Bellman-Ford Algorithm

1958  

1956

Input: directed graph $G = (V, E)$
each edge $(u, v)$ in $E$ has a weight $w(t(u, v))$;
weights can be negative,
however: $G$ does not contain a directed negative-weight cycle
Source vertex $s$.

Output: for each vertex $v$:

$\delta(s, v) =$ length of a shortest path from $s$ to $v$.

Exercise: show that $\delta(s, s) = 0$.  

As before: for each vertex \( v \) maintain a variable
\[
d(v) = \text{length of a shortest path from } s \text{ to } v \text{ found so far.}
\]

**Bellman-Ford:**
for each \( v \in V \):
\[
d(v) = \infty;
\]
\[
d(s) = 0;
\]
for \( i = 1, 2, \ldots, |V| - 1 \):
for each edge \( (u, v) \) in \( E \):
\[
d(v) = \min(d(v), d(u) + wt(u, v))
\]

Running time: \( O(|V| \cdot |E|) \)
Correctness:

Claim 1: At any moment:
\[ \delta(s,v) \leq d(v) \] for every vertex \( v \).

Proof: as on page 93. \( \square \)

Claim 2: Assume, at some moment, \( d(v) \) becomes equal to \( \delta(s,v) \). Then during the rest of the algorithm, \( d(v) \) does not change.

Proof: as on page 93. \( \square \)
Claim: For every vertex $v_j$ at the end of the for-loop, $d(v) = \delta(s, v)$.

Proof: Consider a shortest path from $s$ to $v$:

$$
S = V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow \cdots \rightarrow V_{k-1} \rightarrow V_k = v
$$

Note: this path has $k$ edges, and $k+1$ vertices, $0 \leq k \leq |V| - 1$. [Do you see why?]

We will show: after $k$ iterations of the outer for-loop, $d(v) = \delta(s, v)$.

This, together with Claim 2 on page BF3 will imply the claim.

Induction on $k$.

Base case: $k = 0$, i.e., $v = s$.

After 0 iterations of the for-loop, $d(s) = 0 = \delta(s, s)$.

\[ \therefore \] is true
Let \( k \geq 1 \) and assume \( \ast \) is true for \( k-1 \).

\( S = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_{k-1} \) is a shortest path from \( s \) to \( v_{k-1} \), it has \( k-1 \) edges.

From induction hypothesis:

after \( k-1 \) iterations of the for-loop:

\[
d(v_{k-1}) = \delta(s,v_{k-1}).
\]

During iteration \( k \), the algorithm considers all edges in \( E \), in particular \((v_{k-1},v_k)\).

At the end of iteration \( k \):

\[
d(v_k) \leq d(v_{k-1}) + wt(v_{k-1},v_k)
\]

\[
= \delta(s,v_{k-1}) + wt(v_{k-1},v_k)
\]

\[
= \delta(s,v_k)
\]

\[
\leq d(v_k).
\]

\[
\therefore d(v) = \delta(s,v).
\]

\[\square\]