Breadth First Search II

0.1 Notation

Let's call the graph we consider G and the source vertex s. The distance in G from s to a vertex v is denoted $\delta(s, v)$ and the distance found by BFS is $d[v]$
So BFS is correct if $d[v] = \delta(s, v)$ for every vertex v in G
Let us first show that the d values found cannot be too small

0.2 Lemma 1

When BFS terminates, for each vertex v we have $d[v] \geqslant \delta(s, v)$

0.2.1 Proof

We use induction on the number of vertices added to the queue
The **base case** is when the first vertex s is added to the queue. Clearly $d[s] = 0 = \delta(s, s)$
Now suppose a vertex v is being added to the queue. This means that v has just been discovered from some other vertex u and that as v is added to the queue $d[v]$ is set to be $d[u] + 1$
By induction, we know that $d[u] \geqslant \delta(s, u)$ so we have

$$d[v] = d[u] + 1 \geq \delta(s, u) + 1 \geq \delta(s, v)$$

Where the last inequality follows from the fact that a shortest path to u can be extended to a shortest path to v by adding the edge from u to v (so the distance to v is at most one greater than the distance to u - it might, of course, be less if there is an alternative path to v and this is why we cannot replace the inequality in the lemma by an equals sign and use the same proof). The lemma is proved

Now we know that the value of $d[v]$ cannot be smaller than it should be, we have to think about how to prove it cannot be too big. To do this, we first think about what we can say about the values in the array d if we know the order in which vertices enter the queue

0.3 Lemma 2

In u is enqueued before v then $d[u] \leqslant d[v]$

0.3.1 Proof

In fact, we shall prove the following claim

If a is at the head of the queue and b is at the tail, then $d[b]$ is either $d[a]$ or $d[a] + 1$, and d values of the vertices as you go along the queue never decrease

Notice what this is saying: either $d[w]$ is the same for every vertex in the queue or the first so many vertices have one d value and the ones behind have d value one greater. Also not that the claim implies the lemma: if v enters the queue later than u then $d[v]$ must be at least $d[u]$

We prove this claim by using induction on the number of vertices added to the queue:
The **base case** - when the source is first added to the queue - is clearly true.
Now suppose a vertex w is added to the queue. This happens when a vertex x is removed from the front of the queue and w is a neighbour of x. Let y be the vertex at the tail of the queue at the moment w is added.
By the inductive hypothesis, $d[y]$ is $d[x]$ or $d[x] + 1$ and we set $d[w]$ to be $d[x] + 1$ so the claim remains true (since w is given a d value at least as large as the vertex in front of it and no more than one more than the vertex at the front of the queue).
The lemma is proved

0.4 Theorem 1

When BFS terminates, for each vertex v we have $d[v] = \delta(s, v)$

0.4.1   Proof

Let us assume the Theorem is not true. Then by lemma 1, we have $d[v] > \delta(s, v)$ for some vertex v. Let us say that v is the vertex closest to the source s for which this is true.

Consider the shortest path from s to v. Let u be the penultimate vertex on that shortest path. So $\delta(s, v) = \delta(s, u) + 1$. Because u is nearer to s than v, we have $d[u] = \delta(s, u)$ (by the way that we chose v. So)

$$d[v] > \delta(s, v) = \delta(s, u) + 1 = d[u] + 1$$

That is $d[v]$ is at least $d[u] + 2$. But think about what happens when u is dequeued

- If v was already dequeued, when $d[v] \leqslant d[u]$ (by Lemma 2); a contradiction

- If v is in the queue, then it was added, when some vertex w was dequeued and $d[v]$ was assigned the value $d[w] + 1$. But $d[w] \leqslant d[u]$ (by Lemma 2 again, since w must have been added ahead of u in the queue) so $d[v] \leqslant d[u] + 1$; again, a contradiction

- Finally if v is not yet in the queue, then when u is discovered and $d[v]$ is given the value $d[u] + 1$. This last contradiction proved the Theorem