Negative Cycle Detection

Algorithmique
Fall semester 2011/12

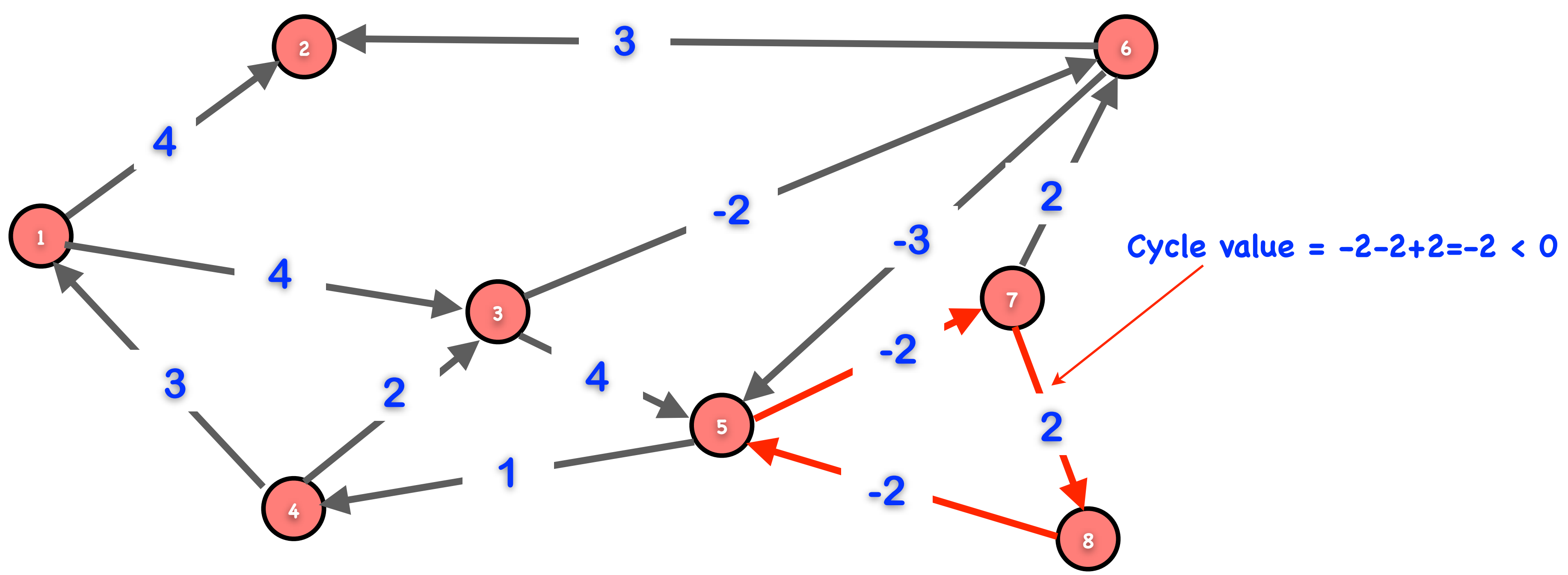
Recap

Given: Directed+connected graph G with edge weights w(e) on edges e.

Definition: A negative cycle in G is a cycle $v_0 - v_1 - \cdots - v_t - v_0$

in which $w(v_0, v_1) + w(v_1, v_2) + \cdots + w(v_t, v_0) < 0$

Goal: Determine whether the graph has a negative cycle.



Moore-Bellman-Ford Algorithm

Assumes: n nodes, set of edges E, set of nodes V, s in V, no negative cycles

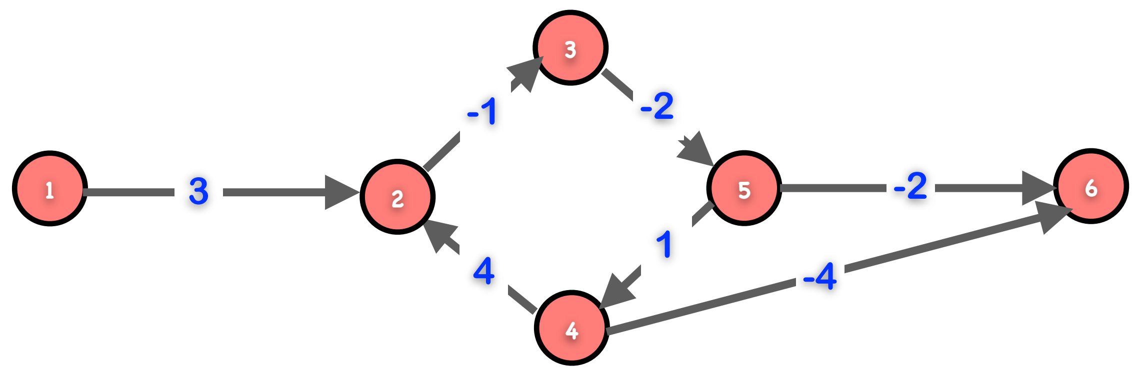
Finds: for all nodes v, the shortest path from s to v

How: for every node v, keep track of a value $l(v)$ and pred(v); $l(v)$ is the current estimate of the length of shortest path to v, pred(v) is the predecessor of v in this shortest path

```
(1)  dist(s) = 0,  dist(v) = infinity if v is not s, pred(v) = NULL for all v
(2) For i from 1 to n-1 do
    (A) For all edges (u,v) in E do
        (a) if ( l(u) + w(u,v) < l(v) ) then
            (i)  Set l(v) to l(u) + w(u,v)
            (ii) Set pred(v) to u
```

Invariant:
at iteration i, l(v) is the length of the shortest path from s to v using at most i edges.

Example

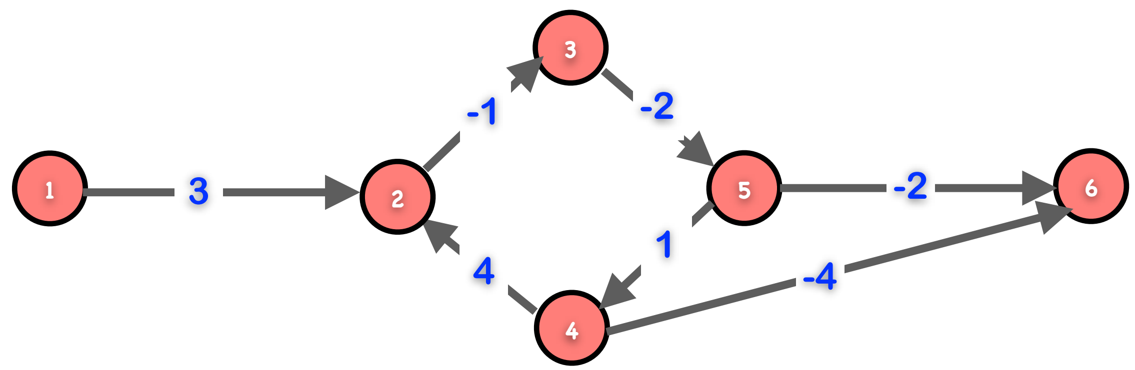


Nodes

Iterations	1	2	3	4	5	6	
0	0	inf	inf	inf	inf	inf	Initialization
1	0	3	inf	inf	inf	inf	
2	0	3	2	inf	inf	inf	
3	0	3	2	inf	0	inf	
4	0	3	2	1	0	-2	
5	0	3	2	1	0	-3	

=min(inf-4, 0-2)=-2

Example



Nodes

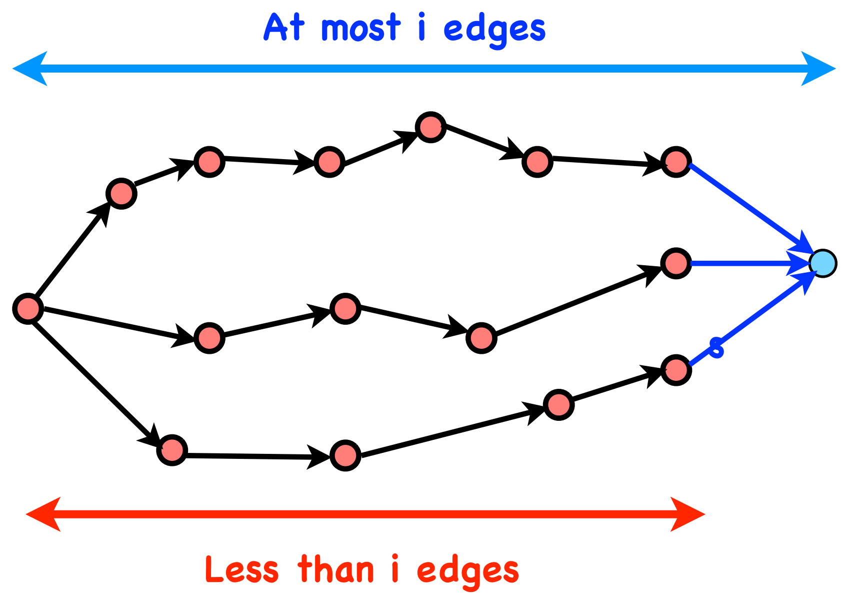
Iterations

	1	2	3	4	5	6	
0	0	inf	inf	inf	inf	inf	Initialization
1	0	3	inf	inf	inf	inf	
2	0	3	2	inf	inf	inf	
3	0	3	2	inf	0	inf	
4	0	3	2	1	0	-2	
5	0	3	2	1	0	-3	
6	0	3	2	1	0	-3	Stays the same in the next iteration

Why does it work?

