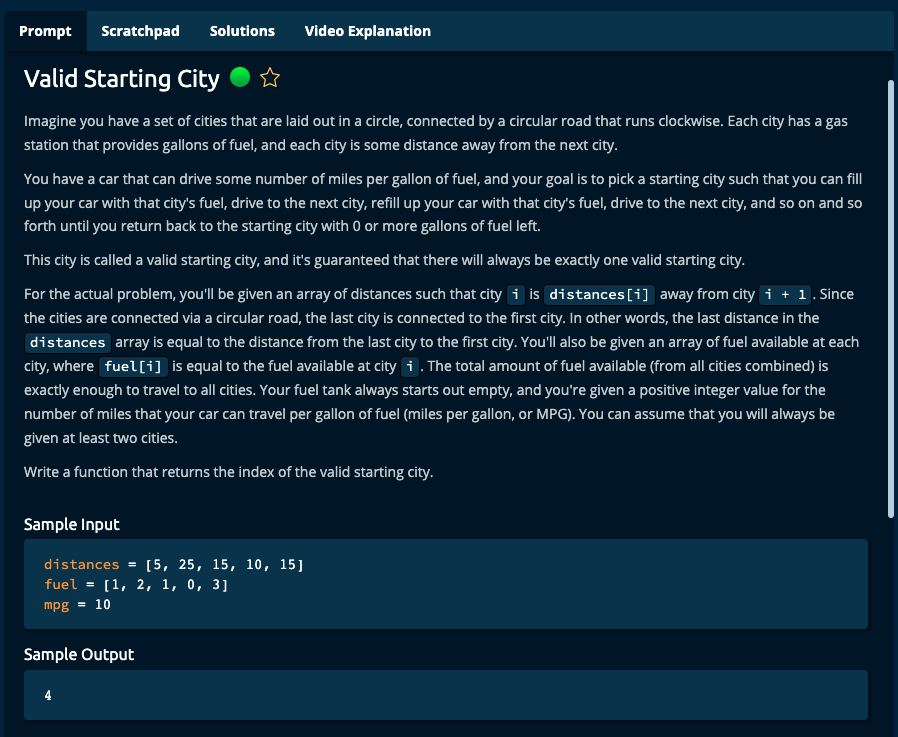
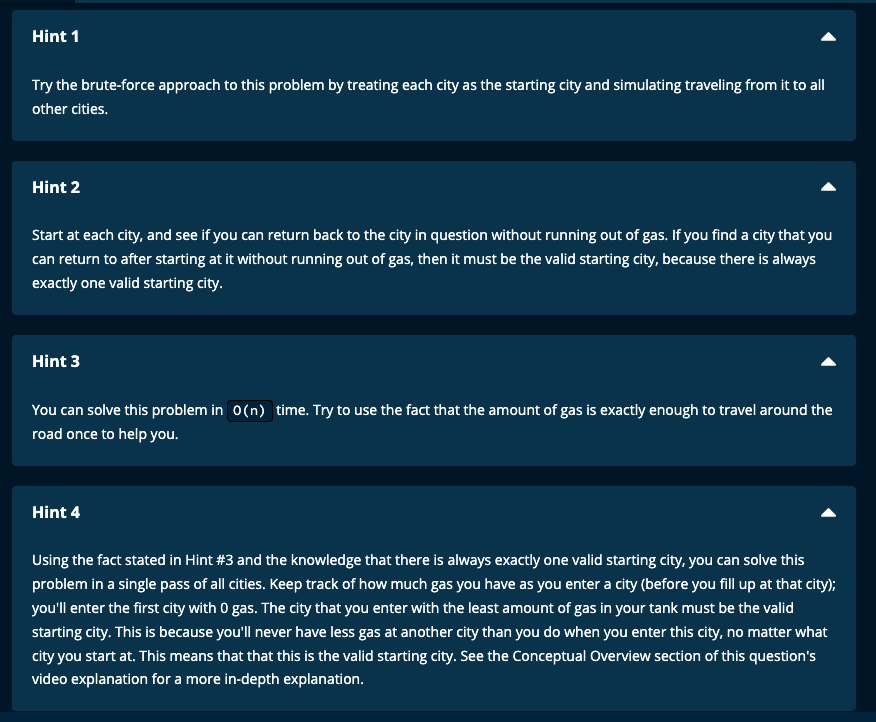
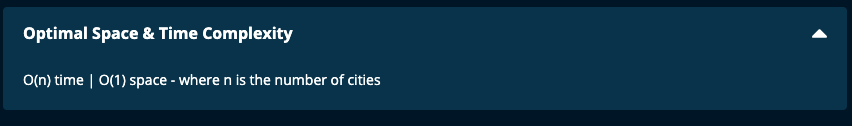
