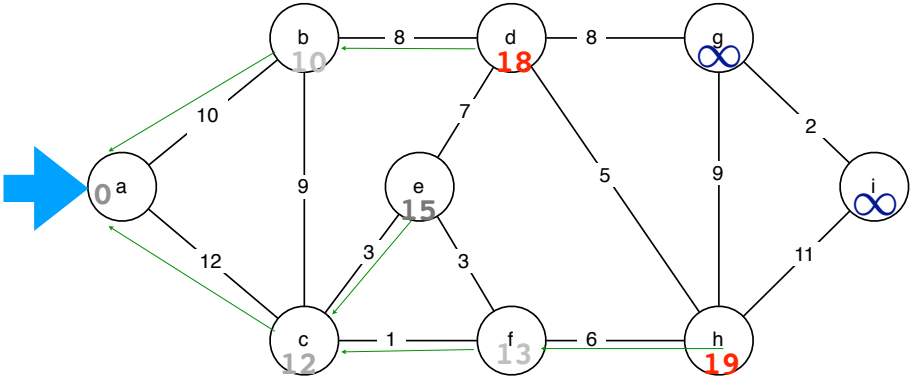

a
b
c
d
e
f
g
h
i
0
10
18
26
15
12
13
19
28
~~30~~
10
12
8
8
7
9
5
9
2
3
3
1
6
11

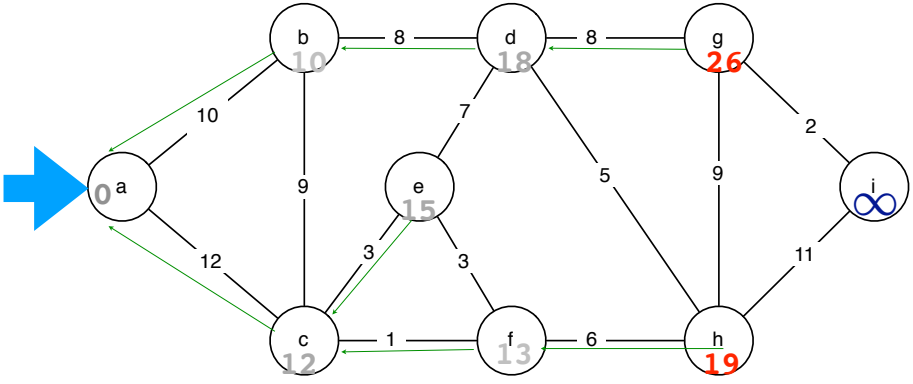

a
0
b
10
c
12
d
∞
e
∞
f
∞
g
∞
h
∞
i
∞
10
12
9
8
8
7
3
3
1
5
6
9
2
11

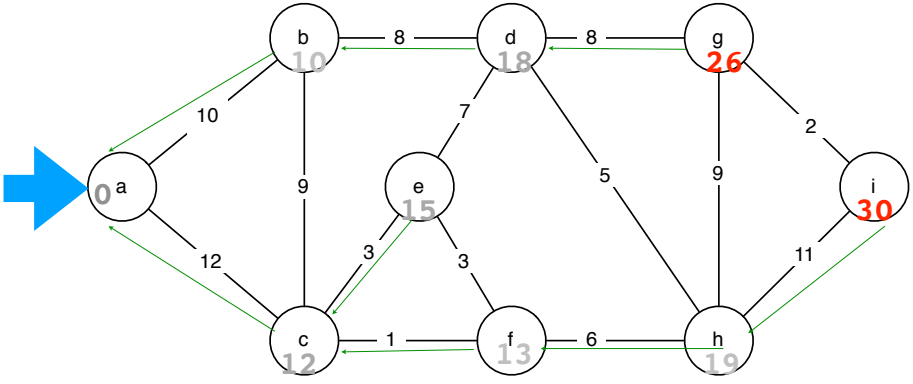

a
0
b
10
c
12
d
18
e
∞
f
∞
g
∞
h
∞
i
∞
10
12
9
8
8
7
3
3
1
5
6
9
2
11

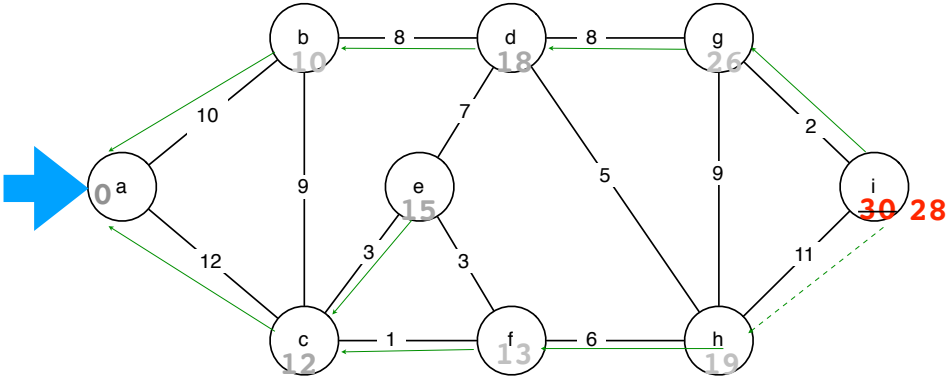

a
0
b
10
c
12
d
18
e
15
f
13
g
∞
h
∞
i
∞
10
8
8
7
2
9
5
9
12
3
3
11
1
6


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
∞
10
12
9
8
8
7
3
3
1
5
6
9
2
11


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


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

Algorithm

Dijkstra(G, s):

1. Initialize keys for $v \in V$ $d_v = \infty$ $d_s = 0$.
2. Add all nodes to a priority queue Q.
3. while the Q is not empty

    $u \leftarrow$ ExtractMin(Q)

    for each neighbor $v \in V$ of u that is in Q

        if $d_v > d_u + w(u,v)$

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

            $\pi_v \leftarrow (u,v)$

            DecreaseKey(v, d_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)    where v ∈ Q:
                 do if d_v > d_u + w(u, v)
                       then d_v ← d_u + w(u, v)
                            π_v ← u
                            DECREASEKEY(Q, v, d_v)
```

Very similar structure

```
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, d_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
```

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)   and v ∈ Q
                do if d_v > d_u + w(u, v)
                      then d_v ← d_u + w(u, v)
                           π_v ← u
                           DECREASEKEY(Q, v)   ~ dominant term.
```

once per edge at most.

The running time is $\Theta(E \log V)$ because each DecreaseKey operation is called at most once on each edge.

why does Dijkstra work?

one particular path to v.

TRIANGLE INEQUALITY:

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

from the definition of shortest paths.

