Class Tutorial 3

1. Shortest path example
Consider the following graph:

![Graph Image]

a. Run the Bellman-Ford algorithm on the graph.

b. Run Dijkstra's algorithm on the graph.

Solution

a. Bellman-Ford Algorithm.
Input: A weighted directed graph $G$, and destination node $t$.
Initialization: $d[t] = 0$, $d[v] = \infty$ for $v \in V \setminus \{t\}$, $\pi[v] = \phi$ for $v \in V$

for $i = 1$ to $|V| - 1$

d'[$v$] = $d[v]$, $v \in V \setminus \{t\}$

for each vertex $v \in V \setminus \{t\}$,

compute $q[v] = \min_u \{w(v,u) + d'[u] : (v,u) \in E\}$

if $q[v] < d[v]$,

set $d[v] = q[v]$. $\pi[v] \in \arg \min_u \{w(v,u) + d'[u] : (v,u) \in E\}$

return \{d[v], \pi[v]\}

Initialization:
Iteration 1:

Iteration 2:

Iteration 3:
b. Dijkstra's Algorithm.

**Input:** A weighted directed graph, and destination node $t$.

**Initialization:**
- $d[t] = 0$,
- $d[v] = \infty$ for $v \in V \setminus \{t\}$,
- $\pi[v] = \phi$ for $v \in V$

Let $S = \phi$.

While $S \neq V$, (***)
- Choose $u \in V \setminus S$ with minimal value $d[u]$, add it to $S$
- For each vertex $v$ with $(v, u) \in E$,
  - if $d[v] > w(v, u) + d[u]$,
    - set $d[v] = w(v, u) + d[u]$, $\pi[v] = u$

Return $\{d[v], \pi[v]\}$
2. Dijkstra’s Algorithm - Correctness

We let \( d^*(v) \) denote the (true) shortest path length from node \( v \in V \) to the destination node \( t \), and let \( d(v) \) denote the value of node \( v \) during execution on Dijkstra’s algorithm. Prove the following properties:

a. Triangle inequality: for any edge \((u,v) \in E\) we have \( d^*(u) \leq w(u,v) + d^*(v) \).

b. Upper bound property: for all \( v \in V \) and at any time in the execution of the algorithm, we have \( d(v) \geq d^*(v) \). Moreover, once equality is obtained, \( d(v) \) never changes.

c. Correctness of Dijkstra’s algorithm: for a graph with non-negative weights, Dijkstra’s algorithm terminates with \( d(v) = d^*(v) \) for all \( v \in V \).
Solution

a. Suppose \( p \) is a shortest path from \( u \) to \( t \). Then \( p \) has no more weight than any other path from \( u \) to \( t \), specifically the path that goes from \( u \) to \( v \) and then continues optimally.

b. We prove by induction on the number of relaxation steps, i.e., number of executions of

\[
\text{if } d[v] > w(v,u) + d[u], \\
\text{set } d[v] = w(v,u) + d[u]
\]

For the first step this is clearly true, due to the initialization procedure.

Assume \( d(x) \leq d^*(x) \) for all \( x \in V \) prior to the relaxation step, and consider a relaxation of edge \((v,u)\). The only value that may change is \( d(v) \). If it changes we have

\[
\begin{align*}
d(v) &= w(v,u) + d(u) \\
&\geq w(v,u) + d^*(u) \\
&\geq d^*(v)
\end{align*}
\]

Where the first inequality is by the induction hypothesis, and the second by the triangle inequality. Thus, the induction invariant is maintained.

Once \( d(v) = d^*(v) \), it cannot decrease as we have now shown, and it cannot increase since relaxations only decrease values.

c. We prove the following loop invariant:

At the start of each while loop (**), we have \( d(v) = d^*(v) \) for all \( v \in S \).

It suffices to show that for all \( u \in V \) we have \( d(u) = d^*(u) \) when \( u \) is added to \( S \). By the upper-bound property, it will never change afterwards.

**Initialization:** initially \( S = \emptyset \) so the invariant is trivially true.

**Maintenance:** For the purpose of contradiction, let \( u \) be the first node added to the set \( S \) such that \( d(u) \neq d^*(u) \).

We must have \( u \neq t \) since \( t \) is the first node added to \( S \) and \( d(t) = d^*(t) = 0 \). We also have that \( S \neq \emptyset \) just before \( u \) is added. There must be a path from \( u \) to \( t \) otherwise \( d(u) = d^*(u) = \infty \). Thus, there is a shortest path \( p \) from \( u \) to \( t \).

Prior to adding \( u \) to \( S \), \( p \) connects a node in \( V - S \) to a node in \( S \). Let \( x \) denote the last node in \( p \) such that \( x \in V - S \) and let \( y \) denote \( x \)'s successor, i.e., \( y \in S \). We can decompose \( p \) into \( u \rightarrow x \rightarrow y \rightarrow t \).

We claim that \( d(x) = d^*(x) \) when \( u \) is added to \( S \). To see this, observe that \( d(y) = d^*(y) \) since \( y \in S \) and \( u \) is the first node for which this property does not hold.
Since $x \rightarrow y \rightarrow t$ is the shortest path from $x$ to $t$, when $y$ was relaxed, we had $d(x) = w(x, y) + d^*(y) = d^*(x)$.

We now obtain the contradiction. Since $x$ appears after $u$ on the shortest path $p$, and since all weights are non-negative, we must have $d^*(x) \leq d^*(u)$. Therefore

\[
d(x) = d^*(x) \\
\leq d^*(u) \\
\leq d(u)
\]

But because $x$ and $u$ were in $V - S$ we must also have $d(u) \leq d(x)$, therefore, $d(x) = d^*(x) = d^*(u) = d(u)$, which contradicts our definition of $u$. 