Invariant:

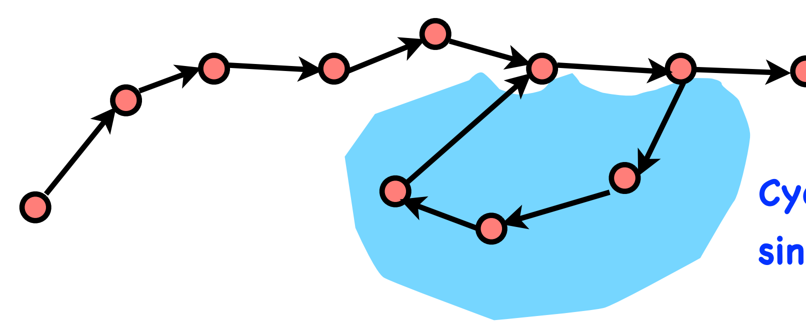
- $l(v)$ is the length of the shortest path from s to v using at most i edges in the i-th iteration.
- Proof by induction.



Negative cycles: If there are no negative cycles visible from s, then for any v there is a shortest path from s to v using at most n-1 edges.

A negative cycle visible from s is a negative cycle on a path from s to some other node v in the graph.

If there is a path with n or more edges, then there is a cycle, and it can be removed.



Visible Negative Cycles

If the *l*-value of at least one node changes in round *n* of the MBF algorithm, then there is a negative cycle that is visible from *s*.

This is because the contrapositive is true: if there are no negative cycles visible from *s*, then the *l*-values don't change in round *n*. See previous slide.

How about the converse?

If the *l*-values of the nodes don't change in round *n*, then there is no negative cycle visible from *s*.

In this case $\forall (u, v) \in E\colon l(u) + w(u, v) \geq l(v)$

So, for a cycle $v_0 - v_1 - \cdots - v_{t-1} - v_t = v_0$

$$\sum_{i=1}^{t} l(v_i) \leq \sum_{i=1}^{t} (l(v_{i-1}) + w(v_{i-1}, v_i)) = \sum_{i=1}^{t} l(v_{i-1}) + \sum_{i=1}^{t} w(v_{i-1}, v_i)$$

Equal!

$$\implies 0 \leq \sum_{i=1}^{t} w(v_{i-1}, v_i)$$

Cycle is not negative.

Visible Negative Cycles

(1) Apply the MBF algorithm to the graph

(2) For all edges in E do

  (a) if ($l(u) + w(u,v) < l(v)$) then

    (i) Output TRUE   Yes, there is a negative cycle

(3) Output FALSE   No negative cycles

Visible Negative Cycles

(1) Apply the MBF algorithm to the graph $O(|E||V|)$

(2) For all edges in E do $O(|E|)$

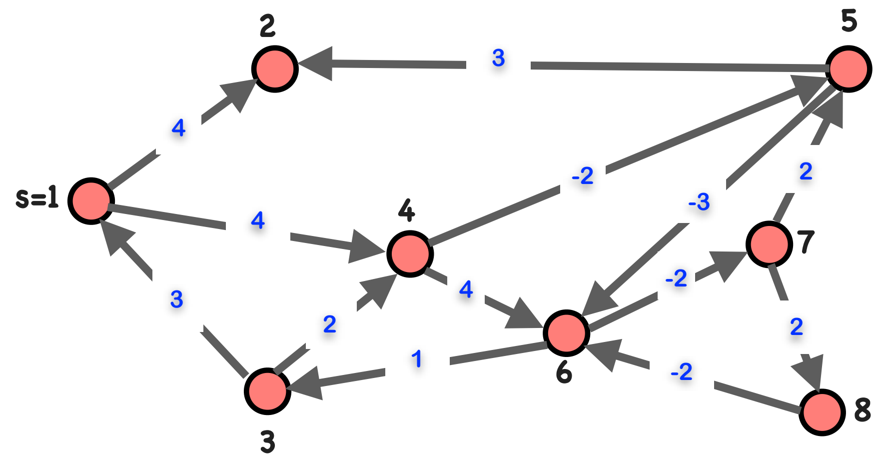
  (a) if ($l(u) + w(u,v) < l(v)$) then

    (i) Output TRUE

(3) Output FALSE

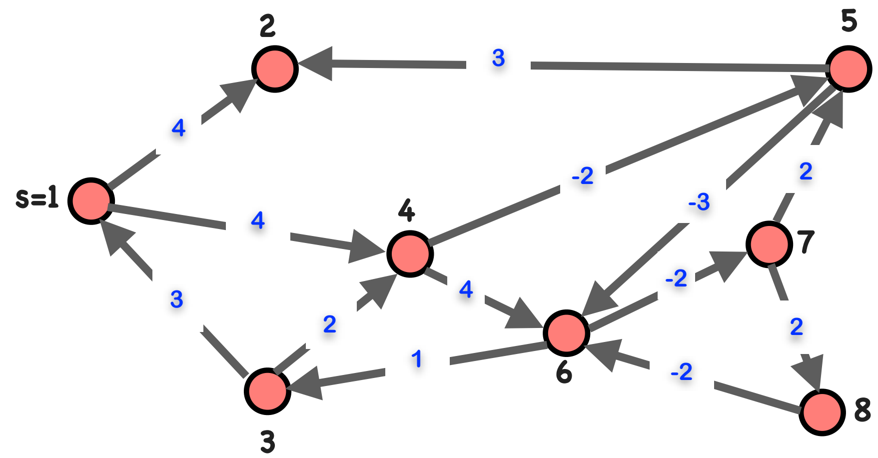
Running time is $O(|E||V|)$

Example



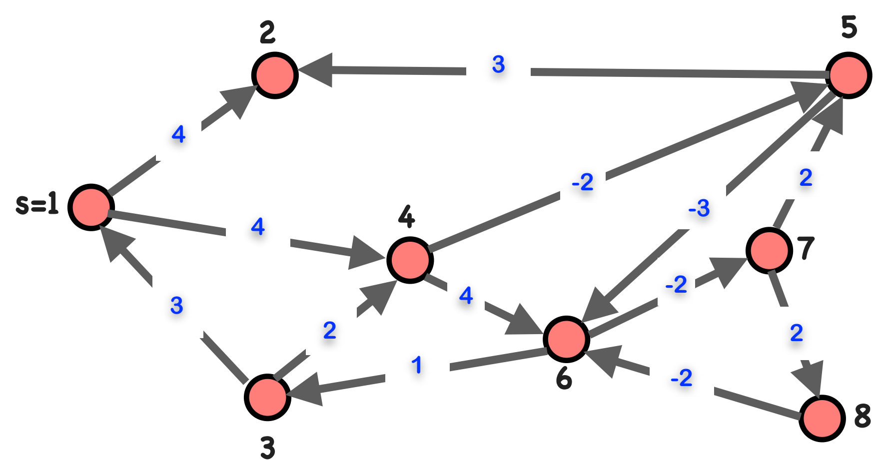
	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf

Example



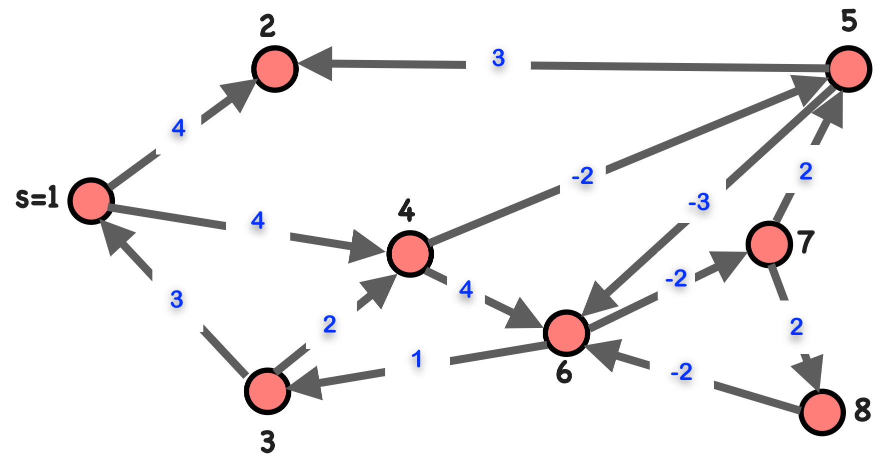
	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf

Example



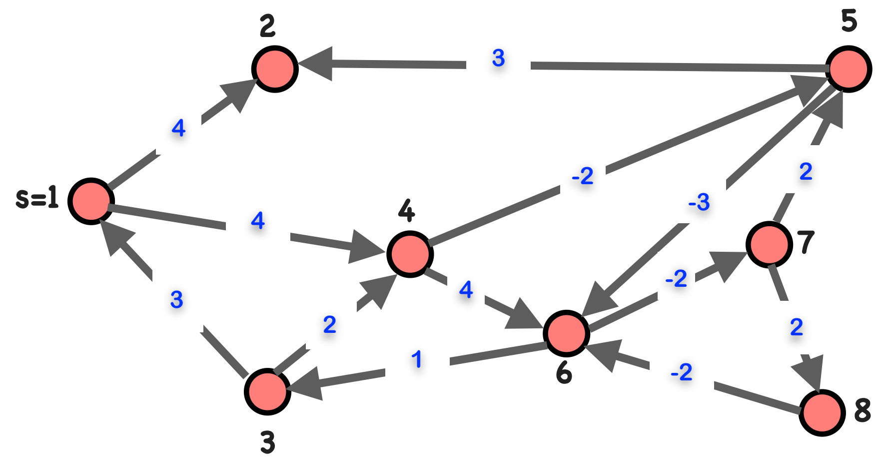
	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf

Example



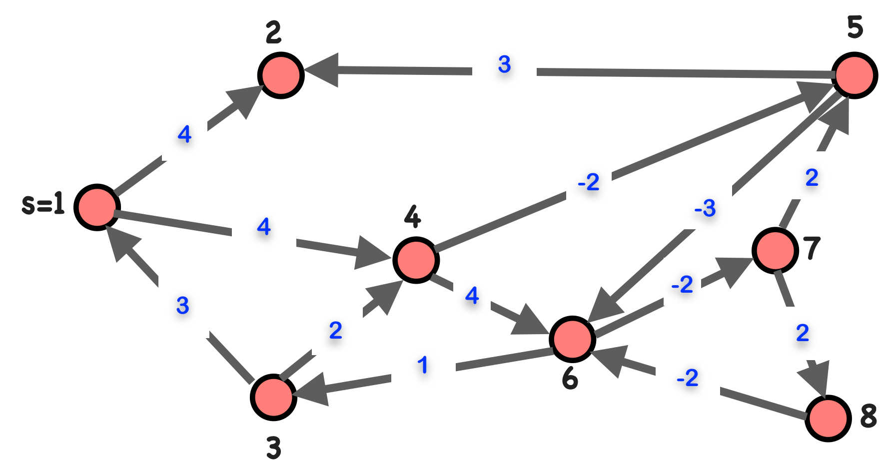
	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf

Example



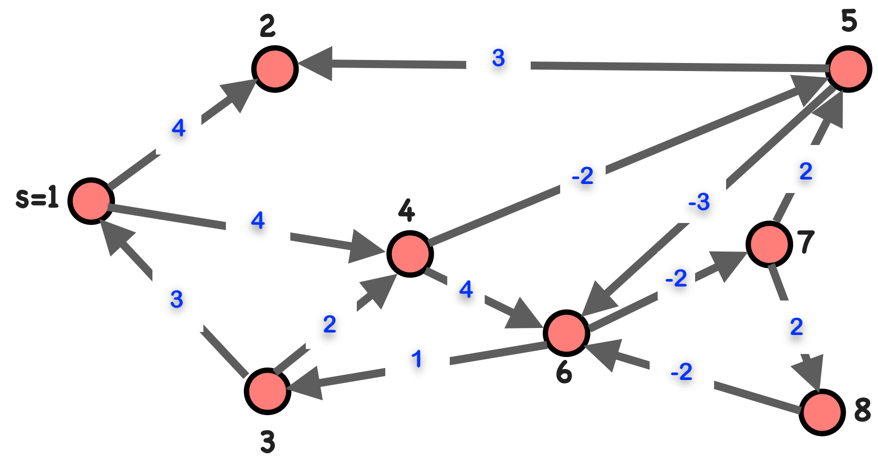
	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8

Example



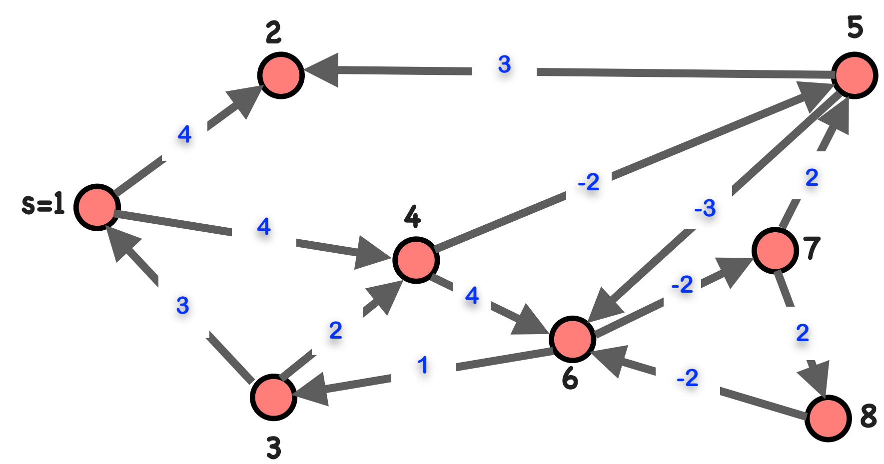
	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8
5	0	4	0	2	-1	-1	-3	-1

Example



	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8
5	0	4	0	2	-1	-1	-3	-1
6	0	2	0	2	-1	-4	-3	-1

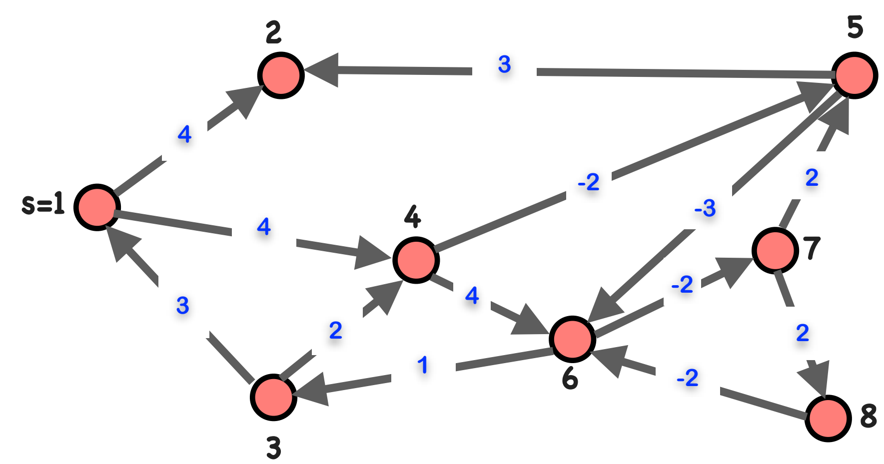
Example



	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8
5	0	4	0	2	-1	-1	-3	-1
6	0	2	0	2	-1	-4	-3	-1
7	0	2	-3	2	-1	-4	-6	-1

The MBF step is finished at this point. We run one more step

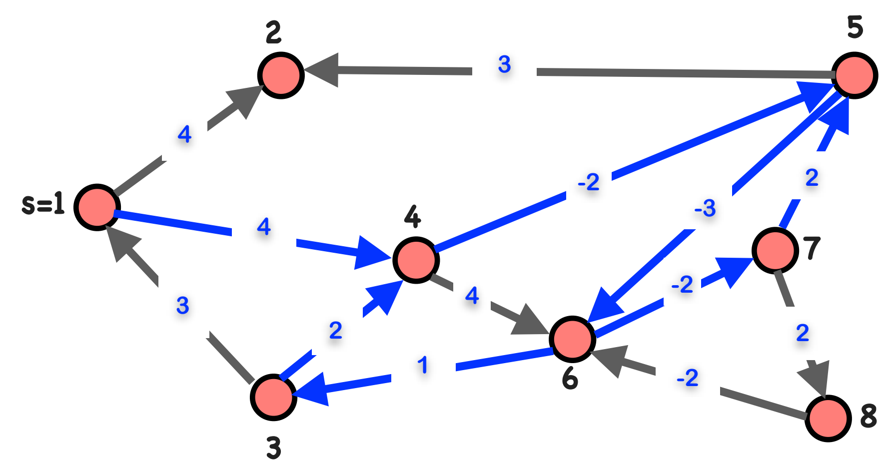
Example



	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8
5	0	4	0	2	-1	-1	-3	-1
6	0	2	0	2	-1	-4	-3	-1
7	0	2	-3	2	-1	-4	-6	-1
8	0	2	-3	-1	-4	-4	-6	-4

These values changed, so we have at least one negative cycle.

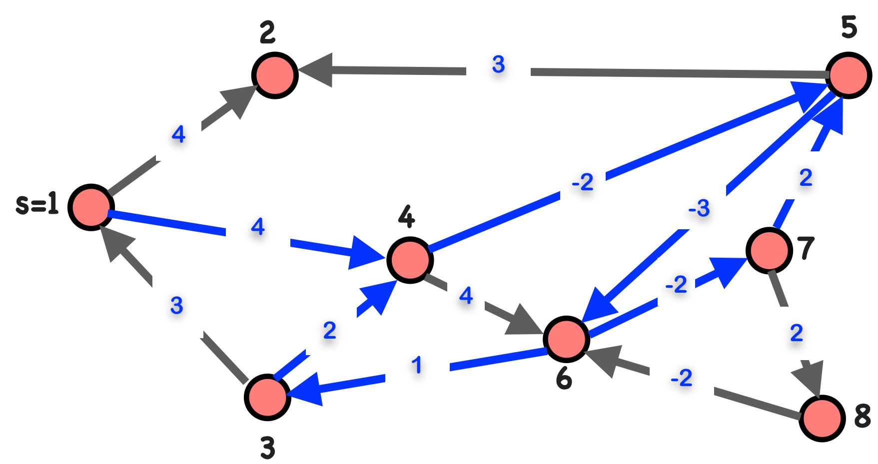
Example



	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8
5	0	4	0	2	-1	-1	-3	-1
6	0	2	0	2	-1	-4	-3	-1
7	0	2	-3	2	-1	-4	-6	-1
8	0	2	-3	-1	-4	-4	-6	-4

To find one, we follow the arrows
in the reverse direction towards s

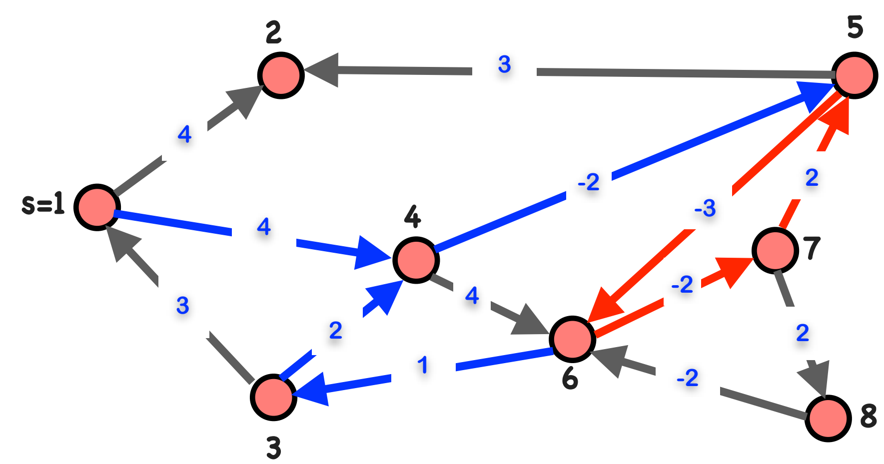
Example



	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8
5	0	4	0	2	-1	-1	-3	-1
6	0	2	0	2	-1	-4	-3	-1
7	0	2	-3	2	-1	-4	-6	-1
8	0	2	-3	-1	-4	-4	-6	-4