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

d_v begins with the value ∞. So this condition holds at the start.

We only update d_v in line 10 when we have determined that $s \rightsquigarrow u + w(u,v)$ is less than the current d_v

Theorem

Given any weighted directed graph $G = (V, E)$ with non-negative weights, and a source s, Dijkstra(G, s) terminates with $d_v = \delta(s, v)$ for all $v \in V$.

Proof: Let S be the set nodes not in Q.

At line 5, S is empty.

Property: for all nodes $u \in S$, $d_u = \delta(s, u)$.

At line 5, this property holds.

Suppose this property holds after i iterations of loop 6

Theorem

Given any weighted directed graph $G = (V, E)$ with non-negative weights, and a source e, Dijkstra(G, s) terminates with $d_u = \delta(s, v)$ for all $v \in V$.

Let S be the set of elements not in Q.
At the beginning, this set is empty.

Theorem

Given any weighted directed graph $G = (V, E)$ with non-negative weights, and a source e, Dijkstra(G, s) terminates with $d_u = \delta(s, v)$ for all $v \in V$.

Let S be the set of elements not in Q.
At the beginning, this set is empty.

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

Proof (cont)

Suppose this property holds for the first i iterations of the loop.

$$w(p) = w(s,x) + w(x,y) + w(y,u)$$

$$\geq \delta(s,x) + w(x,y) + w(y,u)$$

$$= d_x + w(x,y) + w(y,u)$$

because $x \in S$ and our property therefore states $\delta(s,x) = d_x$

$$\geq d_y + w(y,u)$$

because of lines 10-12

But we also know that $d_y \geq d_u$.

Because we ran extractmin, and received u

$$\Rightarrow w(p) \geq d_u + w(y,u)$$

Because all edges have ≥ 0 weight, $w(y,u) \geq 0$

$$w(p) \geq d_u \Rightarrow d_u = \delta(s,u)$$

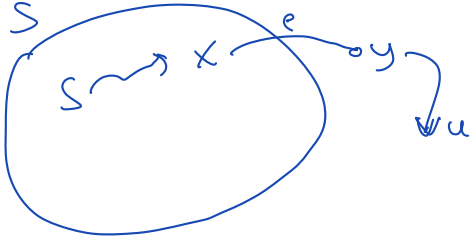
Proof (cont)

Suppose this property holds for the first i iterations of the loop.

Let u be the node extracted in line 7. We know that d_u was set as $d_u = d_z + w(z,u)$ for some node $z \in S$ (or $z = s$ and this is the first iteration which holds automatically).

Consider any path P from s to u. (immediately before line 7).

Let $e = (x,y)$ be the first edge along P to cross the set S.



$$w(P) = w(s,x) + w(x,y) + w(y,u)$$
$$\geq \delta(s,x) + w(x,y) + w(y,u)$$

Proof (cont) Suppose this property holds for the first i iterations of the loop.

Let u be the node extracted on line 7. By lines, 9,10,11, it follows that $d_u = d_z + w(z, u)$ for some node $z \in S$. By the hypothesis, $d_u = \delta(s, z) + w(z, u)$.

Consider any path p from s to u. Let $e = (x, y)$ be the first edge on path p that crosses cut $(S, V - S)$. By lines 9,10,11 and the inductive hypothesis, $d_y \leq \delta(s, x) + w(x, y)$. We now analyze the weight of path p:

$$\begin{aligned} w(p) &= w(s \rightsquigarrow x) + w(x, y) + w(y \rightsquigarrow u) \\ &\geq \delta(s, x) + w(x, y) + \delta(y, u) \end{aligned}$$

Substituting from above, we have that

$$w(p) \geq d_y + \delta(y, u)$$

However, by line 6, $d_u \leq d_y$, and so

$$w(p) \geq d_u + \delta(y, u)$$

Since all edges have non-negative weight, $\delta(y, u) \geq 0$ and so

$$w(p) \geq d_u$$

which implies that $d_u = \delta(s, u)$.

breadth first search

INPUT: $G = (V, E), s$ (All edge weights are 1)

OUTPUT:

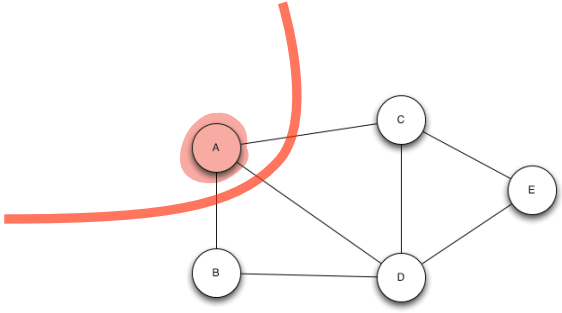
breadth first search

INPUT: $G = (V, E), s$ (All edge weights are 1)

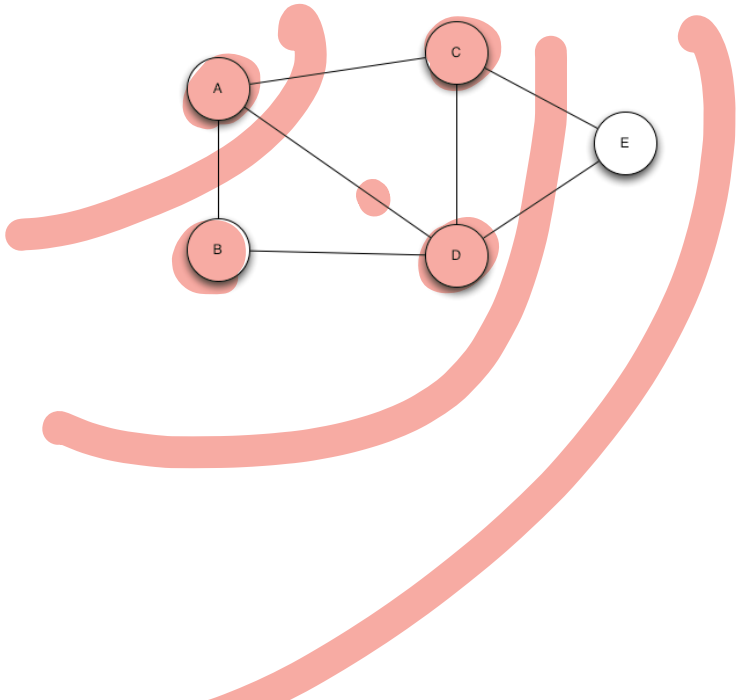
OUTPUT: $\forall v \in V \quad d_v = \delta(s, v)$

