18.2.3.1
Correctness of the Bellman-Ford Algorithm

Bellman-Ford Algorithm: Modified for analysis

for each $u \in V$ **do**
    $d(u, 0) \leftarrow \infty$
$d(s, 0) \leftarrow 0$

for $k = 1$ to n **do**
    for each $v \in V$ **do**
        $d(v, k) \leftarrow d(v, k-1)$
        for each edge $(u, v) \in in(v)$ **do**
            $d(v, k) = \min\{d(v, k), d(u, k-1) + \ell(u, v)\}$

for each $v \in V$ **do**
    $\text{dist}(s, v) \leftarrow d(v, n-1)$

Walks computed correctly

Lemma 18.3.

For each v, $d(v, k)$ is the length of a shortest walk from s to v with at most k hops.

Proof.

Standard induction (left as exercise). □

Bellman-Ford computes the shortest paths correctly

Lemma 18.4.

If G does not has a negative length cycle reachable from s $\implies$ $\forall v$:
$d(v, n) = d(v, n-1)$.

Also, $d(v, n-1)$ is the length of the shortest path between s and v.

Proof.

Shortest walk from s to reachable vertex is a path [not repeated vertex] (otherwise $\exists$ neg cycle).
A path has at most $n - 1$ edges.
$\implies$ Len shortest walk from s to v with at most $n - 1$ edges
$=$ Len shortest walk from s to v
$=$ Len shortest **path** from s to v.
By **Lemma 18.3** : $d(v, n) = d(v, n-1) = \mathrm{dist}(s, v)$, for all v. □

Bellman-Ford computes the shortest paths correctly

Lemma 18.4.

If G does not has a negative length cycle reachable from $s \implies \forall v$: $d(v, n) = d(v, n-1)$.

Also, $d(v, n-1)$ is the length of the shortest path between s and v.

Proof.

Shortest walk from s to reachable vertex is a path [not repeated vertex] (otherwise $\exists$ neg cycle).
A path has at most $n-1$ edges.
$\implies$ Len shortest walk from s to v with at most $n-1$ edges
$=$ Len shortest walk from s to v
$=$ Len shortest **path** from s to v.
By **Lemma 18.3** : $d(v, n) = d(v, n-1) = \mathrm{dist}(s, v)$, for all v. □

Bellman-Ford computes the shortest paths correctly

Lemma 18.4.

If G does not has a negative length cycle reachable from $s \implies \forall v$:
$d(v, n) = d(v, n-1)$.

Also, $d(v, n-1)$ is the length of the shortest path between s and v.

Proof.

Shortest walk from s to reachable vertex is a path [not repeated vertex] (otherwise $\exists$ neg cycle).
A path has at most $n-1$ edges.
$\implies$ Len shortest walk from s to v with at most $n-1$ edges
$=$ Len shortest walk from s to v
$=$ Len shortest **path** from s to v.
By Lemma 18.3 : $d(v, n) = d(v, n-1) = \mathrm{dist}(s, v)$, for all v. □

Bellman-Ford computes the shortest paths correctly

Lemma 18.4.

If G does not has a negative length cycle reachable from $s \implies \forall v$: $d(v, n) = d(v, n-1)$.

Also, $d(v, n-1)$ is the length of the shortest path between s and v.

Proof.

Shortest walk from s to reachable vertex is a path [not repeated vertex] (otherwise $\exists$ neg cycle).

A path has at most $n - 1$ edges.

$\implies$ Len shortest walk from s to v with at most $n - 1$ edges

$=$ Len shortest walk from s to v

$=$ Len shortest **path** from s to v.

By Lemma 18.3 : $d(v, n) = d(v, n-1) = \text{dist}(s, v)$, for all v. □

Bellman-Ford computes the shortest paths correctly

Lemma 18.4.

If G *does not has a negative length cycle reachable from* $s \implies \forall v$*:*
$d(v, n) = d(v, n-1)$*.*

Also, $d(v, n-1)$ *is the length of the shortest path between* s *and* v*.*

Proof.

Shortest walk from s to reachable vertex is a path [not repeated vertex] (otherwise $\exists$ neg cycle).
A path has at most $n-1$ edges.
$\implies$ Len shortest walk from s to v with at most $n-1$ edges
$=$ Len shortest walk from s to v
$=$ Len shortest **path** from s to v.
By **Lemma 18.3** : $d(v, n) = d(v, n-1) = \text{dist}(s, v)$, for all v. □

Bellman-Ford computes the shortest paths correctly

Lemma 18.4.

If G *does not has a negative length cycle reachable from* $s \implies \forall v$*:*
$d(v, n) = d(v, n-1)$.

Also, $d(v, n-1)$ *is the length of the shortest path between* s *and* v.

Proof.

Shortest walk from s to reachable vertex is a path [not repeated vertex]
(otherwise $\exists$ neg cycle).
A path has at most $n-1$ edges.
$\implies$ Len shortest walk from s to v with at most $n-1$ edges
$=$ Len shortest walk from s to v
$=$ Len shortest **path** from s to v.
By Lemma 18.3 : $d(v, n) = d(v, n-1) = \mathrm{dist}(s, v)$, for all v. □

THE END

...

(for now)