SHORTEST PATH PROBLEM, REVISITED

Thursday, November 8, 2018 5:08 PM

(SINGLE-SOURCE SHORTEST PATHS)

INPUT: a directed, weight graph $G=(V,E,w)$ and a source vertex s.

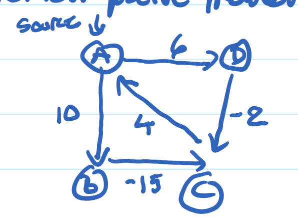
OUTPUT: shortest paths from s to the other vertices v in G.

1) If $w(i,j) \geq 0$ for all i,j, then a GREEDY solution exists (Dijkstra's alg)
2) There exist graphs with negative weights that Dijkstra's alg gives the wrong answer

   Exercise: find a simple example

3) The shortest path problem may not be well-defined if negative weights are allowed

Ex:



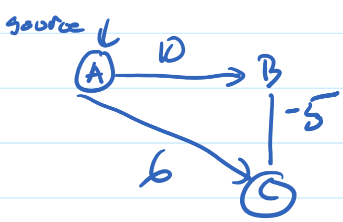
What is the shortest path from A to C?

$A \xrightarrow{10} B \xrightarrow{-15} C$: distance -5

$A \xrightarrow{10} B \xrightarrow{-15} C \xrightarrow{4} A \xrightarrow{10} B \xrightarrow{-15} C$: distance -6

In general, SP problem is not well-defined if there is a cycle whose total weight is negative.

Ex:

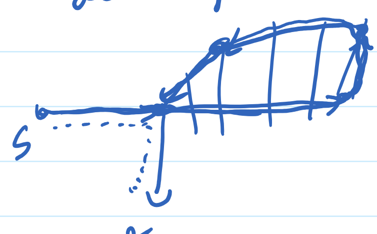


shortest path problem is well-defined here

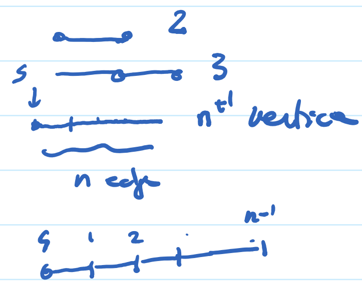
WANT: an alg to solve the SP problem when negative weights are allowed BUT NO NEGATIVE CYCLES. Further, the alg should detect existence of NEGATIVE CYCLES.

CLAIM: if no negative cycles exist, any shortest path does not repeat a vertex

Proof:



Corollary: shortest paths contain at most $n-1$ edges



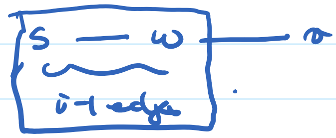
BELLMAN-FORD Alg

DP: subproblems $d[v][i]$ = shortest distance from s to v USING AT MOST i edges.

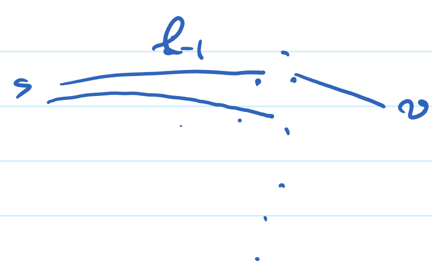
Base case: $d[v][0] = \begin{cases} 0 & \text{if } v = s \\ \infty & \text{otherwise} \end{cases}$

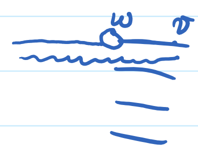
General case:

shortest path from s to v with i edges



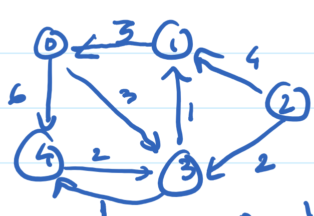

```
BELLMAN-FORD (V, E, W, s)
{
    for each v ∈ V
        d[v][0] = ∞
    d[s][0] = 0;
    for (l = 1; l < |V|; ++l)
    {
        for each v ∈ V
            for each w ∈ V
                if d[w][i-1] + W(w, v) < d[v][i-1]
                    d[v][i] = d[w][i-1] + W(w, v)
    }
}
```






```
for each edge (w,v) ∈ E
    if d[w][i-1] + W(w,v) < d[v][i-1]
        d[v][i] = d[w][i-1] + W(w,v)
```

Running time $\Theta(|V||E|) = \Theta(nm)$

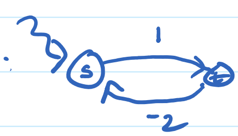
Ex: 