SMALLEST # OF EDGES FROM S TO V

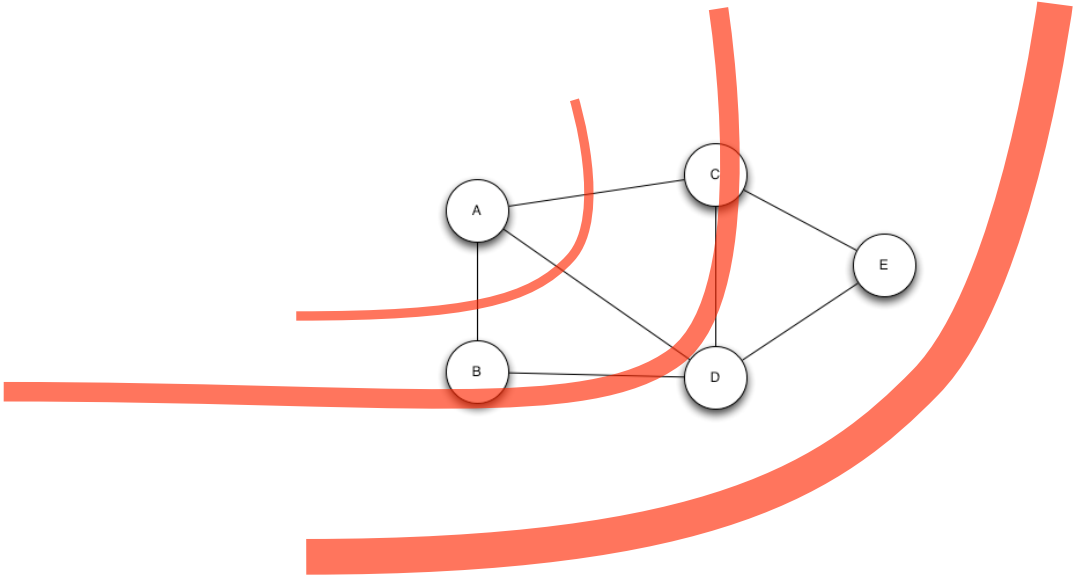
breadth-first search



breadth-first search



breadth-first search



breadth first search

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

OUTPUT: $d_v \quad \forall v \in V$

SMALLEST # OF EDGES FROM S TO V

breadth first search

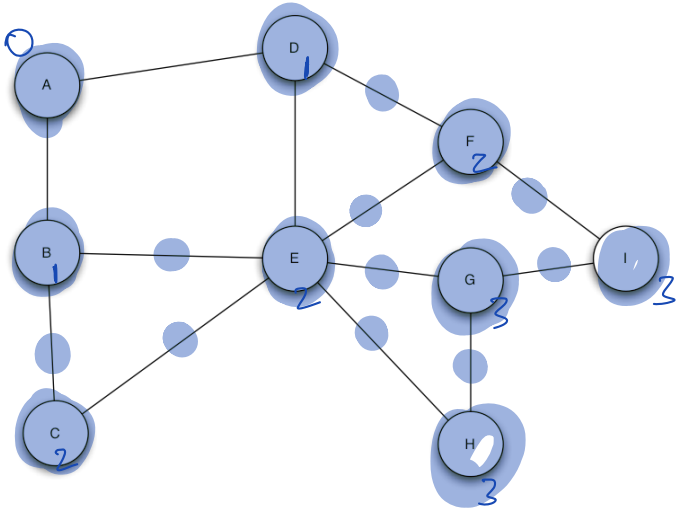
INPUT: $G = (V, E), s$

OUTPUT: $d_v \quad \forall v \in V$

SMALLEST # OF EDGES FROM S TO V

Big Idea: like Dijkstra, but use a simpler data structure: Queue.

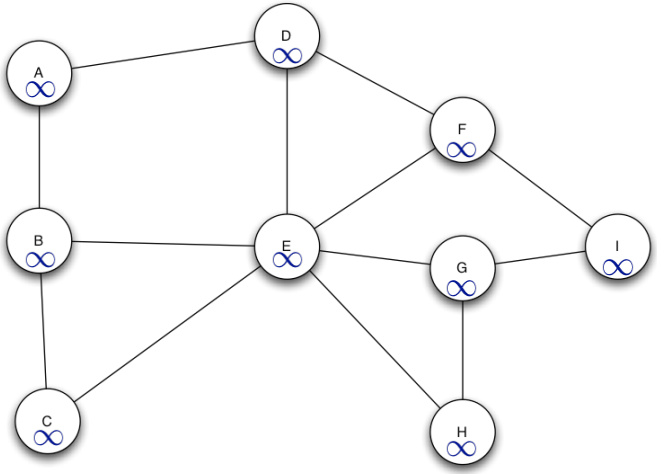
bfs(G, a)



Q

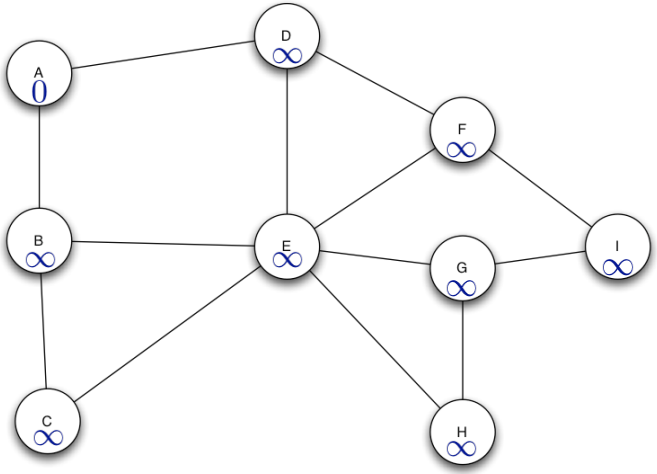
~~A~~
~~B~~
~~D~~
~~C~~
~~E~~
~~F~~
~~G~~
~~H~~
~~I~~

bfs(G, a)



Q

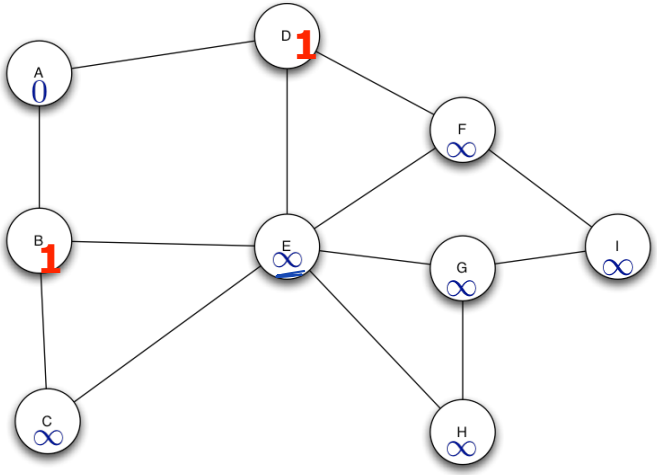
bfs(G, a)



