Salt and Pepper

Case Study 1: Solving Real-World Problems using Computational Thinking

Maximizing Delivery Performance Through Optimal Storage and Routing

Members:
Pornobe, Reuel Christian, M.
Rodriguez, Kyle Franz, P.

Salt and Pepper

Problem:

PACKAGES BEING DELIVERED ARE SOMETIMES DELAYED.

Iteration 1

Problem Identification:

How many vehicles are necessary to carry the packages to be delivered?

Decomposition

- Identify the carry weight of each transportation
- Identify the weight of each package
- Identify if the package can still be carried by a vehicle, else put it into another vehicle

Pattern Recognition

- All packages are arranged in descending order by weight
- Each package (its whole weight) is sent to a ground transport that it can fit into.

Abstraction

- A list of packages with their weight
- Weight capacity of the vehicle

Iteration 2

Problem Identification:

What is the optimal route of each ground transport chosen?

Decomposition

- Identify the items stored in each vehicle.
- Identify the distance to travel from start to nearest package
- Acquire the optimal route

Pattern Recognition

- The shortest route for each node will be calculated using Dijkstra’s Algorithm.

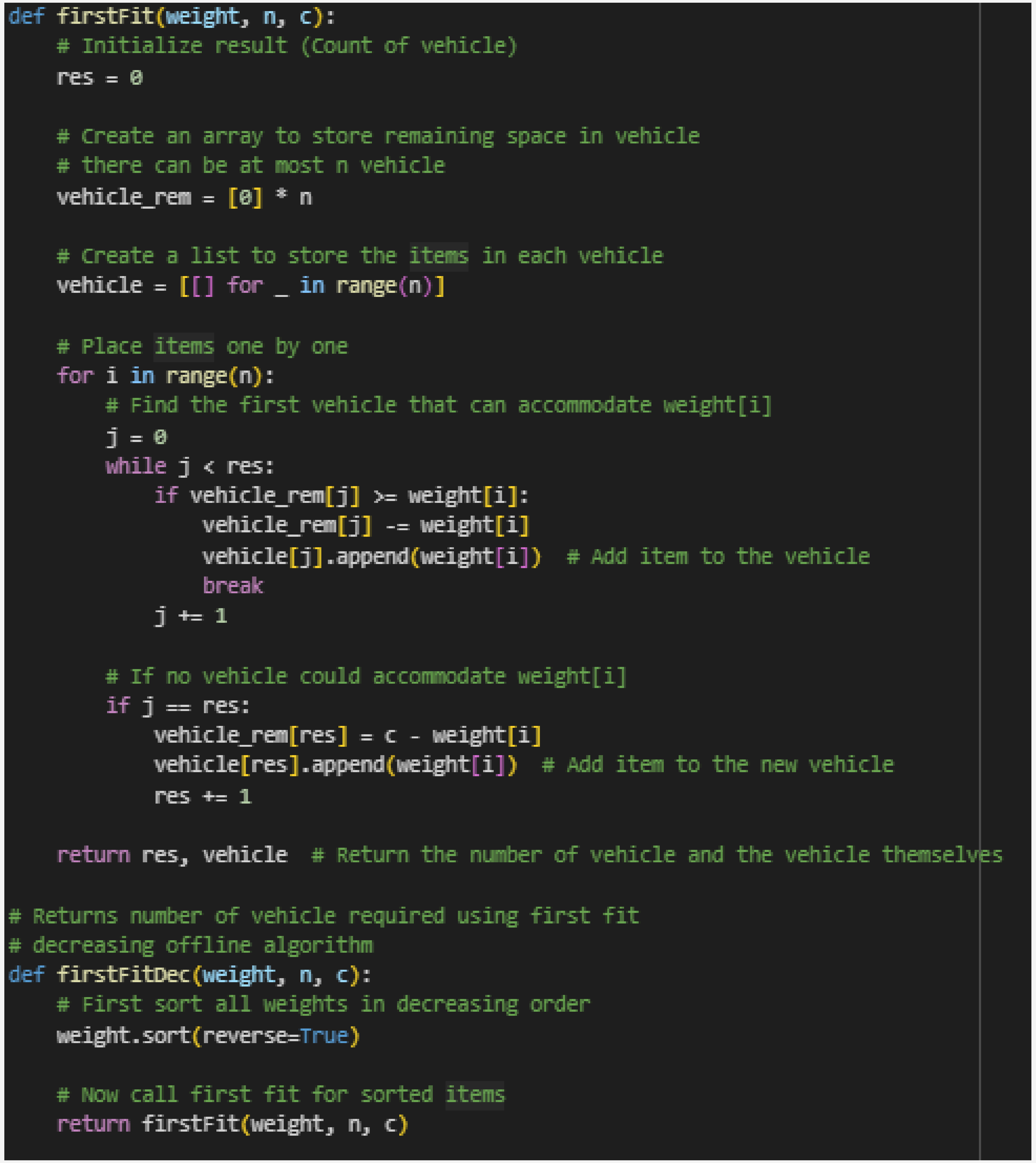
Abstraction

- List of the location of each package.

Code:

Iteration 1