Source = 2

$l = 0$

$l = 1$

	0	1	2	3	4
d[0]	∞	∞	0	∞	∞
d[1]	∞	4	0	2	∞
d[2]	7	3	0	2	3
d[3]	6	3	0	2	3

Ex: 

	s	t
$l=0$	0	∞
$l=1$	0	1
$l=2$	-1	1
$l=3$	-1	0

d[2]	7	3	0	2	3	
d[3]	6	3	0	2	3	
d[4]	6	3	0	2	3	

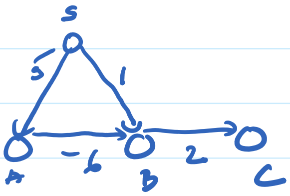
$\ell = 3$ | -1 | 0

⋮

STOP when there are no changes

If there are changes at step n, then NEGATIVE CYCLES EXIST

Ex:



	✓ S	✓ A	✓ B	✓ C
	0	∞	∞	∞
	0	5	1	∞
	0	5	1	3

Dikjstra's alg fails !!

$d(B) = -1$ not 1

	S	A	B	C
0	0	∞	∞	∞
1	0	5	1	∞
2	0	5	-1	3
3	0	5	-1	1

Bellman-Ford

ALL-PAIRS SHORTEST PATH

Input: (V, E, w) weighted digraph

Output: shortest path from s to t for EVERY pair s, t

Transform: call Bellman-Ford n times for each source vertex

$\Theta(n^2 m)$; when graph is dense $m \in \Theta(n^2)$

this alg runs in time $\Theta(n^4)$

An $O(n^3)$ dynamic-programming solution

Assume vertices are labeled from 1 to n.

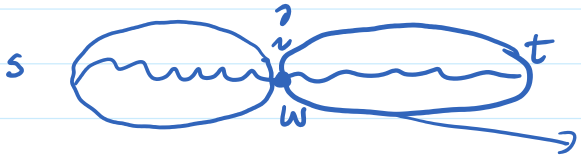
SUBPROBLEMS: find shortest path from s to t whose intermediate vertices have label $\leq i$

BASE CASES: $i = 0$ (no intermediate vertices allowed)

$d[s][t] = \begin{cases} 0 & \text{if } s = t \\ w(s,t) & \text{if } (s,t) \in E \end{cases}$

$$\begin{cases} w(s,t) & \text{if } (s,t) \in E \\ \infty & \text{otherwise} \end{cases}$$

GENERAL CASE:



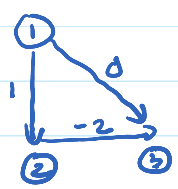
intermediate vertices $\le i$

$$d[s][t][i] = \min\big(d[s][t][i-1],\ d[s][i][i-1] + d[i][t][i-1]\big)$$

```
WARSHALL-FLOYD (V, E, w)
{
    for each s ∈ V
        for each t ∈ V
            if (s == t)
                d[s][t][0] = 0
            else if (s,t) ∈ E
                d[s][t][0] = w(s,t)
            else
                d[s][t][0] = ∞
    for (i = 1; i ≤ n; ++i)
        for each s ∈ V
            for each t ∈ V
                d[s][t][i] = min( d[s][t][i-1],
                                  d[s][i][i-1] + d[i][t][i-1]
                                );
}
```

$\Theta(n^3)$

Ex:



$i = 0$

	1	2	3
1	0	1	0
2	∞	0	-2
3	∞	∞	0

$i = 1$

	1	2	3
1	0	1	0
2	∞	0	-2
3	∞	∞	0

$i = 2$

	1	2	3
		1	-1

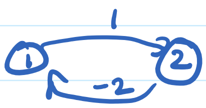
$i=2$

	1	2	3
1	0	1	-1
2	∞	0	-2
3	∞	∞	0

$i=3$

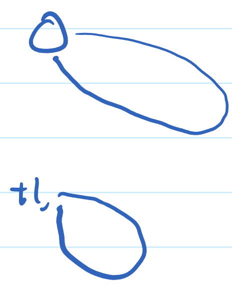
	1	2	3
1	0	1	-1
2	∞	0	-2
3	∞	∞	0

WARSHALL - FLOYD can detect NEGATIVE CYCLES also



$i=0$

	1	2
1	0	1
2	-2	0



$i=1$

	1	2
1	0	1
2	-2	-1

there is a negative cycle if negative DIAGONAL elements exist

$i=2$

	1	2
1	-1	0
2	-3	-2