Q

A

bfs(G, a)



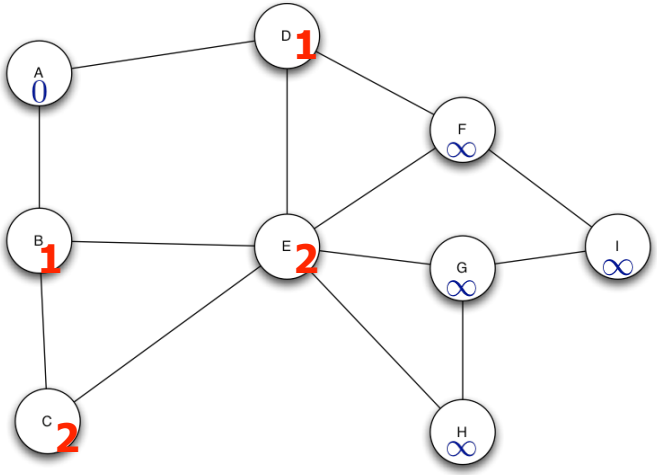
Q

~~A~~

B

D

bfs(G, a)



Q

~~A~~

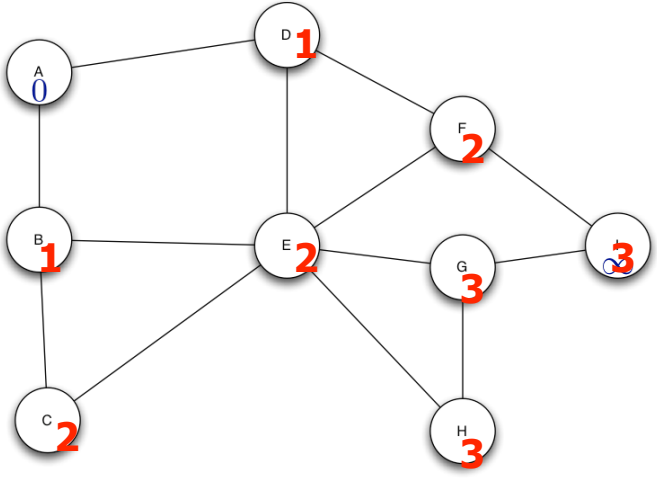
~~B~~

D

C

E

bfs(G, a)



Q

~~A~~

~~B~~

~~D~~

~~C~~

~~E~~

~~F~~

~~G~~

~~H~~

~~I~~

breadth first search

$\Theta(E+V)$

```
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)
```

$\Theta(V)$

$\Theta(1)$

total $\Theta(V)$

total $\Theta(E)$

$\Theta(V)$

BFS theorem

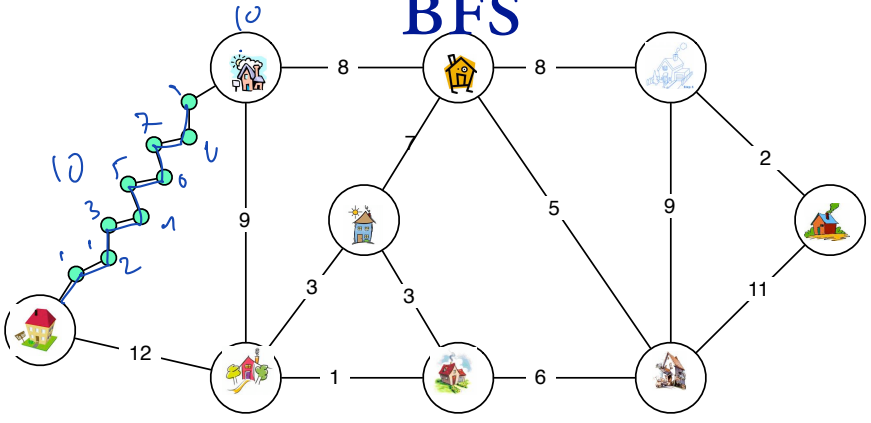
BFS theorem

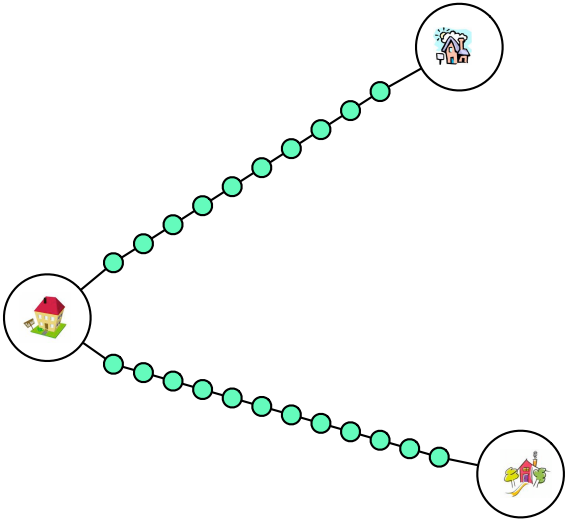
Thm: Let G be a graph and s a vertex. After $BFS(G, s)$ runs, then $d_v = \delta(s, v)$ for all nodes in the graph.

