18.2.5
Variants on Bellman-Ford

Finding the Paths and a Shortest Path Tree

How do we find a shortest path tree in addition to distances?

- For each $\boldsymbol{v}$ the $\boldsymbol{d}(\boldsymbol{v})$ can only get smaller as algorithm proceeds.
- If $\boldsymbol{d}(\boldsymbol{v})$ becomes smaller it is because we found a vertex $\boldsymbol{u}$ such that $\boldsymbol{d}(\boldsymbol{v}) > \boldsymbol{d}(\boldsymbol{u}) + \ell(\boldsymbol{u}, \boldsymbol{v})$ and we update $\boldsymbol{d}(\boldsymbol{v}) = \boldsymbol{d}(\boldsymbol{u}) + \ell(\boldsymbol{u}, \boldsymbol{v})$. That is, we found a shorter path to $\boldsymbol{v}$ through $\boldsymbol{u}$.
- For each $\boldsymbol{v}$ have a $\boldsymbol{prev}(\boldsymbol{v})$ pointer and update it to point to $\boldsymbol{u}$ if $\boldsymbol{v}$ finds a shorter path via $\boldsymbol{u}$.
- At end of algorithm $\boldsymbol{prev}(\boldsymbol{v})$ pointers give a shortest path tree oriented towards the source $\boldsymbol{s}$.

Negative Cycle Detection

Negative Cycle Detection

Given directed graph G with arbitrary edge lengths, does it have a negative length cycle?

1. Bellman-Ford checks whether there is a negative cycle C that is reachable from a specific vertex s. There may negative cycles not reachable from s.
2. Run Bellman-Ford $|V|$ times, once from each node u?

Negative Cycle Detection

Negative Cycle Detection

Given directed graph G with arbitrary edge lengths, does it have a negative length cycle?

1. Bellman-Ford checks whether there is a negative cycle C that is reachable from a specific vertex s. There may negative cycles not reachable from s.
2. Run Bellman-Ford $|V|$ times, once from each node u?

Negative Cycle Detection

1. Add a new node s' and connect it to all nodes of G with zero length edges. Bellman-Ford from s' will fill find a negative length cycle if there is one. Exercise: why does this work?
2. Negative cycle detection can be done with one Bellman-Ford invocation.

THE END

...

(for now)