**Pathfinder algorithm pseudocode**

* **Each destination needs a set of neighbours that it can travel to**
* **Function to calculate min distance from node to neighbours (which is the distance thing we already have)**

Take in user specs

Clear all lists

V =single destination

Set their distance to unknown/infinity

Set previous node/optimal path to undefined

Set distance of source to source = 0

Q = Set of all unvisited destinations

While there are unvisited places, find the minimum distance u from the source to that unvisited place, and the total distance from the root to that node.

Remove that visited node from Q

Once all of the nodes are removed from Q (check if Q empty), walk the list to find minimum distance

If the distance to the node is the minimum and the node is not the target, use this node as the new/temp source and repeat.

S is a set of the best journey

U is the total minimum distance from source to end

1. Create required sets
   1. Set of unvisited nodes
   2. Set of visited nodes
   3. Empty set of the journey route
   4. Set of distances of root to node
2. Set first item in journey set to be source
3. Find smallest distance from the start to the end
   1. Calculate and set distance from source for all unvisited nodes, return minimum distance
   2. Remove node of minimum distance from unvisited set
   3. Add node of minimum distance to visited set and journey set
   4. For that node, get distance from it to all unvisited nodes, return minimum distance
   5. Remove node of minimum distance from unvisited set
   6. Add node of minimum distance to visited set
   7. Check total distance so far against distance from source to the latest visited node
   8. If distance from source is less than total distance then make next node in the journey set be the latest visited node, and remove previous visited node from journey set
   9. Repeat until the latest visited node is the final destination