Input: A graph \( G(V,E) \) with source node \( start \) and goal node \( end \).

Output: Least cost path from \( start \) to \( end \).

Steps:

Initialise

\[
\text{open\_list} = \{ \text{start} \} \quad \text{/* List of nodes to be traversed*/}
\]

\[
\text{closed\_list} = \{ \} \quad \text{/* List of already traversed nodes*/}
\]

\[
g(\text{start}) = 0 \quad \text{/* Cost from source node to a node*/}
\]

\[
h(\text{start}) = \text{heuristic\_function} (\text{start, end}) \quad \text{/* Estimated cost from node to goal node*/}
\]

\[
f(\text{start}) = g(\text{start}) + h(\text{start}) \quad \text{/* Total cost from source to goal node*/}
\]

while \( \text{open\_list} \) is not empty

\[
m = \text{Node on top of open\_list}, \text{ with least } f
\]

if \( m = = \text{end} \)

    return

remove \( m \) from \( \text{open\_list} \)

add \( m \) to \( \text{closed\_list} \)

for each \( n \) in \( \text{child}(m) \)

    if \( n \) in \( \text{closed\_list} \)

        continue

    \[
    \text{cost} = g(m) + \text{distance}(m,n)
    \]

    if \( n \) in \( \text{open\_list} \) and \( \text{cost} < g(n) \)

        remove \( n \) from \( \text{open\_list} \) as new path is better

    if \( n \) in \( \text{closed\_list} \) and \( \text{cost} < g(n) \)

        remove \( n \) from \( \text{closed\_list} \)

    if \( n \) not in \( \text{open\_list} \) and \( n \) not in \( \text{closed\_list} \)

        add \( n \) to \( \text{open\_list} \)

        \[
g(n) = \text{cost}
        \]

        \[
h(n) = \text{heuristic\_function}(n, \text{end})
        \]

        \[
f(n) = g(n) + h(n)
        \]

return failure