If the path crosses the same column twice, then the corresponding node is on a negative cycle

Example



	1	2	3	4	5	6	7	8
0	0	inf	inf	inf	inf	inf	inf	inf
1	0	4	inf	4	inf	inf	inf	inf
2	0	4	inf	4	2	8	inf	inf
3	0	4	9	4	2	-1	6	inf
4	0	4	0	4	2	-1	-3	8
5	0	4	0	2	-1	-1	-3	-1
6	0	2	0	2	-1	-4	-3	-1
7	0	2	-3	2	-1	-4	-6	-1
8	0	2	-3	-1	-4	-4	-6	-4

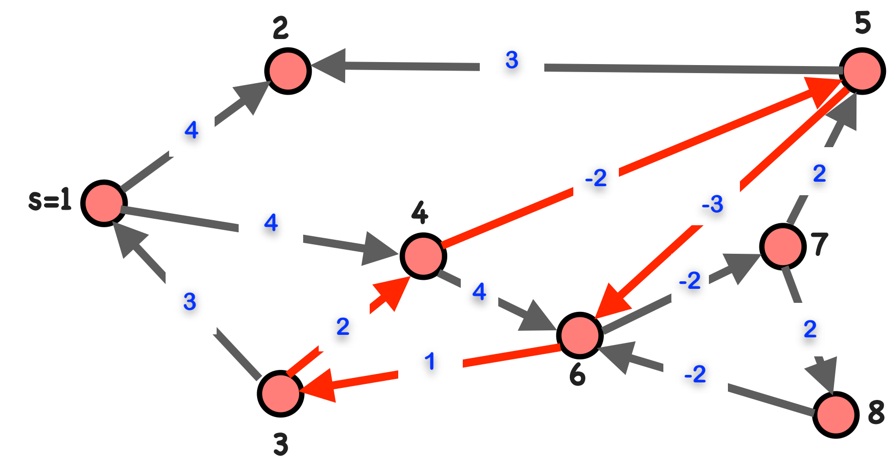
If the path crosses the same column twice, then the corresponding node is on a negative cycle

Karp's Algorithm

What is the smallest *average* weight of a cycle in the graph?

$$C : v_0 - v_1 - \cdots - v_{t-1} - v_t = v_0$$

$$\mu(C) := \frac{1}{t} \sum_{i=1}^{t} w(v_{i-1}, v_i)$$



Average weight of this cycle is (-2-3+1+2)/4 = -1/2

$$\mu^*(G) \leq -\frac{1}{2}$$

$$\mu^*(G) := \min_{C \text{ cycle in } G} \mu(C)$$

Karp has designed an algorithm to compute this number which we will study in the following.

Karp's Algorithm: First Step

For k from 0 to n calculate the shortest path from s to all v using ***exactly*** k edges.

Set value to infinity if no such path exists.

This is the same as adding all non-existent edges to the graph with a weight of infinity, and calculating shortest paths with exactly k edges

Solution: dynamic programming.

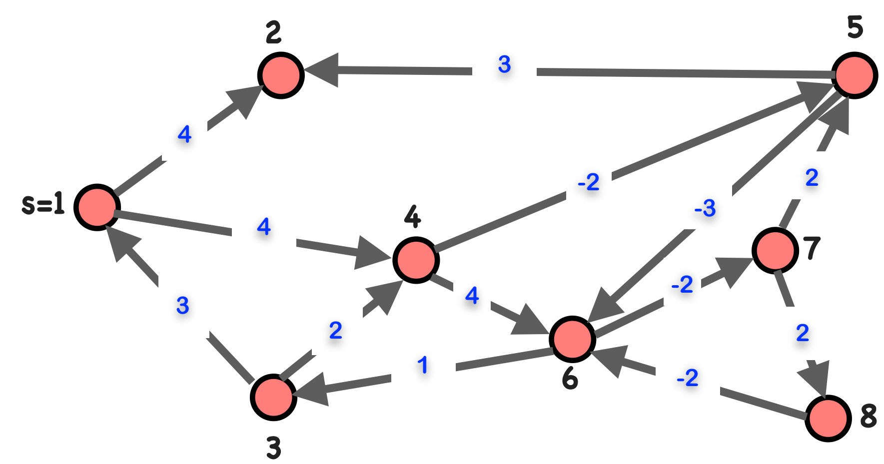
$$F_0(v) := \begin{cases} 0 & \text{if } v = s \\ \infty & \text{else} \end{cases}$$

And for k=1,...,n

$$F_k(v) := \begin{cases} \min_{(u,v)\in E} F_{k-1}(u) + w(u,v) & \text{if } \exists (u,v) \in E \\ \infty & \text{else} \end{cases}$$

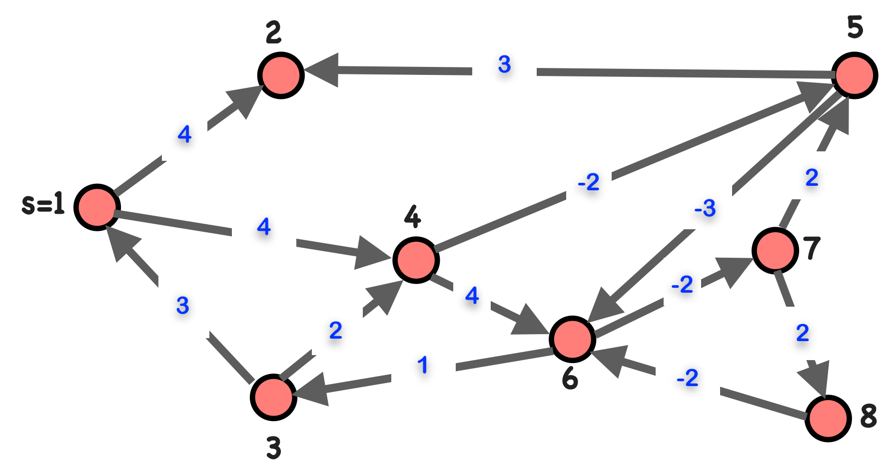
Extend to path to u by one more edge to obtain path with k edges

Example



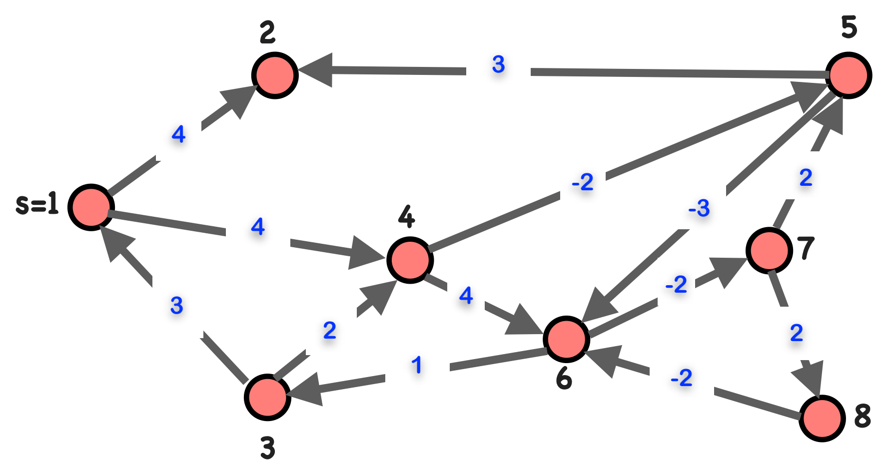
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf

Example



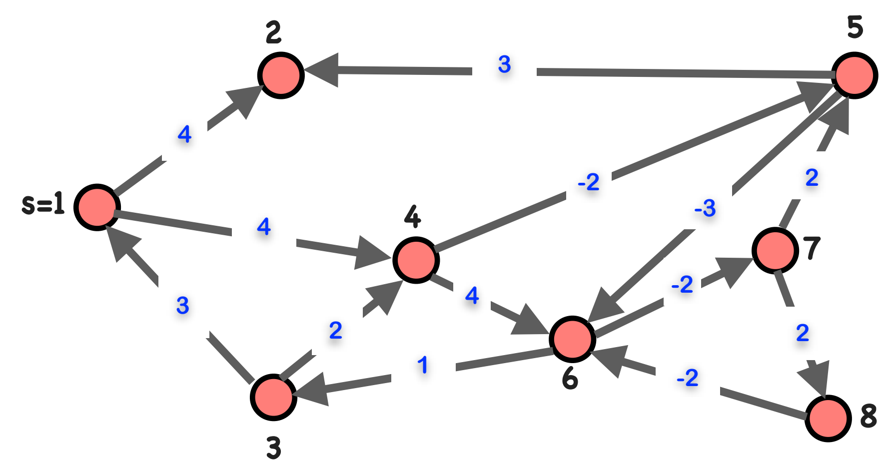
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf

