Proof by Induction: Dijkstra’s Algorithm---

**The hypothesis:**

For each vertex that has been visited, u, in the set of visited vertices S, distances[u] is the minimum distance between the vertex u and the target vertex t.

**Proof:**

When the size of S is 1, |S| = 1, the hypothesis does not apply and is therefore trivial.

Now, assume the hypothesis is correct for |S| = n, where n > 1. Then choose the edge u-v where distances[v] is the least of any vertex not in S and the edge u-v is such that distances[u] = distances[v] + length[u,v]. distances[u] must be the shortest path from u to t by our conditions. Now suppose there is a vertex w not in S, such that w provides a path to t from u, where distances[w] < distances[v]. In this case there is a contradiction, so distances[v] must be the shortest path connecting u to t. Suppose there is a path from u to t without the use of any unvisited nodes. Then distances[u] would be less that distances[v] + length[u,v], in the case that distances[v] is the least of any vertex not in S. In either case, there is a contradiction based on our hypothesis.

Therefore, after processing u, it will still be true that for each vertex w not in S, distances[w] is the shortest path from w to t using vertices only in S because if there was a shorter path which does not visit u, then it would have been found previously and if there is a shorter path involving u then it will simply be updated when processing u.

Now, since the hypothesis is true for |S| = n, take |S| = k = n+1. There would simply be one more vertex to process and in-so-doing, the conclusion for |S| = n would still hold. Therefore, for any graph S, where |S| > n, the hypothesis will still hold.