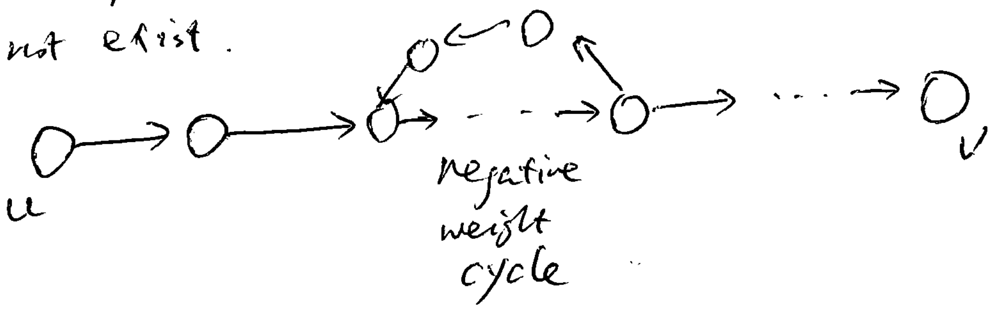
# Bellman-Ford algorithm

- Dijkstra's algorithm assumes all $w(u,v) \geq 0$, for all $(u,v) \in E$
- If a graph has a negative weight cycle, then some $\delta(u,v)$ may not exist.




- Bellman-Ford can handle negative-weight cycle.

Algorithm: computes the shortest path weights $\delta(s,v)$ from source vertex $s \in V$ to all vertices $v \in V$

OR report that a negative-weight cycle exists.

Pseudo-code

init.

$d[s] \leftarrow 0$
for each $v \in V - \{s\}$
  do $d[v] = \infty$

$d[v]$ is the estimate for $\delta(s,v)$

relaxation

for $i \leftarrow 1$ to $|V|-1$
do for each edge $(u,v) \in E$
  do if $d[v] > d[u] + w(u,v)$
    $d[v] = d[u] + w(u,v)$

we won't use $i$. it's just a counter

We basically relax every edge $|V|-1$ times here.

validation
For each edge $(u,v) \in E$
if $d[v] > d[u] + w(u,v)$
then report that a negative-weight cycle exists.

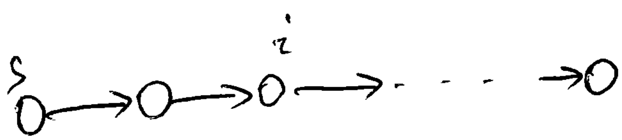
OR

$\delta(s,v) = d[v]$ for all $v \in V$

Q:/ running time ?

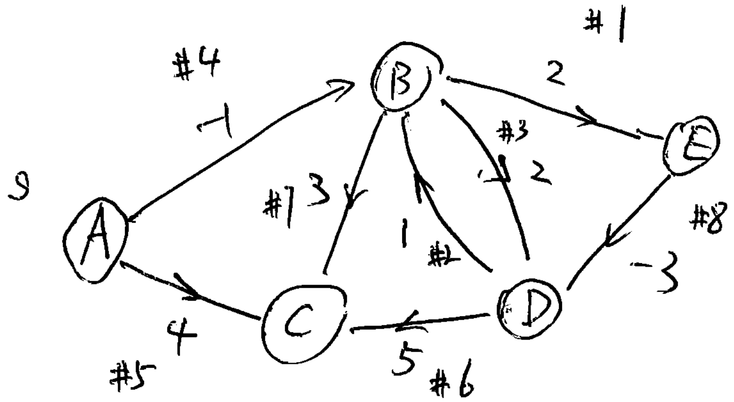
A:/ $O(EV)$ - slower than Dijkstra

The key proof of this algorithm is that after $|V|-1$ iterations, we would have found any negative-weight cycle if any. Thus the validation part is to verify that.




after at most $|V|-$ rounds, we are done
because each round finds $d[i]$ as $\delta(s,i)$

e.g.




| Steps on edge | A | B | C | D | E | |
|---|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ | init. |
| on #4 | 0 | -1 | ∞ | ∞ | ∞ | i=1 |
| #5 | 0 | -1 | 4 | ∞ | ∞ | |
| #7 | 0 | -1 | 2 | ∞ | ∞ | |
| #1 | 0 | -1 | 2 | ∞ | 1 | i=2 |
| #3 | 0 | -1 | 2 | 1 | 1 | |
| #8 | 0 | -1 | 2 | -2 | 1 | |

no change for i=3,4

#

The validation step also checks out.

In fact, stop if no change during a round.