Example



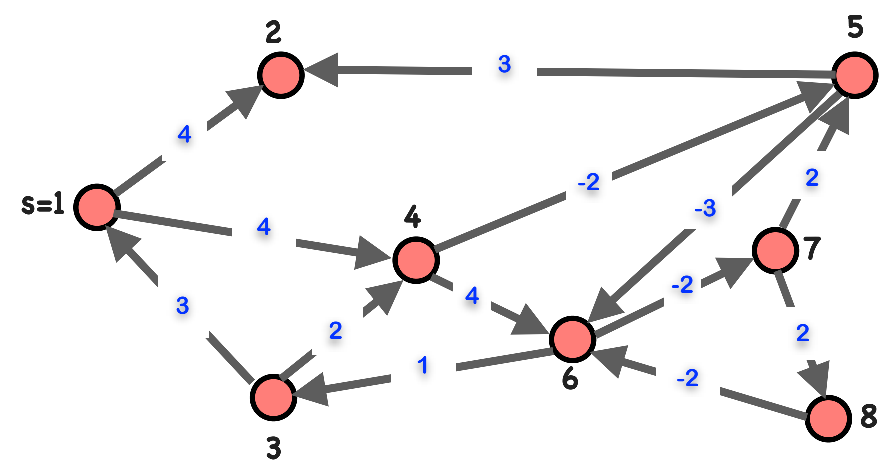
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf

Example



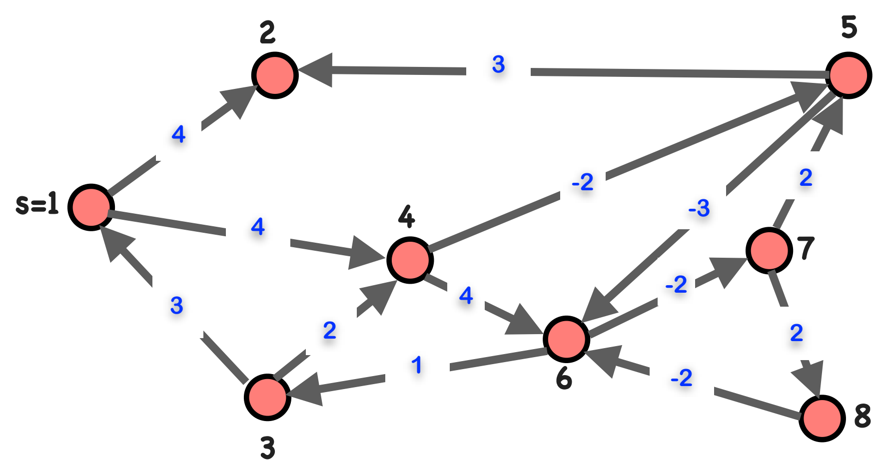
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf

Example



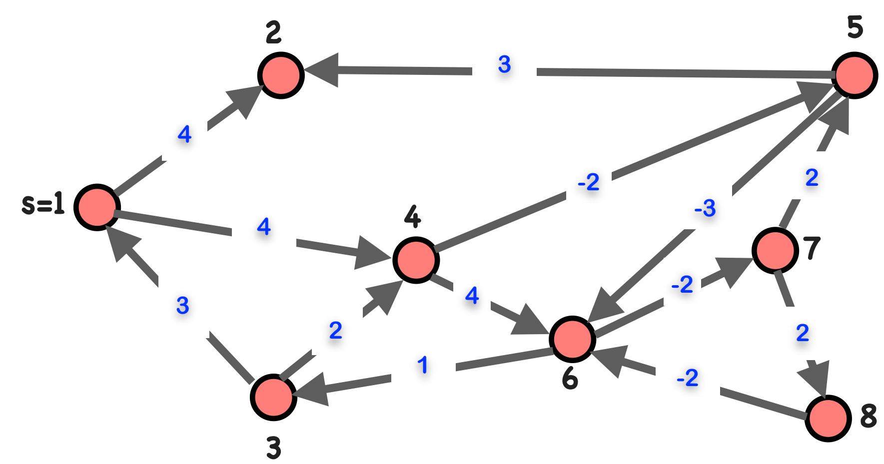
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8

Example



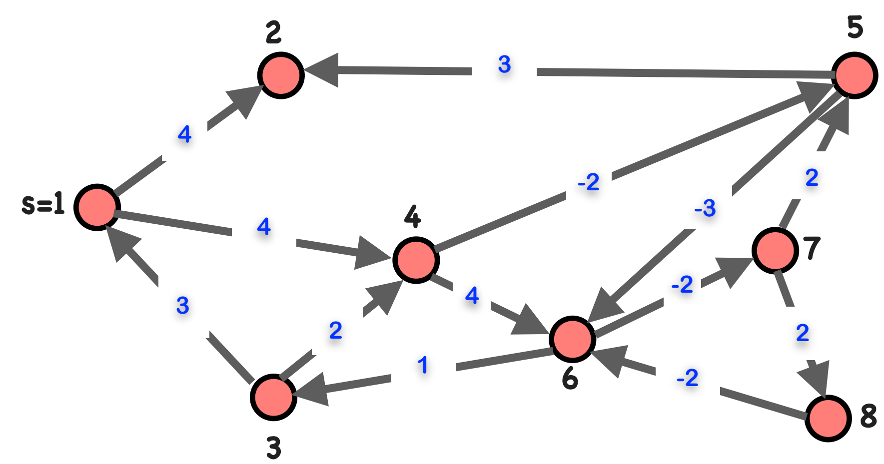
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1

Example



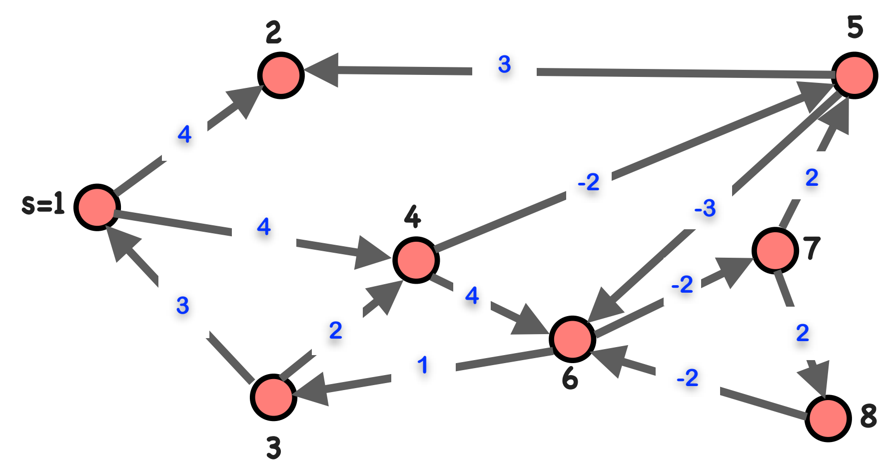
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf

Example



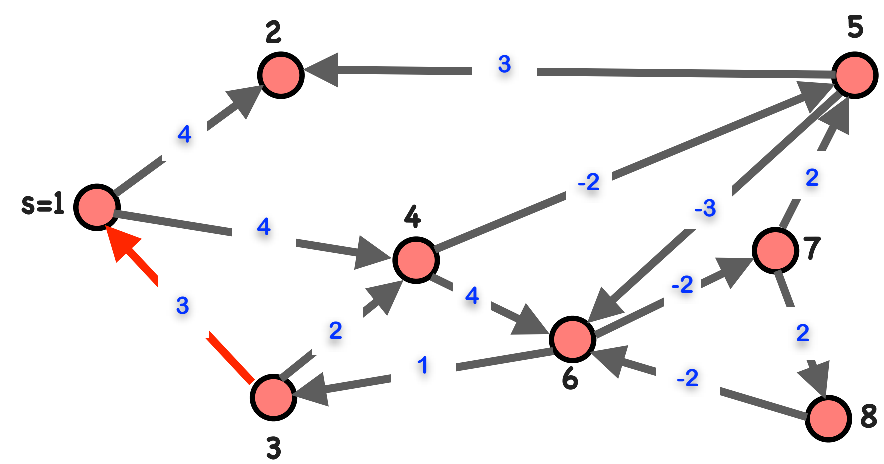
		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5

