

Find shortest path algorithm

get data

generate a list of all node candidates

set the initial node (start node) path costs to zero
set the initial node (start node) path list to empty

while length of candidates list > 0

try to pop the node with the minimum path costs from the candidates list

id of popped min candidate is equal to the id of the end node

T

F

append end node id to the path of min candidate

∅

break

append the min candidate to the popped list

the id of min candidate is not in the path of the min candidate

T

F

append the id of the min candidate to its path list

set the path to "NO PATH"

set the path costs to infinity

break

try to assign (update) the refreshed path of min candidate in the list of candidates

foreach key in the adjacency of the min candidate

try to check if the key is not in the list of popped nodes

T

F

call updateNode(startNode, nextNode)

generate a sub dictionary from the start node

try to check if the path costs of the next node in the candidates list are higher than the path costs of the start node in the popped list plus the current edge costs

T

F

∅

assign the lower costs to the path costs of the next node in the candidates list

∅

assign the corresponding path with the lower costs to the next node in the candidates list