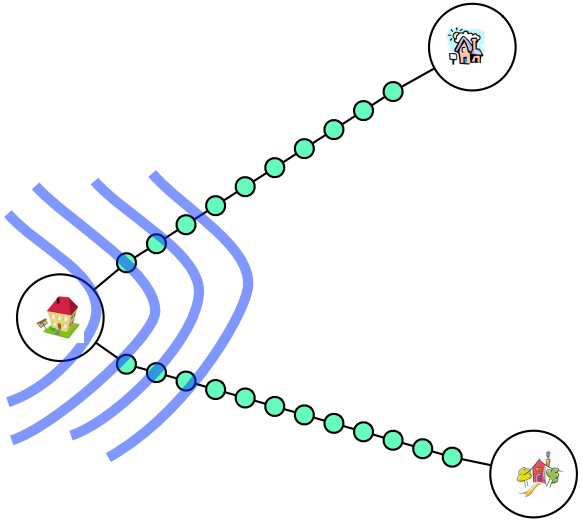
```
BFS(V, E, s)
for each u    V - {s}
    do d[u]    ∞
d[s]    0
Q
ENQUEUE(Q, s)
while Q =
    do u    DEQUEUE(Q)
        for each v    Adj[u]
            do if d[v] = ∞
                then d[v]    d[u] + 1
                    ENQUEUE(Q, v)
```

```
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

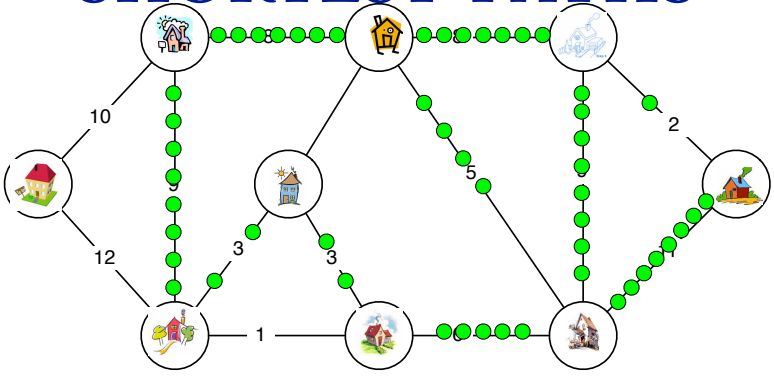


image:www.princegeorgeva.org, thefranciscofamily.org, www.rightdriveacademy.co.uk, www.ccscambridge.org, www.drawingcoach.com, www.pastoral.org.uk, www.daasgallery.com

What about Negative edge weights?

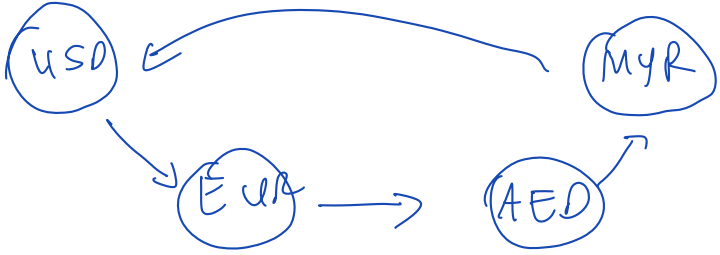
Mid-market rates as of **2016-03-31 17:40 UTC**

Currency code ▲▼	Currency name ▲▼	Units per EUR	EUR per Unit
USD	US Dollar	1.1386632306	0.8782227907
EUR	Euro	1.0000000000	1.0000000000
GBP	British Pound	0.7921136388	1.2624451227
INR	Indian Rupee	75.3658843112	0.0132686030
AUD	Australian Dollar	1.4859561878	0.6729673514
CAD	Canadian Dollar	1.4796754127	0.6758238945
SGD	Singapore Dollar	1.5347639238	0.6515660060
CHF	Swiss Franc	1.0917416715	0.9159676012
MYR	Malaysian Ringgit	4.4140052400	0.2265516114
JPY	Japanese Yen	128.1388820287	0.0078040325
CNY	Chinese Yuan Renminbi	7.3411003512	0.1362193612
NZD	New Zealand Dollar	1.6484648003	0.6066250246
THB	Thai Baht	39.9627318192	0.0250233143
HUF	Hungarian Forint	313.9042436792	0.0031856849
AED	Emirati Dirham	4.1823100458	0.2391023117