Example



		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Reading the Paths

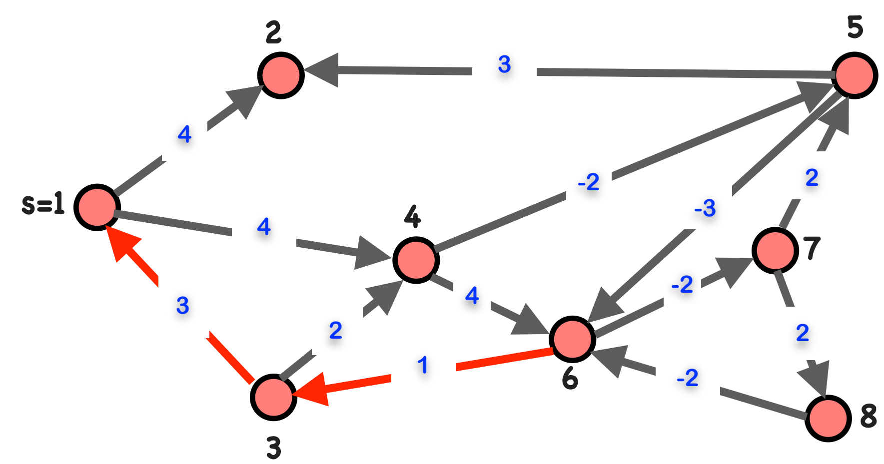


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

-3-1 Current path

Reading the Paths

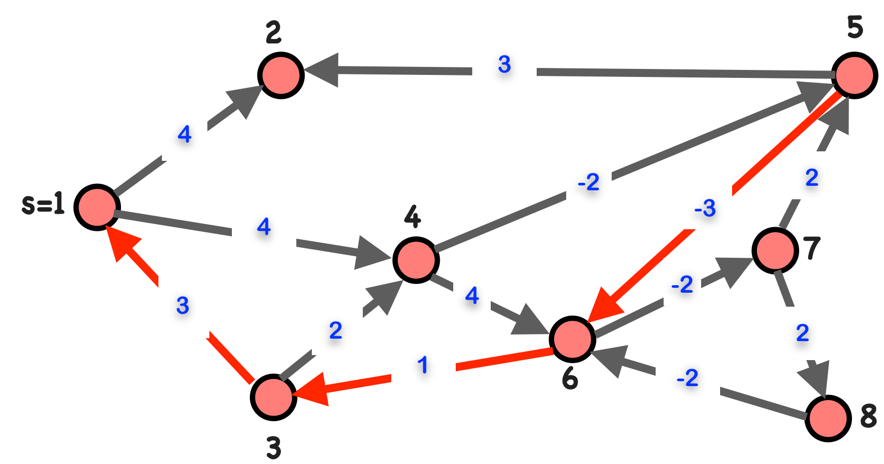


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

-6-3-1 Current path

Reading the Paths

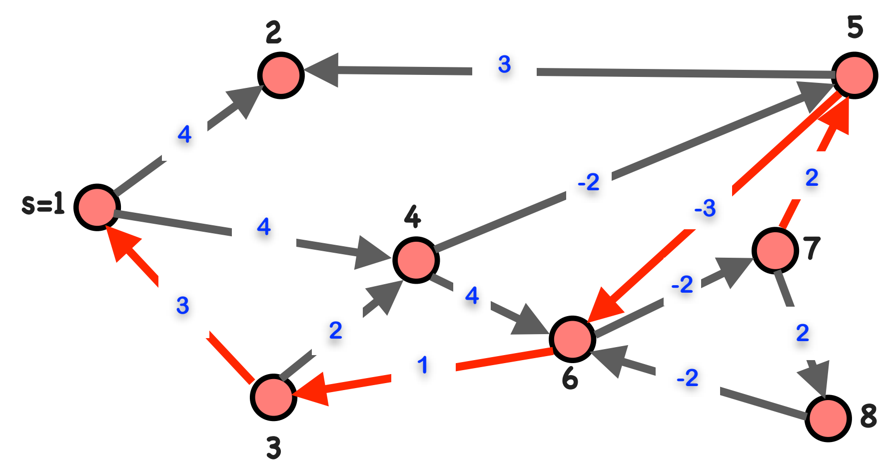


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

-5-6-3-1 Current path

Reading the Paths

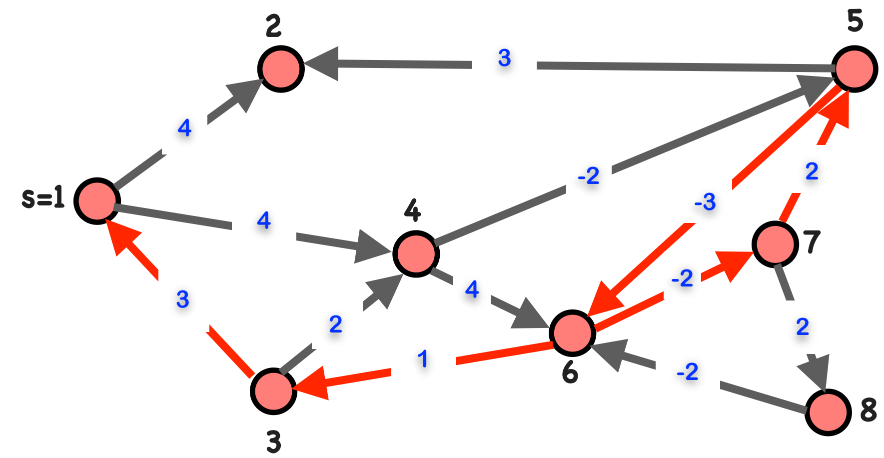


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

-7-5-6-3-1 Current path

Reading the Paths

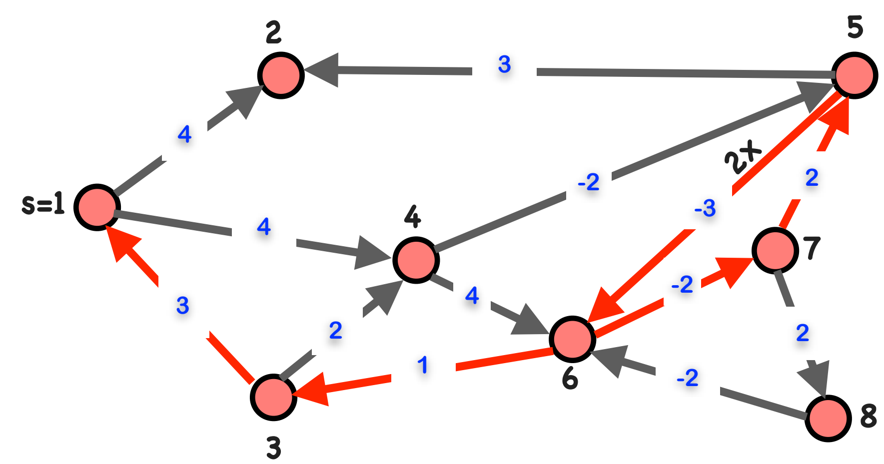


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

-6-7-5-6-3-1 Current path

Reading the Paths

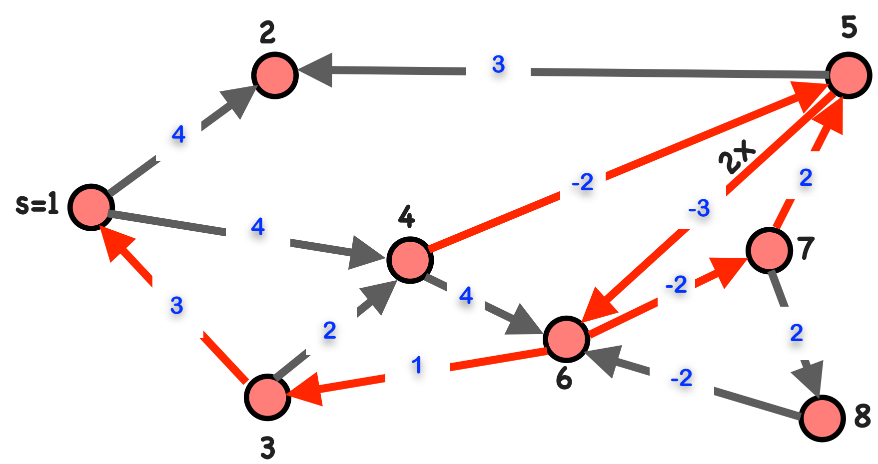


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

-5-6-7-5-6-3-1 Current path

Reading the Paths

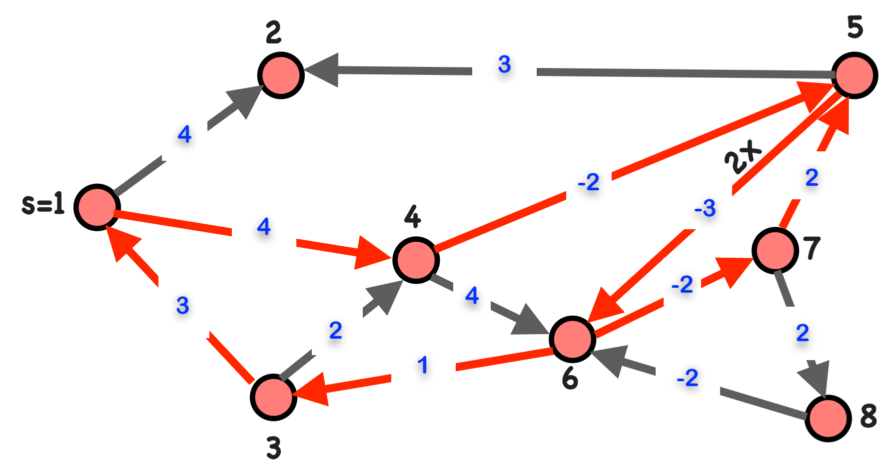


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

-4-5-6-7-5-6-3-1 Current path

Reading the Paths



		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

Follow the arrows backwards

1-4-5-6-7-5-6-3-1 Current path

Karp's Algorithm: Second Step

Now we have calculated $F_k(v)$ for all k=0,...,n and all nodes v.

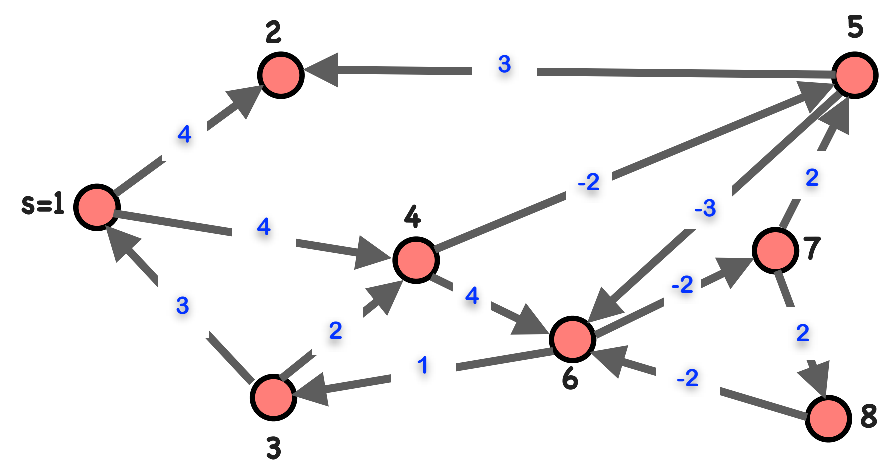
For all v, calculate

$$\alpha(v) := \max_{0 \leq k < n} \frac{F_n(v) - F_k(v)}{n - k}$$

Karp's theorem says:

