A proof that BFS finds shortest paths

In class, I presented a proof that BFS actually computes shortest paths, a fact that is intuitively “obvious,” and yet, a careful proof takes a bit of work. This handout gives the same proof, with some of the details expanded, and is provided for your reference.

First, some notation. There are $n$ vertices in the graph, numbered $1 \ldots n$. The BFS starts at vertex $r$, which forms the root of the BFS tree, and a total of $\ell$ vertices are reachable from $r$ (and hence visited during BFS). For a vertex $v$, we define

- $\text{level}[v]$ to be the level of $v$ in the BFS tree,
- $\text{dist}[v]$ to be the actual distance from $r$ to $v$ in the graph, and
- $\text{pos}[v]$ to be the “position number” $i$ of $v$, where $1 \leq i \leq \ell$, and $v$ was the $i$th vertex to be inserted into the queue during BFS.

We want to show that $\text{level}[v] = \text{dist}[v]$ for all $v$. We do this by induction on $\text{pos}[v]$, and strengthen the induction hypothesis. Namely, we prove by induction on $i = 1 \ldots \ell$ that for $v$ with $\text{pos}[v] = i$, we have

(\textbf{I1}) $\text{dist}[v] = \text{level}[v]$, and

(\textbf{I2}) for any vertex $w$, if $\text{dist}[w] < \text{dist}[v]$, then $\text{pos}[w] < \text{pos}[v]$.

The case $i = 1$ is trivially true, as the reader may verify. We now prove (I1) and (I2) for $1 < i \leq \ell$, assuming (I1) and (I2) hold for all $i' < i$.

To prove (I1) for $i$, let $\text{pos}[v] = i$, let $v'$ be the parent of $v$ in the BFS tree. Suppose, by way of contradiction, that there is a path

$$r \sim w \rightarrow v$$

of length $h < \text{level}[v]$. First, we have

$$\text{pos}[v'] < \text{pos}[v], \quad (1)$$

since $v$ is placed in the queue when $v'$ is removed from the queue. Second, we have

$$\text{dist}[w] < \text{dist}[v'], \quad (2)$$

since by (1) we may apply the induction hypothesis (I1) at $\text{pos}[v']$, obtaining

$$\text{dist}[v'] = \text{level}[v'] = \text{level}[v] - 1 > h - 1 \geq \text{dist}[w']. \quad (3)$$
Third, we have
\[ \text{pos}[w] < \text{pos}[v'], \]  
(3) since by (1) we may apply the induction hypothesis (I2) at \text{pos}[v'], together with (2), to obtain (3).

But now consider the point in time during the execution of BFS when \( w \) was removed from the queue. Since there is an edge \( w \rightarrow v \), the BFS algorithm would visit \( v \) at this point in time, if it had not already at an earlier time. Thus, the parent of \( v \) has position number at most \text{pos}[w], which by (3) is strictly less than \text{pos}[v], and so \( v' \) cannot be the parent of \( v \), as assumed — a contradiction.

To prove (I2), suppose \( \text{pos}[v] = i \) and
\[ \text{dist}[w] < \text{dist}[v]. \]  
(4)
By way of contradiction, assume that \( \text{pos}[w] \geq \text{pos}[v] \). Since by (4), \( w \neq v \), it follows that \( \text{pos}[w] \neq \text{pos}[v] \), and hence
\[ \text{pos}[w] > \text{pos}[v]. \]  
(5)
Let \( v' \) the parent of \( v \) in the BFS tree, and let
\[ r \rightsquigarrow w' \rightarrow w \]
be a shortest path from \( r \) to \( w \). This implies that \( r \rightsquigarrow w' \) is a shortest path from \( r \) to \( w' \). From this, we may conclude that
\[ \text{dist}[w'] = \text{dist}[w] - 1. \]  
(6)
First, just as in the proof of (I1), we have
\[ \text{pos}[v'] < \text{pos}[v]. \]  
(7)
Second, by hypothesis (I1) at \( \text{pos}[v] \), which we just proved above, the tree path \( r \rightsquigarrow v' \rightarrow v \) is a shortest path, and hence the tree path \( r \rightsquigarrow v' \) is also a shortest path. From this, we may conclude that
\[ \text{dist}[v'] = \text{dist}[v] - 1. \]  
(8)
Now, (4), together with (6) and (8), imply that
\[ \text{dist}[w'] < \text{dist}[v']. \]  
(9)
Applying the induction hypothesis (I2) at $pos[v']$, which by (7) is less than $pos[v]$, along with (9), we obtain

$$pos[w'] < pos[v']. \tag{10}$$

Now, $v$ was placed in the queue when $v'$ was removed from the queue, and $w$ was placed in the queue when $w'$ was removed or at some earlier time. By (10), this means that $w$ would be placed in the queue before $v$ was placed in the queue, which contradicts (5).