XE Currency Table: AED - Emirati Dirham

Mid-market rates as of **2016-03-31 17:39 UTC**

Currency code ▲▼	Currency name ▲▼	Units per AED	AED per Unit
USD	US Dollar	0.2722570106	3.6730000000
EUR	Euro	0.2391289974	4.1818433177
GBP	British Pound	0.1893997890	5.2798369266
INR	Indian Rupee	18.0207422309	0.0554916100
AUD	Australian Dollar	0.3552996418	2.8145257760
CAD	Canadian Dollar	0.3538334124	2.8261887234
SGD	Singapore Dollar	0.3669652245	2.7250538559
CHF	Swiss Franc	0.2610686193	3.8304105746
MYR	Malaysian Ringgit	1.0548325619	0.9480177576
JPY	Japanese Yen	30.6399242607	0.0326371564
CNY	Chinese Yuan Renminbi	1.7555154332	0.5696332719
NZD	New Zealand Dollar	0.3941937299	2.5368237088
THB	Thai Baht	9.5553789460	0.1046530970
HUF	Hungarian Forint	75.0637936939	0.0133220019
AED	Emirati Dirham	1.0000000000	1.0000000000



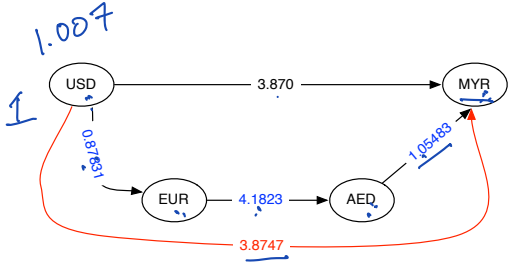
Mid-market rates as of **2016-03-31 17:40 UTC**

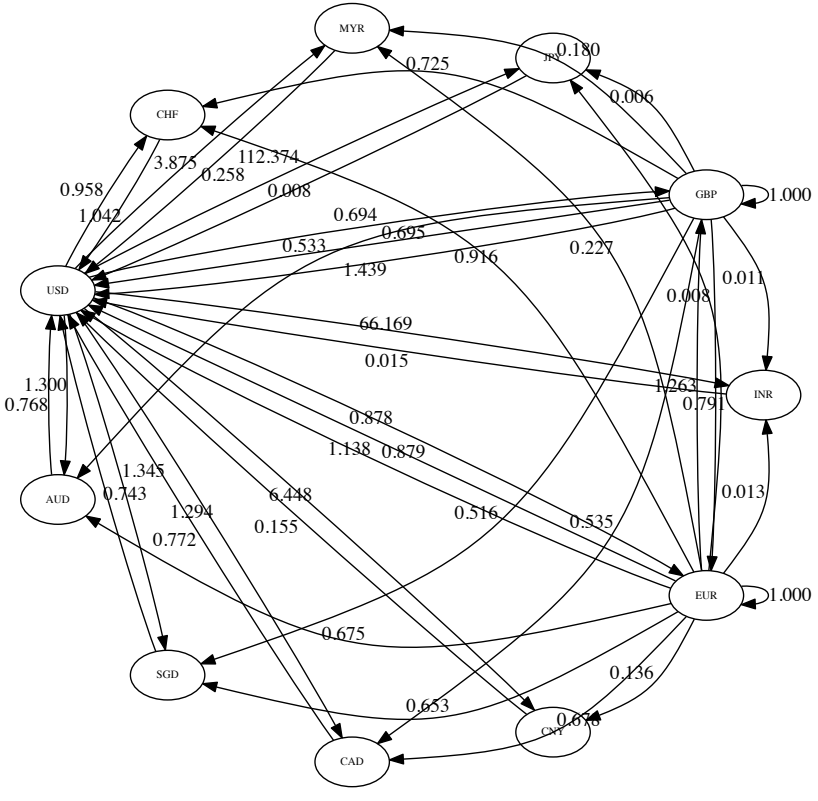
Currency code ▲▼	Currency name ▲▼	Units per EUR	EUR per Unit
USD	US Dollar	1.1386632306	0.8782227907
EUR	Euro	1.0000000000	1.0000000000
GBP	British Pound	0.7921136388	1.2624451227
INR	Indian Rupee	75.3658843112	0.0132686030
AUD	Australian Dollar	1.4859561878	0.6729673514
CAD	Canadian Dollar	1.4796754127	0.6758238945
SGD	Singapore Dollar	1.5347639238	0.6515660060
CHF	Swiss Franc	1.0917416715	0.9159676012
MYR	Malaysian Ringgit	4.4140052400	0.2265516114
