```
Initialise an empty queue, q
Initialise an n × n 2D distance array with ∞ everywhere
Inisialise an n × n 2D parent array with (−2, −2) everywhere
Enqueue (source_r, source_col) to q
while q is not empty and the goal is not visited do
    curr ← front of q
    dequeue from q
    foreach neighbour ∈ {down, left, up, right} do
        if neighbour is a valid move and is unvisited then
            distance[neighbour] ← distance[curr] + 1
            parent[neighbour] ← curr
            enqueue neighbour to q;
        end
    end
end
if q is empty and we haven't visited the goal then
    return No Path
else
    Initialise output list
    curr ← goal
    while curr ≠ start do
        add curr to the front of output
        curr ← parent[curr]
    end
    add start to the front of output
    return output
end
```