How many vehicles are necessary to carry the packages to be delivered?

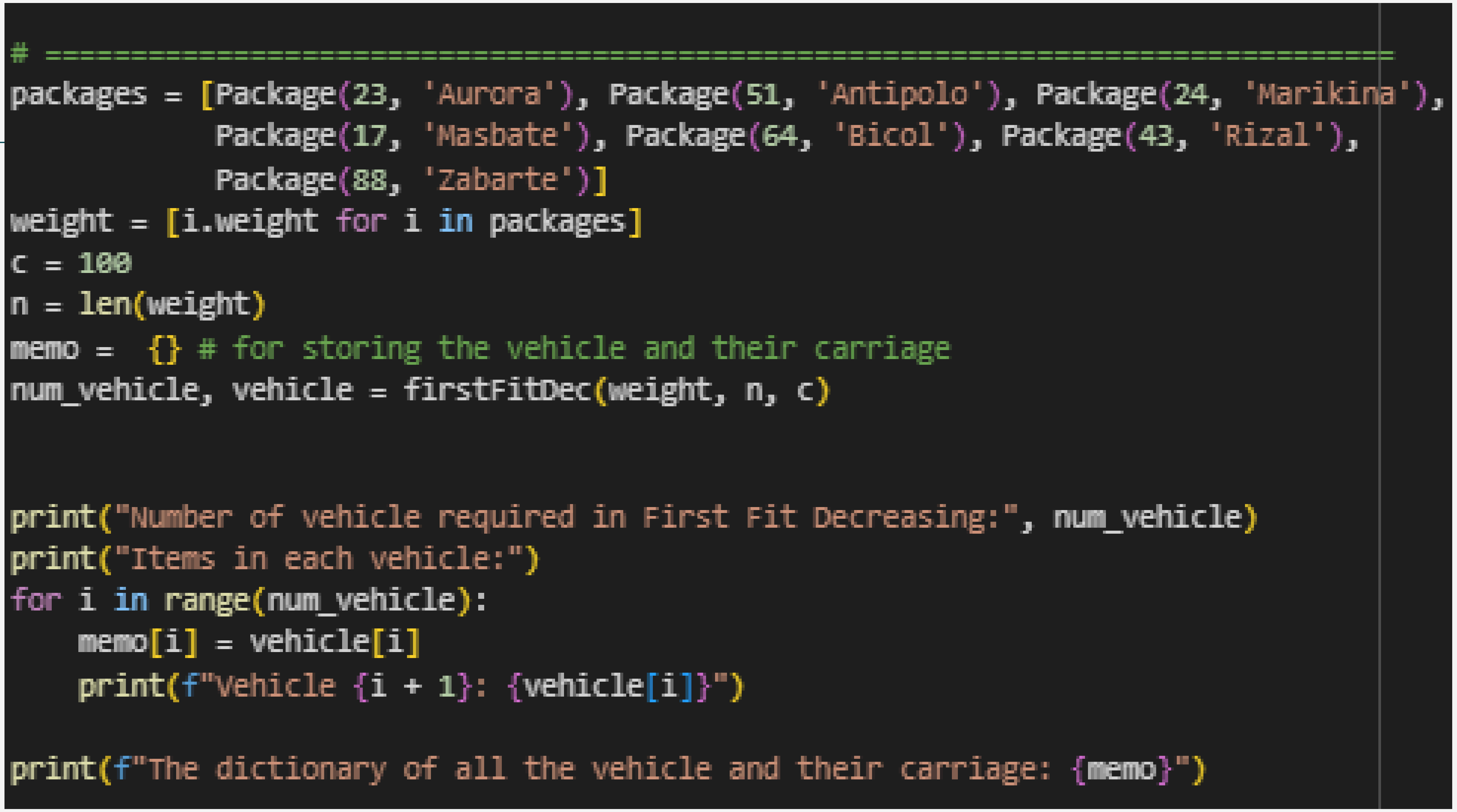


```python
def firstFit(weight, n, c):
    # Initialize result (Count of vehicle)
    res = 0

    # Create an array to store remaining space in vehicle
    # there can be at most n vehicle
    vehicle_rem = [0] * n

    # Create a list to store the items in each vehicle
    vehicle = [[] for _ in range(n)]

    # Place items one by one
    for i in range(n):
        # Find the first vehicle that can accommodate weight[i]
        j = 0
        while j < res:
            if vehicle_rem[j] >= weight[i]:
                vehicle_rem[j] -= weight[i]
                vehicle[j].append(weight[i])  # Add item to the vehicle
                break
            j += 1

        # If no vehicle could accommodate weight[i]
        if j == res:
            vehicle_rem[res] = c - weight[i]
            vehicle[res].append(weight[i])  # Add item to the new vehicle
            res += 1

    return res, vehicle  # Return the number of vehicle and the vehicle themselves

# Returns number of vehicle required using first fit
# decreasing offline algorithm
def firstFitDec(weight, n, c):
    # First sort all weights in decreasing order
    weight.sort(reverse=True)

    # Now call first fit for sorted items
    return firstFit(weight, n, c)
```

