

Algorithm 2: Greedy Approach to Hamiltonian Problem

Group 9:

Xander Lahti (xanderla@csu.fullerton.edu)

Xareni Merino Rita (xmerino@csu.fullerton.edu)

T e Yen Lee (leeteyen91@csu.fullerton.edu)

Juan Martinez Vasquez (diegomarvas@csu.fullerton.edu)

Algorithm 2: Greedy Approach to Hamiltonian Problem

Pseudocode

```
Function whichCityToStart( ListCityDistances, ListFuel, mpg) {  
    TotalFuel = 0;  
    CurrentFuel = 0;  
    StartingCity = 0;  
    For currentCity in Size(ListCityDistances)  
        CurrentFuel = (fuel[currentCity] * mpg) - ListCityDistances[currentCity]  
        TotalFuel = (fuel[currentCity] * mpg) - ListCityDistances[currentCity]  
        If CurrentFuel < 0;  
            StartingCity = i + 1;  
            CurrentFuel = 0  
    If TotalFuel >= 0  
        Return StartingCity  
    Else  
        Return none
```

Algorithm 2: Greedy Approach to Hamiltonian Problem

Proven Efficiency of the pseudocode “Step Counts”

```
Function whichCityToStart( ListCityDistances, ListFuel, mpg) {  
    TotalFuel = 0; O(1)  
    CurrentFuel = 0; O(1)  
    StartingCity = 0; O(1)  
    For currentCity in Size(ListCityDistances) // n = size O(n)  
        CurrentFuel = (fuel[currentCity] * mpg) - ListCityDistances[currentCity] O(1)  
        TotalFuel = (fuel[currentCity] * mpg) - ListCityDistances[currentCity] O(1)  
        If CurrentFuel < 0; O(1)  
            BestCity = i + 1; O(1)  
            CurrentFuel = 0 O(1)  
        If TotalFuel >= 0 O(1)  
            Return StartingCity O(1)  
        Else O(1)  
            Return none O(1)
```

$$O(1)+O(1)+O(1)+O(n)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1)+O(1) = O(n)$$

The time complexity for this Algorithm is $O(n)$