$$\mu^*(G) = \min_{v \in V} \alpha(v)$$

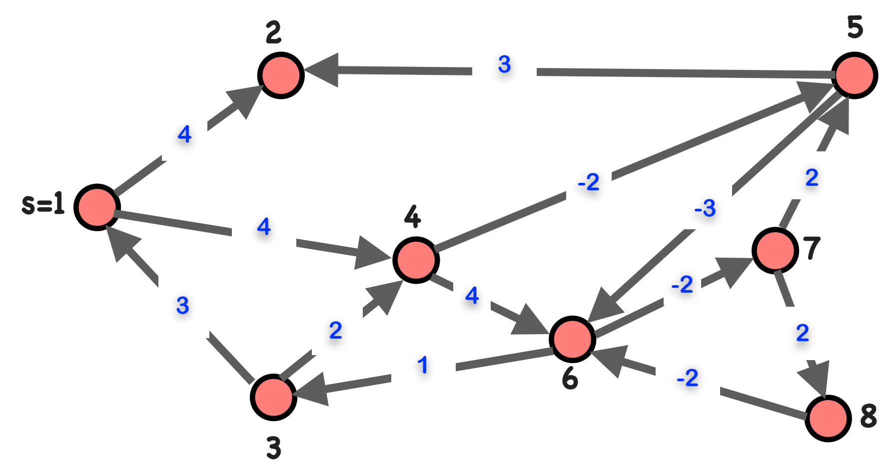
Example



		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4

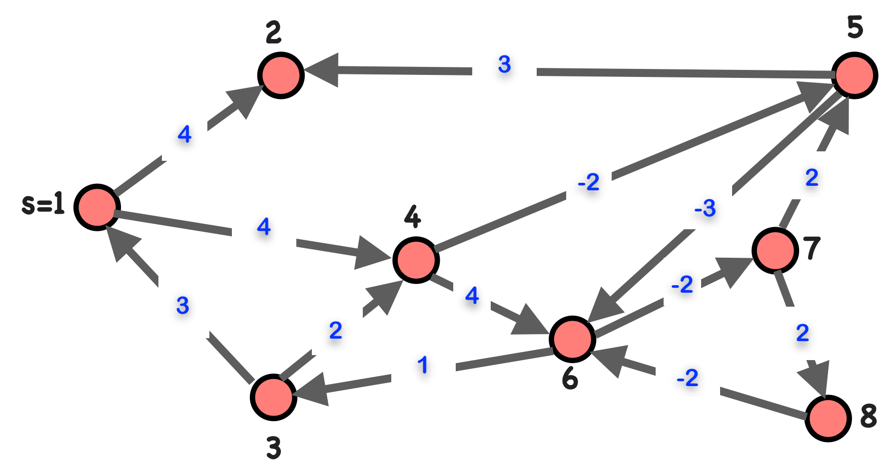
alpha(3) = max((-2+3)/1, (-2-6)/2, (-2-inf)/3, (-2-0)/4, (-2-9)/5,(-2-inf)/8)=
max(1,-4,-inf,-1/2,-11/5) = 1

Example



		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4
	alpha	0	5	1	-5/7	-1	5	1	-1

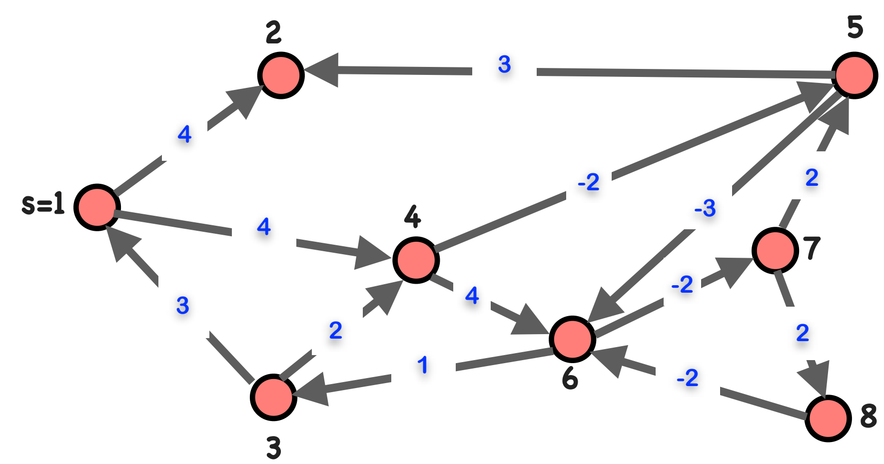
Example



	1	2	3	4	5	6	7	8
$F_0(v)$ 0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$ 1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$ 2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$ 3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$ 4	12	inf	0	11	8	inf	-3	8
$F_5(v)$ 5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$ 6	inf	2	6	7	0	-4	3	inf
$F_7(v)$ 7	9	3	-3	8	5	-3	-6	5
$F_8(v)$ 8	0	8	-2	-1	-4	2	-5	-4
alpha	0	5	1	-5/7	-1	5	1	-1

(-4-2)/(8-2)=-1

Example

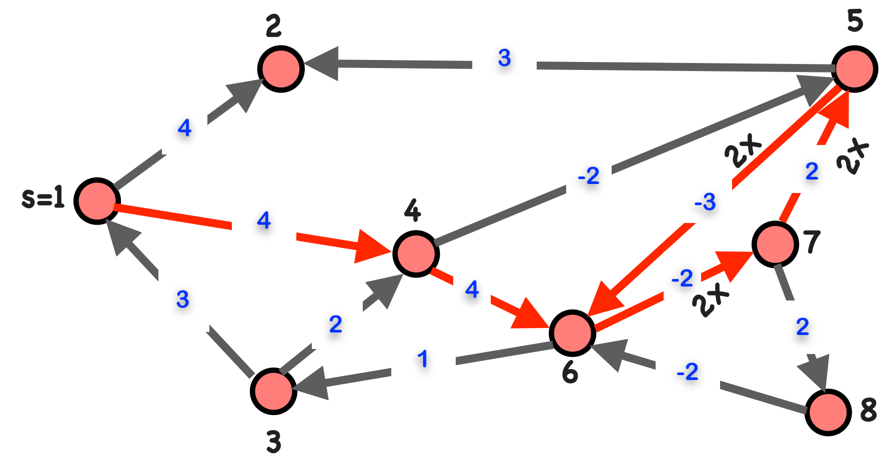


		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4
	alpha	0	5	1	-5/7	-1	5	1	-1

$\mu^*(G) = -1$

(-4-2)/(8-2)=-1

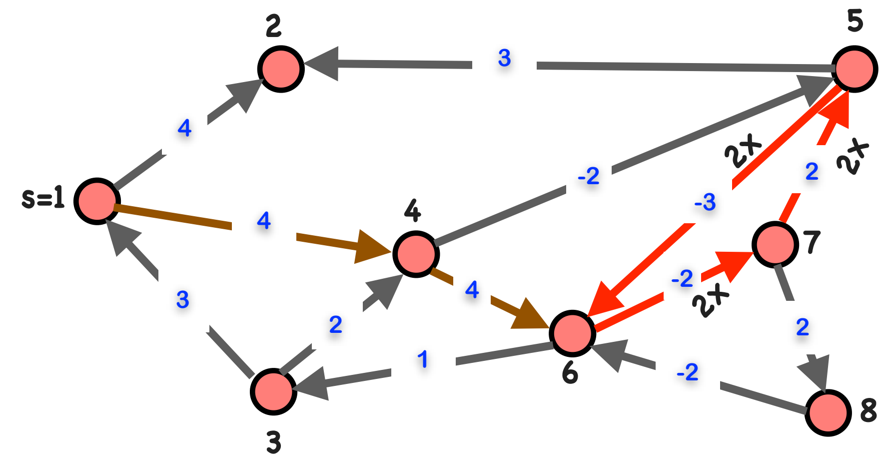
What does it Have to do with Cycles?



		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4
	alpha	0	5	1	-5/7	-1	5	1	-1

Path of length 8
and cost -4

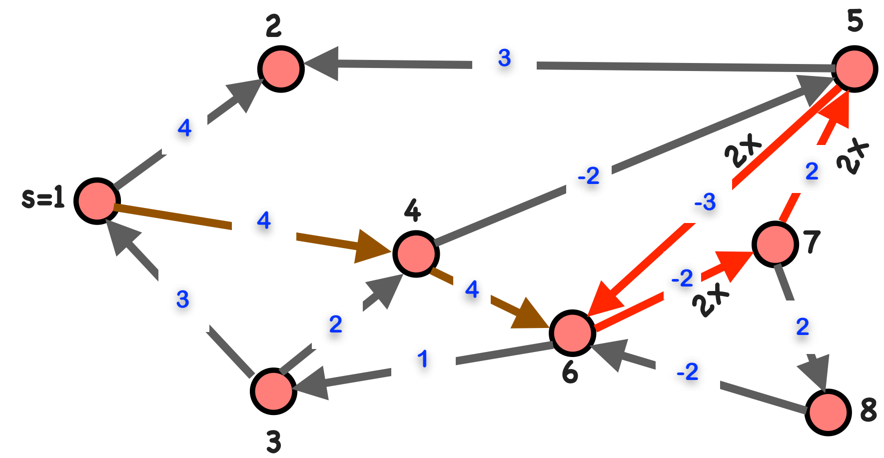
What does it Have to do with Cycles?



		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4
	alpha	0	5	1	-5/7	-1	5	1	-1

Path of length 2
and cost 2

What does it Have to do with Cycles?



		1	2	3	4	5	6	7	8
$F_0(v)$	0	0	inf	inf	inf	inf	inf	inf	inf
$F_1(v)$	1	inf	4	inf	4	inf	inf	inf	inf
$F_2(v)$	2	inf	inf	inf	inf	2	8	inf	inf
$F_3(v)$	3	inf	5	9	inf	inf	-1	6	inf
$F_4(v)$	4	12	inf	0	11	8	inf	-3	8
$F_5(v)$	5	3	11	inf	2	-1	5	inf	-1
$F_6(v)$	6	inf	2	6	7	0	-4	3	inf
$F_7(v)$	7	9	3	-3	8	5	-3	-6	5
$F_8(v)$	8	0	8	-2	-1	-4	2	-5	-4
	alpha	0	5	1	-5/7	-1	5	1	-1

What remains is a cycle with 6 edges and cost -6

$F_n(v)-F_k(v)$ is the weight of a cycle with n-k edges starting and ending in v

Proof of Karp's Therem

Now we have calculated $F_k(v)$ for all k=0,...,n and all nodes v.

