Correctness Proof of Dijkstra's Algorithm

Michiel Smid*

February 23, 2024

Let $G = (V, E)$ be a directed graph in which each edge (u, v) has a real weight $wt(u, v) \geq 0$. Let $s \in V$ be a source vertex. In these notes, we prove that Dijkstra's algorithm computes for each vertex v in V, the length $\delta(s, v)$ of a shortest directed path from s to v.

Algorithm $\textsc{Dijkstra}(G, s)$:
for each $v \in V$
do $d(v) = \infty$
endfor;
$d(s) = 0$;
$S = \emptyset$;
$Q = V$;
while $Q \neq \emptyset$
do $u =$ vertex in Q for which $d(u)$ is minimum;
  comment: we will prove below that $d(u) = \delta(s, u)$.
  delete u from Q;
  insert u into S;
  for each edge (u, v)
  do if $d(u) + wt(u, v) < d(v)$
    then $d(v) = d(u) + wt(u, v)$
    endif
  endfor
endwhile

Lemma 1 *For each vertex v in V and at any moment during the algorithm,*

$$\delta(s, v) \leq d(v).$$

Proof. The lemma follows from the fact that either $d(v) = \infty$ or $d(v)$ is equal to the length of some directed path from s to v. ∎

*School of Computer Science, Carleton University, Ottawa, Canada.

Lemma 2 *Let v be a vertex in V and assume that, at some moment, $d(v)$ becomes equal to $\delta(s, v)$. Then the value of $d(v)$ does not change afterwards.*

Proof. It follows from the algorithm that, if $d(v)$ changes, it becomes smaller. By Lemma 1, $d(v)$ cannot be smaller than $\delta(s, v)$. ∎

Lemma 3 *Let u be a vertex in V. Consider the iteration of the while-loop in which u is chosen as the vertex in Q for which $d(u)$ is minimum. At the moment when u is chosen, $d(u) = \delta(s, u)$.*

Proof. The proof is by contradiction. Consider the *first* iteration of the while-loop for which the lemma does not hold. In other words, consider the first vertex u having the property that

$$\delta(s, u) < d(u) \tag{1}$$

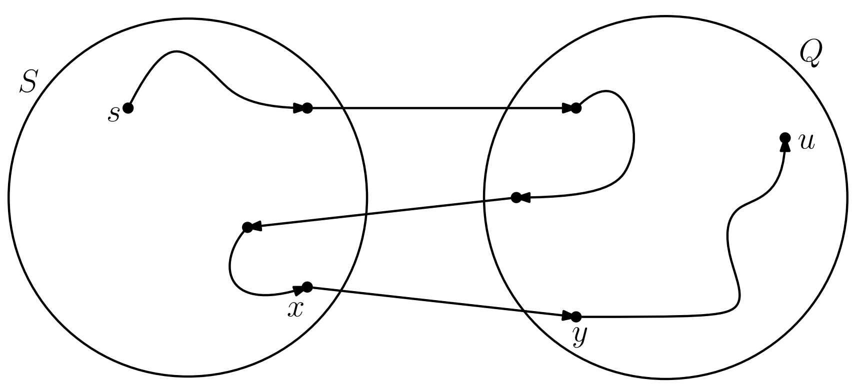
during the iteration in which u is chosen as the vertex in Q for which $d(u)$ is minimum.

Exercise: Convince yourself that $u \neq s$.

We define *time* t to be the moment when u is chosen, but before u is deleted from the set Q. At time t, the following hold:

- For every vertex z in S, $d(z) = \delta(s, z)$. This follows from the way we have chosen u and from Lemma 2.
- The source vertex s is in S.
- The vertex u is in Q.

Let P be a shortest directed path from s to u. Since, at time t, $s \in S$ and $u \in Q$, this path contains an edge, say (x, y), such that, at time t, $x \in S$ and $y \in Q$. (In fact, there may be several such edges.)



At time t, u is chosen as the vertex in Q for which $d(u)$ is minimum. Since at that time, y is in Q, we have

$$d(u) \leq d(y). \tag{2}$$

Consider the iteration in which x is chosen as the vertex in Q for which $d(x)$ is mimimum. Note that this happens before time t. It follows from the algorithm that, at the end of this iteration,

$$d(y) \leq d(x) + wt(x, y). \tag{3}$$

By Lemma 2, $d(x)$ does not change afterwards. The value of $d(y)$ may change afterwards, but if it does, it becomes smaller. Therefore, (3) still holds at time t.

Since P is a shortest path from s to u, we have

$$\delta(s, y) = \delta(s, x) + wt(x, y). \tag{4}$$

Since all edge weights are non-negative, we have

$$\delta(s, y) \leq \delta(s, u). \tag{5}$$

By combining the above inequalities, we obtain

$$\begin{array}{rcll} d(u) & \leq & d(y) & \text{(from (2))} \\ & \leq & d(x) + wt(x, y) & \text{(from (3))} \\ & = & \delta(s, x) + wt(x, y) & \text{(since } x \in S \text{ at time } t\text{)} \\ & = & \delta(s, y) & \text{(from (4))} \\ & \leq & \delta(s, u) & \text{(from (5))} \\ & < & d(u). & \text{(from (1))} \end{array}$$

Thus, we have shown that $d(u) < d(u)$, which is a contradiction. ∎