We will solve the more general problem of computing the number of shortest paths from v to every other node.

We perform BFS from v, obtaining a set of layers $L_0, L_1, L_2, \ldots$, where $L_0 = \{v\}$. By the definition of BFS, a path from v to a node x is a shortest v-x path if and only if the layer numbers of the nodes on the path increase by exactly one in each step.

We use this observation to compute the number of shortest paths from v to each other node x. Let $S(x)$ denote this number for a node x. For each node x in L_1, we have $S(x) = 1$, since the only shortest-path consists of the single edge from v to x. Now consider a node y in layer L_j, for $j > 1$. The shortest v-y paths all have the following form; they are a shortest path to some node x in layer L_{j-1}, and then they take one more step to get to y. Thus, $S(y)$ is the sum of $S(x)$ over all nodes x in layer L_{j-1} with an edge to y.

After performing BFS, we can thus compute all these values in order of the layers; the time spent to compute a given $S(y)$ is at most the degree of y (since at most this many terms figure in the sum from the previous paragraph). Since we have seen that the sum of the degrees in a graph is $O(m)$, this gives an overall running time of $O(m + n)$.

[1] ex427.691.966