BELLMAN-FORD ALGORITHM

simple, but made even simpler
(not identifying negative cycles)

A. Shimbel (1954) → slower variant
L. Ford (1956) → slower variant
E. Moore (1957) → non-negative weights
R. Bellman (1958)

Finds a shortest path from s to ALL vertices
1) set score of $s$ : zero
   set score of $\neq s$ : $\infty$
   set parent of $\neq s$ : null

2) for $i = 1$ to $V - 1$
   RELAX every edge in $G$

   RELAX an edge $x \rightarrow y$
   if $d(x) + w(x,y) < d(y)$
   then $d(y) = d(x) + w(x,y)$
   parent$(y) = x$

   Why does this work?
   For any $t$
   In iteration $i$
   we will get $d(p_i)$
   on some shortest path $p^+_i$
   $= \{p_1^+, p_2^+, p_3^+, \ldots, p_k^+ = t\}$
   from $s \rightarrow t$
Depending on order of processing edges in "RELAX every edge" we might get only 1 finite score or actually be done.

What matters is that we get final score of a vertex on shortest path to target & in fact we extend a chain of such vertices.
$i = 2$

Order of relaxing?

$x \ldots a \ldots b \ldots c$

$y \ldots c$

$z \ldots b$
i = 3

we have found shortest path to u ahead of schedule
$i = 5, 6, 7$

redundant
BELLMAN-FORD ALGORITHM

Works for negative weights & can detect negative cycles.

(scores will still be changing after \( i = V-1 \))

(see CLRS)