Proof for Dijkstra’s Algorithm

In what follows, we denote the distance between \( s \) and \( u \), by \( \delta(u) \).

**Claim 0.1**  
In any moment where \( \lambda(u) = k \), there exist a path from \( s \) to \( u \) of length \( k \).

We prove this by induction on the time that this value \( \lambda \) was given. The first value of \( \lambda \) given is \( \lambda(s) ← 0 \), and clearly there is a path from \( s \) to \( s \) of length 0.

Now, next time line 5 takes place, \( \lambda(v) ← \lambda(u) + l(e) \), we may assume by the induction hypothesis that there is a path between \( s \) and \( u \) of length \( \lambda(u) \). Now, this path plus the edge \( u \rightarrow v \) gives a path from \( s \) to \( v \) of length \( \lambda(u) + l(e) \) which is exactly the current value of \( \lambda(v) \).

**Corollary 0.2**  
In each moment, \( \lambda(u) \geq \delta(u) \).

Our main claim is:

**Theorem 0.3**  
Whenever \( u \) is elected on line 3 of the algorithm, \( \lambda(u) = \delta(u) \)

Since \( t \) is eventually chosen, this gives the desired claim. We prove this by induction on the iteration of line 3. In the base of the induction, \( s \) is chosen and the claim is clear.

Now suppose that \( u \) is chosen in iteration \( i + 1 \). Assume for the sake of contradiction that \( \lambda(u) \neq \delta(u) \). By Corollary 0.2, it must be the case that

\[
\lambda(u) > \delta(u). \tag{1}
\]

Choose a shortest path \( P \) between \( s \) and \( u \). Let \( x \) be the rightmost vertex in the path that was chosen before in line 3 (that is, chosen before \( u \) that is now the chosen one). See next figure. We first claim that

\[
\delta(y) = \delta(x) + l(x, y) \tag{2}
\]

This is explained as follows. First assume that \( \delta(y) < \delta(x) + l(x, y) \). Let \( P_3 \) be a shortest path between \( s \) and \( y \). Let \( P_2 \) be the part after that in \( P \) namely, the path from \( y \) to \( u \). Append \( P_2 \) to \( P_3 \), and you will get a path from \( s \) to \( u \) of length smaller that \( l(P) \) (the length of \( P \)). This can not be (\( P \) is a shortest path between \( s \) and \( u \)).

Now, since there is a path from \( s \) to \( y \) of length \( \delta(x) + l(x, y) \) (indeed, this is the shortest path from \( s \) to \( x \) that goes then to \( y \) via the edge \( (x, y) \)) it can not be that \( \delta(y) > \delta(x) + l(x, y) \). Hence we have established Equation 2.

Since \( x \) is chosen before \( u \), by the induction hypothesis, we have that:

\[
\lambda(x) = \delta(x) \tag{3}
\]

Now, when \( x \) is chosen, we update in line 3 the value of \( \lambda(y) \), and thus

\[
\lambda(y) \leq \lambda(x) + l(x, y) = \delta(x) + l(x, y). \tag{4}
\]

The last equality follows from Equation 3. By Equation 2 and Corollary 0.2 we get that

\[
\lambda(y) = \delta(y) \tag{5}
\]
Now, as $y$ is not chosen yet in line 3 of the algorithm, and as we choose the vertex with minimum $\lambda$, we get that:

$$\lambda(u) \leq \lambda(y)$$

Finally, we also establish that

$$\delta(y) \leq \delta(u).$$

This is because $y$ is on the way to $u$ from $s$ in the shortest path. Also, the length of $P_2$, $l(P_2)$ is non-negative, as we deal with non-negative weights. These two remarks give Inequality 7.

Thus we finally conclude:

$$\lambda(u) \leq \lambda(y) = \delta(y) \leq \delta(u) < \lambda(u)$$

This gives $\lambda(u) < \lambda(u)$, a contradiction.