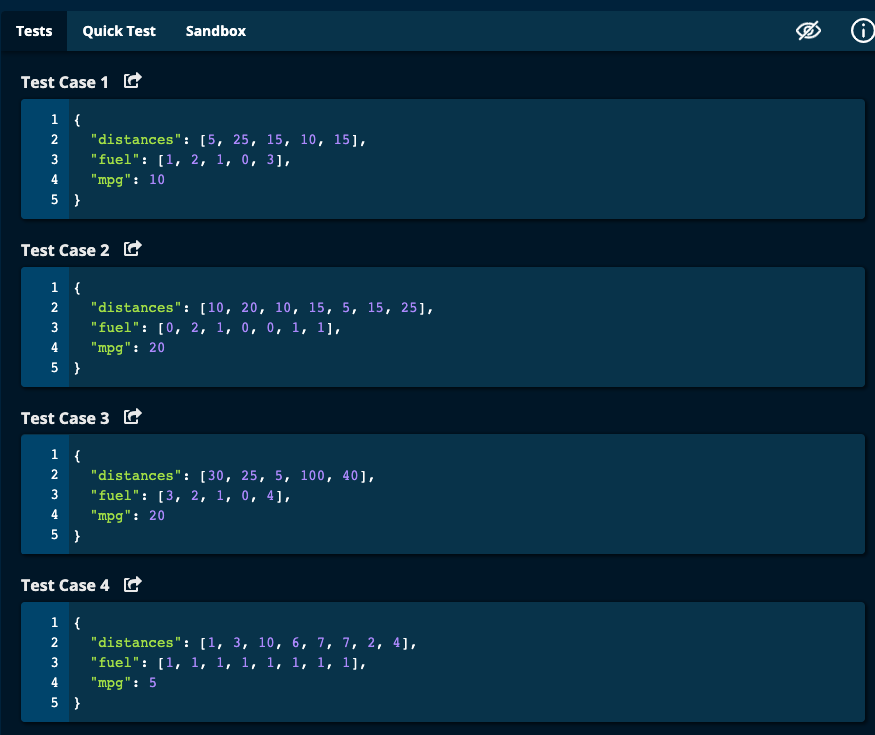
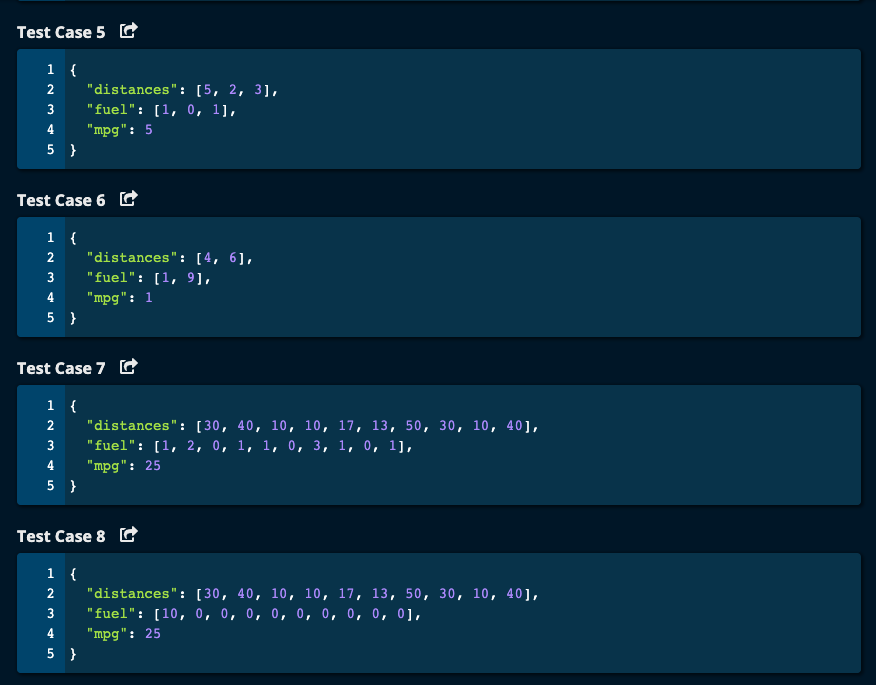
Valid Starting City (Medium)

