Thi Nguyen

ririchiroyo@csu.fullerton.edu

Algorithm 2: Greedy Approach to Hamiltonian Problem

Pseudocode:

Problem: finding the best starting city to make a round trip given distance, mpg, and fuel

Input: distance (vec int), fuel (vec int), mpg (int)

Output: best starting city

Constraints: valid ints (no negatives), total fuel from all cities must cover the journey, must be a circular route

```
FUNCTION findStartingCity(distance, fuel, mpg)
Int citycount = distance.size
Int fuelbalance = 0

For I from 0 to n-1:
    Calculate total fuel for one round trip
    If fuelbalance < 0
        Start with the next city
        Reset fuelbalance to 0
Return starting city
```

Time Complexity:

Code	**Time Complexity**
Int citycount = distance.size	O(1)
Int fuelBalance = 0	O(1)
For 1 from 0 n-1	O(n+1)
Calculate	O(n)

If fuelBalance < 0
Start with next city
fuelBalance = 0
	O(n)
	O(1)
	O(1)

Return starting city	O(1)

Worst	O(n)

Proving with Limits:

$F(n) = 1 + 1 + (n + 1) + n + n + 1 + 1 + 1 = 3n + 6$

$F(n) = 3n + 6 \qquad G(n) = n$

$\lim n \rightarrow \infty \; F(n) / G(n)$

$\lim n \rightarrow \infty \; 3n + 6 / n$

$\lim n \rightarrow \infty \; (3n/n) + (6/n)$

$\lim n \rightarrow \infty \; 3 + (6/n)$

$3 + \lim n \rightarrow \infty \; 6/n$

$3 + 6/\infty$

$3 + 0 = 3$

$\theta(n)$

Overall worst time complexity is: θ(n)