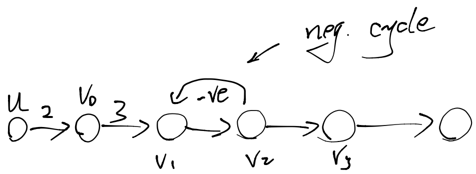
Bellman-Ford Algorithm



$\delta(u, v_1)$
$\delta(u, v_2)$ undefined
$\vdots$

Generate Shortest Path algorithm

- Initialize for $v \in V$, $d[v] \leftarrow \infty$, $\pi[v] = NIL$
- $d[s] \leftarrow 0$
- Main repeat select edge [somehow], relax edge(u, v, w)
  Until can't relax any more.

Problems.

① Complexity could be exp time (even for positive edge weights) $O(2^{n/2})$

② Will not terminate if there is a negative cycle reachable from the source.

Bellman-Ford Algorithm

Bellman-Ford (G, W, S)

Initialize ( );

$O(VE)$ for $i = 1$ to $|V| - 1$

for each edge $(u, v) \in E$

Relax (u, v, w)

Relax (u, v, w)

if $d[v] > d[u] + w(u, v)$

$\{ d[v] = d[u] + w(u, v)$

$\pi[v] = u \}$

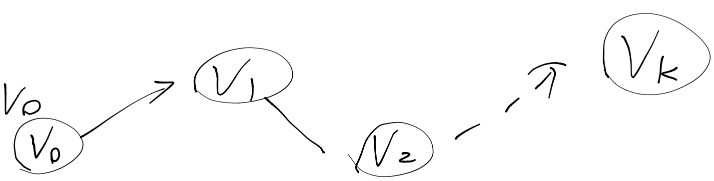
$O(E)$ for each edge $(u,v) \in E$:
      if $d[v] > d[u] + w[u,v]$
         report neg. cycle exists   } check neg. cycles

∘ Theorem

If $G=(V,E)$ contains no neg. weight edges, then after B-F algorithm executes, $d[v] = \delta(s,v)$ for all $v \in V$

Corollary: If a value $d[v]$ fails to converge after $|V|-1$ cycles, there exists a neg. cycle that reachable from S.

$S = V_0$ $v = V_k$



Path: $V_0, V_1, V_2, \ldots\ldots V_{k-1}$

$k \le |V|-1$, else we have neg. cycle

∘ Proof theorem:

Let $v \in V$, $p = \langle v_0, v_1, v_2, \ldots v_k \rangle$ $v_0 = s$ to $v_k = v$

this path p is a shortest path with min # edges

no neg-cycles implies p is simple $\Rightarrow k \le |V|-1$

After 1 path thru all edges E, we have

$d[v_1] = \delta(s, v_1)$, because we relax (v_0, v_1)

~~After~~ After 2 pass, $d[v_2] = \delta(s, v_2)$, because in 2^{nd} pass we will relax (v_1, v_2)

After k passes, $d[v_k] = \delta(s, v_k)$,

After $|V|-1$ passes $\Rightarrow$ all reachable vertex has δ values.