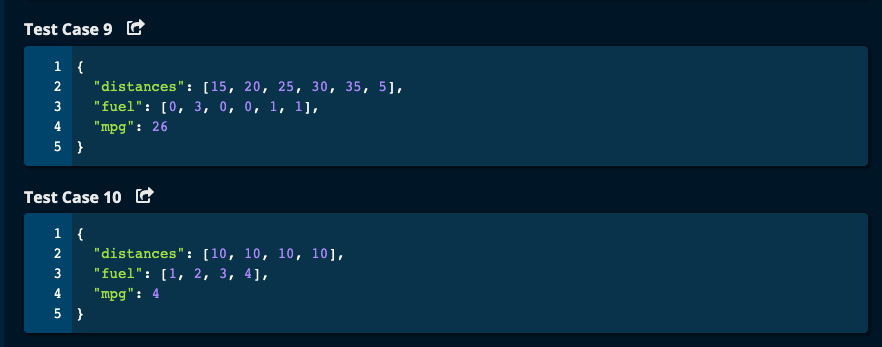












My Solution:

def validStartingCity(distances, fuel, mpg):

minMilesLeft = 0

minMilesIndex = 0

milesLeft = 0

for i in range(1, len(distances)):

milesLeft += (fuel[i - 1] \* mpg) - distances[i - 1]

print("i = ", i, "distances[i - 1] = ", distances[i - 1], "fuel[i - 1] = ", fuel[i - 1], "milesLeft = ", milesLeft)

if milesLeft < minMilesLeft:

minMilesLeft = milesLeft

minMilesIndex = i

return minMilesIndex

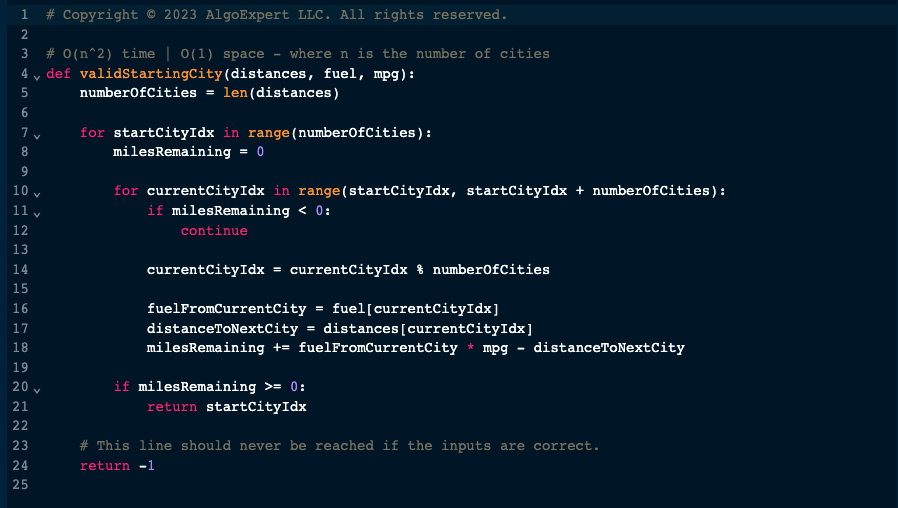
JJ Notes:

1. Initialize milesLeft (i.e. miles remaining), minMilesLeft and minMilesIndex to 0.
2. Iterate through the distances array from index 1.
3. Miles remaining will be whatever is in milesLeft and product of how far you can go which is the difference of fuel \* mpg from the previous city and the distance already traveled to the current city.
4. If milesLeft is less than minMilesLeft, then update minMilesLeft with milesLeft. Then minMilesIndex with be the current index.
5. Return the current index.
6. Time Complexity is O(n) as we iterate through the array once. Space Complexity = O(1) since we have a few variables that we update.

Algoexpert Soltuions:

Solution 1: Brute Force:

Start at each city and see which one is feasible.



Solution 2:

