**1-page documentation of approach**

I rapidly understood that an A\* search was the ideal strategy for locating a viable route connecting the two city pairs that might also include their optimality. I began by utilising a buffer read to import the CSV file database. This is due to the fact that it keeps the lines in tiny buffer arrays, making it simpler to call when required. I continued on to the A\* search algorithm after that. I was unable to complete it so I sought assistance from another github project, to which I sent an email to the user requesting permission.

The A\* Search Algorithm selects the node at each stage based on a value, "f," which is a parameter equal to the sum of two additional factors, "g" and "h." It chooses the node or cell with the lowest "f" at each step and processes that node or cell.

g is the cost in movement to go along a path created to get from a starting point to a particular city. h is the calculated cost of moving from a specific grid square to the desired location. This is frequently referred to as the heuristic, which is just another word for an educated guess.

After that, I optimised the haversine formula and then I made all outputs write into a separate file.