JPY	Japanese Yen	128.1388820287	0.0078040325
CNY	Chinese Yuan Renminbi	7.3411003512	0.1362193612
NZD	New Zealand Dollar	1.6484648003	0.6066250246
THB	Thai Baht	39.9627318192	0.0250233143
HUF	Hungarian Forint	313.9042436792	0.0031856849
AED	Emirati Dirham	4.1823100458	0.2391023117

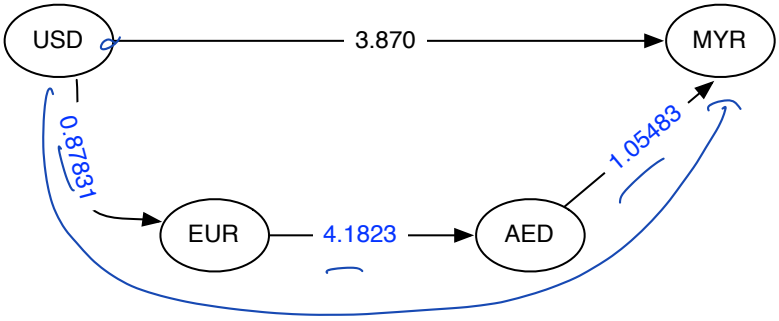
XE Currency Table: AED - Emirati Dirham

Mid-market rates as of **2016-03-31 17:39 UTC**

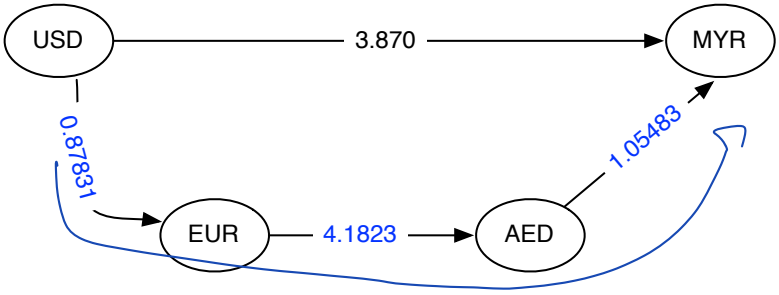
Currency code ▲▼	Currency name ▲▼	Units per AED	AED per Unit
USD	US Dollar	0.2722570106	3.6730000000
EUR	Euro	0.2391289974	4.1818433177
GBP	British Pound	0.1893997890	5.2798369266
INR	Indian Rupee	18.0207422309	0.0554916100
AUD	Australian Dollar	0.3552996418	2.8145257760
CAD	Canadian Dollar	0.3538334124	2.8261887234
SGD	Singapore Dollar	0.3669652245	2.7250538559
CHF	Swiss Franc	0.2610686193	3.8304105746
MYR	Malaysian Ringgit	1.0548325619	0.9480177576
JPY	Japanese Yen	30.6399242607	0.0326371564
CNY	Chinese Yuan Renminbi	1.7555154332	0.5696332719
NZD	New Zealand Dollar	0.3941937299	2.5368237088
THB	Thai Baht	9.5553789460	0.1046530970
HUF	Hungarian Forint	75.0637936939	0.0133220019
AED	Emirati Dirham	1.0000000000	1.0000000000



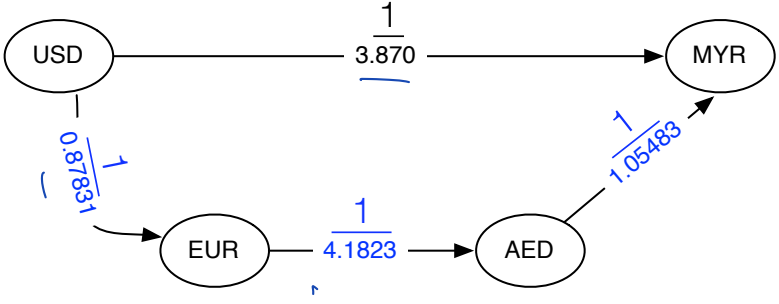

MYR
JPY
CHF
GBP
USD
INR
AUD
EUR
SGD
CNY
CAD
0.180
0.725
0.006
112.374
3.875
0.258
0.008
0.958
1.042
1.000
0.694
0.695
0.533
0.916
0.227
1.439
0.011
0.008
66.169
0.015
1.263
0.791
1.300
0.768
0.878
1.138
0.879
1.345