For all v, calculate

$$\alpha(v) := \max_{0 \leq k < n} \frac{F_n(v) - F_k(v)}{n - k}$$

Karp's theorem says:

$$\mu^*(G) = \min_{v \in V} \alpha(v)$$

First Step: Zero Cycles

Consider the graph $\hat{G}$ obtained from G by subtracting from all edge values the quantity $\mu^*(G)$

$\mu^*(\hat{G}) = 0$

Proof: weights in new graph are $\hat{w}(u, v) = w(u, v) - \mu^*(G)$

Average weight of a cycle in the new graph is

$$\frac{1}{t}\sum_{i=1}^{t} \hat{w}(v_{i-1}, v_i) = \mu(C) - \mu^*(G)$$

All cycle weights in the old graph are at least $\mu^*(G)$.

So, smallest average weight cycle weight in new graph is 0. Q.E.D.

Without loss of generality: consider a graph in which smallest average cycle length is 0.

Second Step:
Zero is Smallest Average Cycle Weight

Need to show that

$$\min_{v\in V}\max_{0\le k<n}\frac{F_n(v)-F_k(v)}{n-k}=0$$

Since $\mu^*(G)=0$, there are no negative cycles in the graph, so

$$\sigma := \min_{k=0,\ldots,n-1} F_k(v)$$

is the length of the shortest path from s to v. On the other hand, $F_n(v)$ is the length of the shortest path from s to v with exactly n edges, so it is not smaller than σ. Therefore

$$\max_{k=0,\ldots,n-1} F_n(v)-F_k(v)\ge 0$$

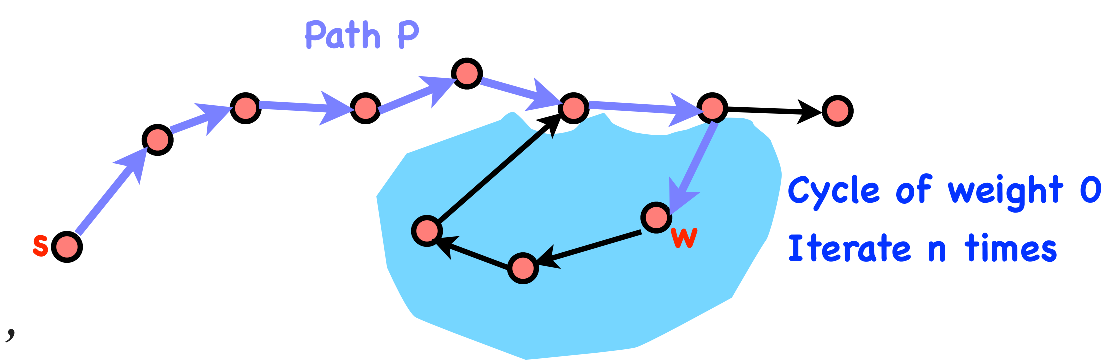
We therefore see that

$$\min_{v\in V}\max_{0\le k<n}\frac{F_n(v)-F_k(v)}{n-k}\ge 0$$

Need to show: there are v and k such that $F_n(v)-F_k(v)=0$

Second Step:
Zero is Smallest Average Cycle Weight

Need to show: there are *v* and *k* such that $F_n(v) - F_k(v) = 0$



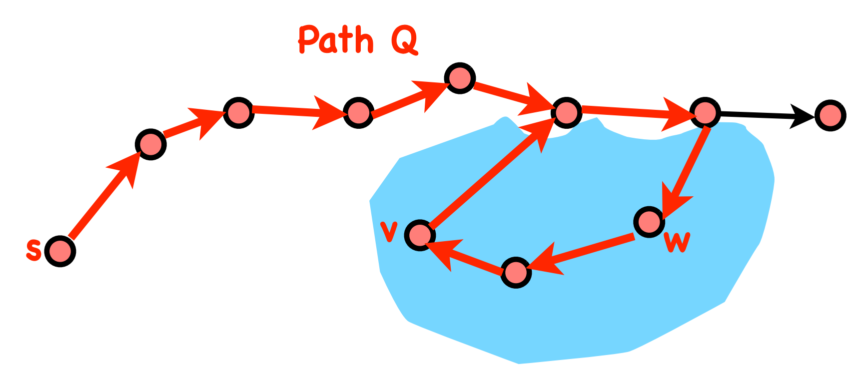
Take cycle *C* of weight 0, node *w* on *C*, and *simple* path *P* from *s* to *w*. **Simple path is one in which no node is repeated**

Extend path *P* by *n* iterations of cycle *C* to obtain a path *P'*.

This path has at least *n* edges.

Let *Q* be the path formed by the first *n* edges of *P'*

v is final node on path *Q*



$$Q\colon s = v_0 \to v_1 \to \cdots \to v_{n-1} \to v_n = v$$

Second Step: Zero is Smallest Average Cycle Weight

Need to show: there are v and k such that $F_n(v) - F_k(v) = 0$

Take cycle C of weight 0, node w on C, and *simple* path P from s to w. Simple path is one in which no node is repeated

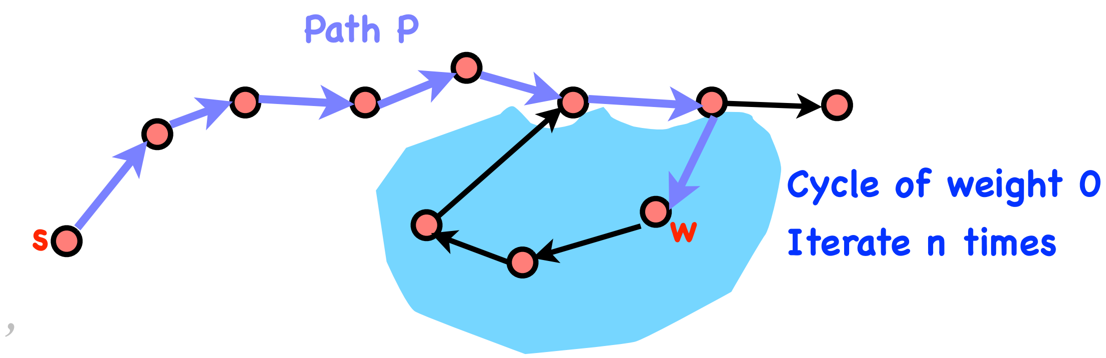
Extend path P by n iterations of cycle C to obtain a path P'.

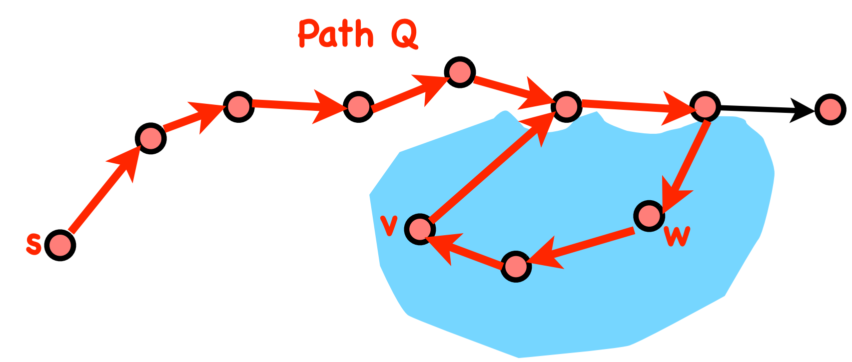
This path has at least n edges.

Let Q be the path formed by the first n edges of P'

v is final node on path Q

$$Q: s = v_0 \to v_1 \to \cdots \to v_{n-1} \to v_n = v$$





k smallest index such that $v = v_k$ Note: k < n by choice of path

Then, $F_k(v) = F_n(v)$ since cycle C has zero weight Q.E.D.

Karp's Algorithm

(1) Set $F_0(s)=0$, $F_0(v)=inf$ for v not s

Initiatlization

(2) For i from 1 to n do

  (i) For v in V set $F_i(v) = inf$

  (ii) For (u,v) in E do

    (a) Set $F_i(v) = min(F_i(v), F_{i-1}(v)+w(u,v))$

This loop calculates the Fi(v) for all v in V and all i from 0 to n

(3) For v in V do

  (i) Set alpha(v)=-inf

  (ii) For i from 1 to n-1 do

    (a) Set alpha(v) = max(alpha(v), $(F_n(v)-F_i(v))/(n-i)$)

This loop calculates the alpha values

(4) Set mu=alpha(s)

(5) For v in V do

  (i) Set mu = min(mu, alpha(v))

This loop calculates the final result

(6) Return mu

Karp's Algorithm

(1) Set $F_0(s)=0$, $F_0(v)=\text{inf}$ for v not s — O(n)

(2) For i from 1 to n do — O(|E| n)

  (i) For v in V set $F_i(v) = \text{inf}$

  (ii) For (u,v) in E do

    (a) Set $F_i(v) = \min(F_i(v), F_{i-1}(v)+w(u,v))$

(3) For v in V do — O(n^2)

  (i) Set alpha(v)=-inf

  (ii) For i from 1 to n-1 do

    (a) Set alpha(v) = max(alpha(v), $(F_n(v)-F_i(v))/(n-i)$)

(4) Set mu=alpha(s)

(5) For v in V do — O(n)

  (i) Set mu = min(mu, alpha(v))

(6) Return mu

Running time is O(|E||V|)