Algorithm to find out all shortest paths from source vertex 's' to a dest. vertex 't' in an unweighted graph.

for each \( v \in V \), \{ \( \text{dist}(v) = \infty \); \( \text{paths}(v) = " " \); \( \text{numpaths}(v) = 0 \); \( \text{visited}(v) = \text{NO} \); \}

\( \text{numpaths} = \) array containing the number of shortest paths

\( \text{dist}(s) = 0 \); \( \text{visited}(s) = \text{VISITED} \);

add\( Q(s) \);

while (!empty \( Q() \))

\( \{ \)

\( w = \) delete\( Q() \);

for (each neighbor \( x \) of \( w \))

\( \{ \)

\( \text{if} ( \text{visited}(x) = = \text{NO} ) \)

\( \{ \)

\( \text{dist}(x) = \text{dist}(w) + 1 \);

\( \text{numpaths}(x) = \text{numpaths}(w) + 1 \);

\( \text{paths}(\text{numpaths}(x)) = [\text{paths}(w), x] \);

\( \text{visited}(x) = \text{YES} \);

add\( Q(x) \);

\( \} \)

\( \} \)

else

\( \{ \)

\( \text{if} ( \frac{\text{dist}(w) + 1}{\text{dist}(x)} \)