0.743
6.448
1.294
0.155
0.516
0.535
0.013
0.772
1.000
0.675
0.136
0.653
0.678



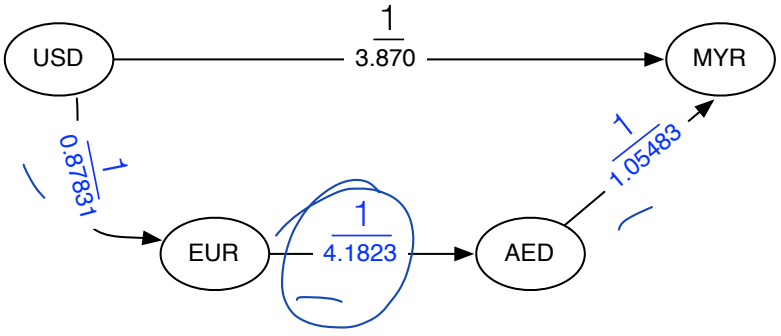
Trying to find Max weight (mult) Paths



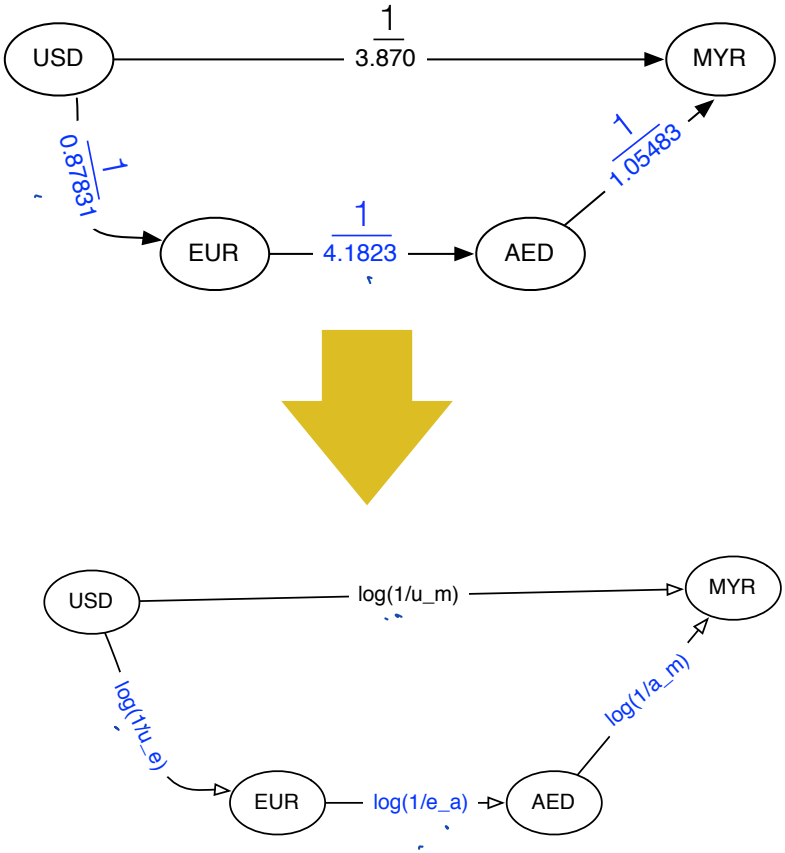
Trying to find Max weight (mult) Paths



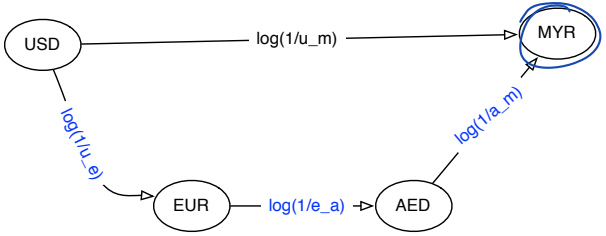
Trying to find MIN weight (mult) Paths



Trying to find MIN weight (mult) Paths



Trying to find MIN weight (mult) Paths



$$\log\left(\frac{1}{4.183}\right)$$

$$\log(0.23906)$$

$$= -0.6214$$

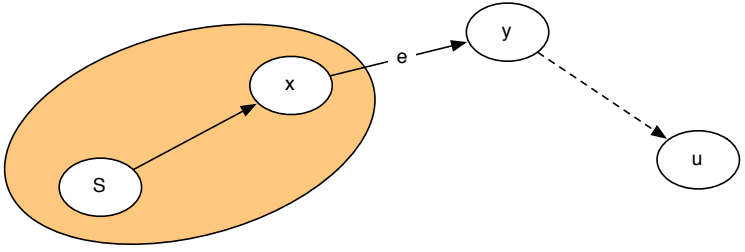
where does old argument break down

Recall this line in our proof of Dijkstra:

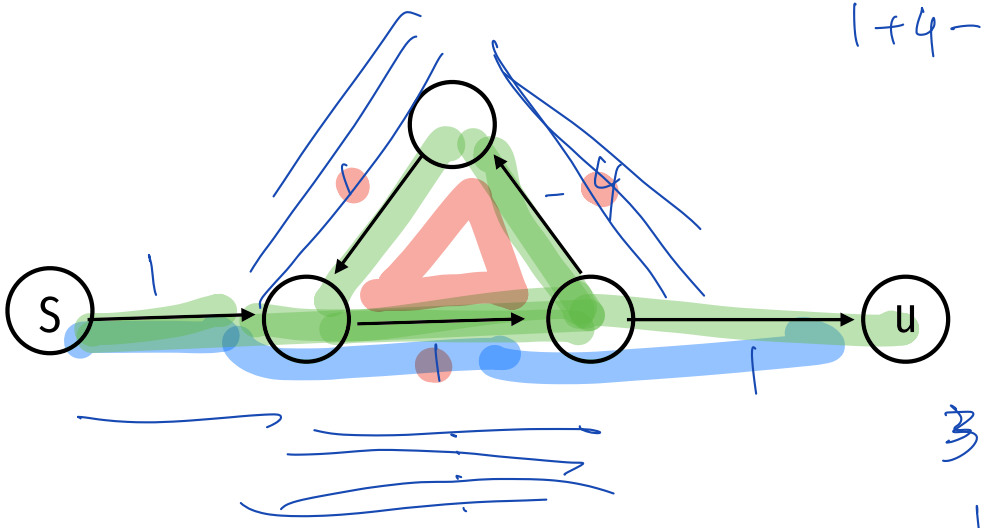
$$w(p) = d_y + \underline{w(y, u)}$$

$w(y,u) \geq 0$

where does old argument break down



2nd problem: cycles



$1+4-16+4+1$

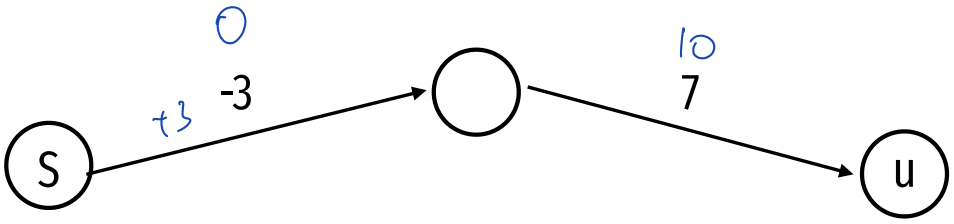
$= -6$

3

1

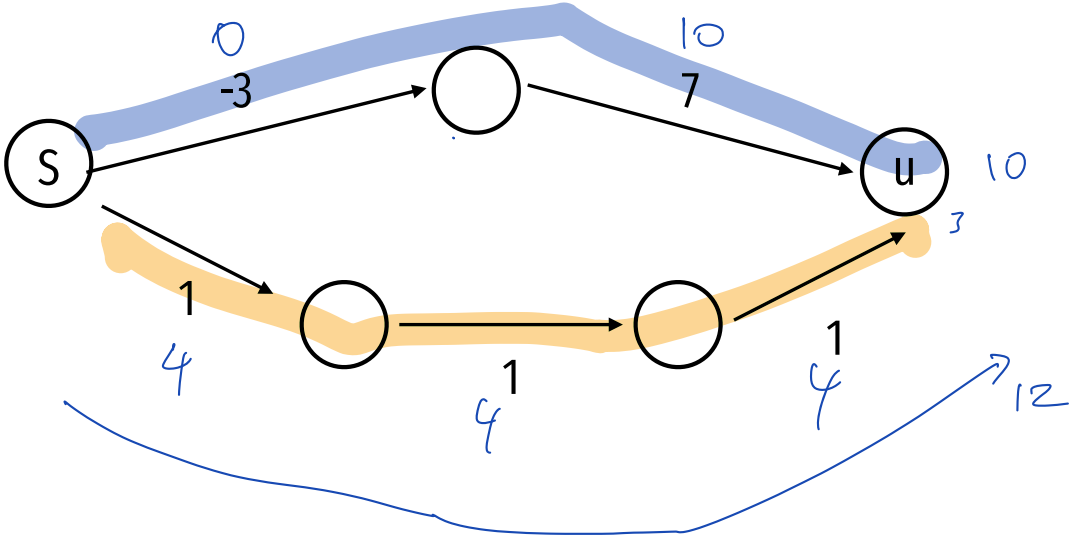
first ideas: Add to each edge

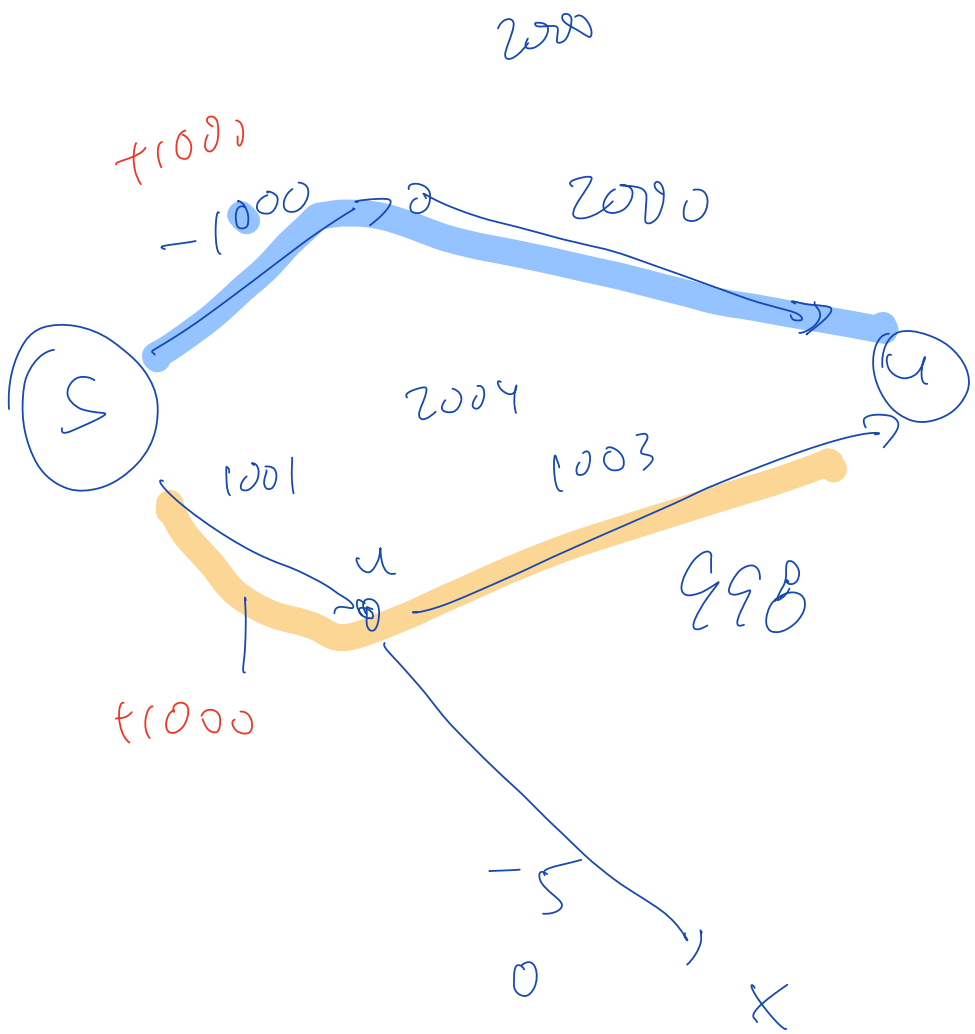
Idea: add a constant to each edge so that all weights are non-negative



first ideas: Add to each edge

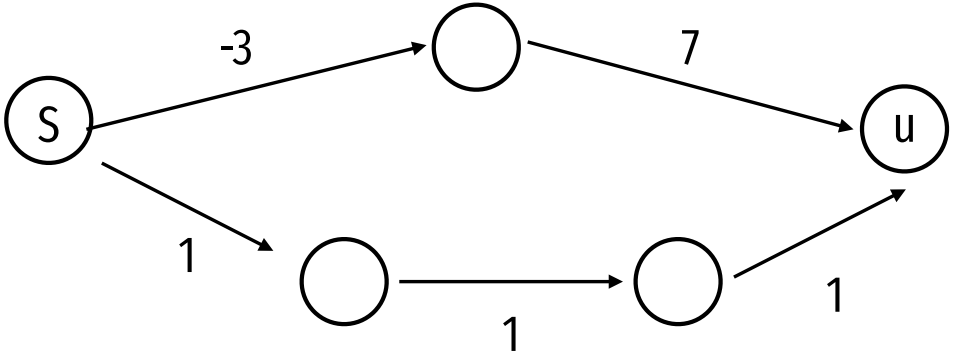
Idea: add a constant to each edge so that all weights are non-negative




2000
+1000
-1000
v
2000
s
2004
t
1001
1003
u
998
+1000
-5
0
x

first ideas: Add to each edge

Idea: add a constant to each edge so that all weights are non-negative



Problem: this can change the shortest paths of the graph.

sssp(G,s)

$$\textsc{short}_{i,v} =$$