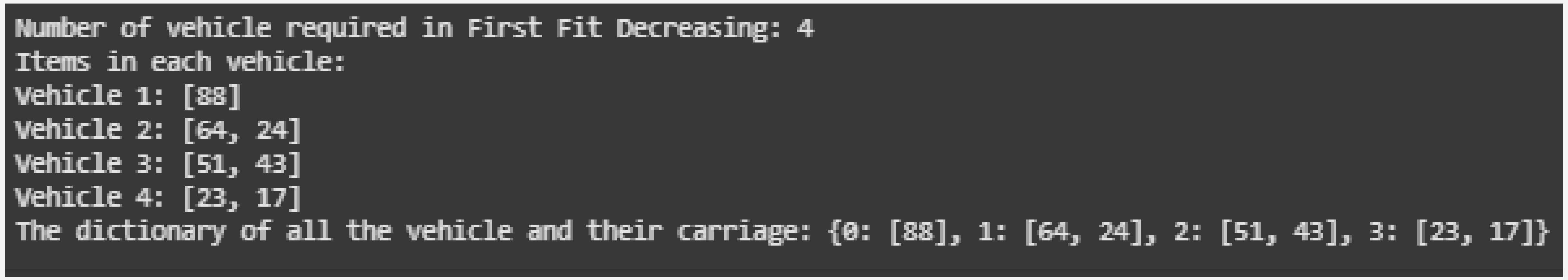
Code:

Iteration 1

How many vehicles are necessary to carry the packages to be delivered?



```python
# ==============================================================================
packages = [Package(23, 'Aurora'), Package(51, 'Antipolo'), Package(24, 'Marikina'),
            Package(17, 'Masbate'), Package(64, 'Bicol'), Package(43, 'Rizal'),
            Package(88, 'Zabarte')]
weight = [i.weight for i in packages]
c = 100
n = len(weight)
memo =  {} # for storing the vehicle and their carriage
num_vehicle, vehicle = firstFitDec(weight, n, c)


print("Number of vehicle required in First Fit Decreasing:", num_vehicle)
print("Items in each vehicle:")
for i in range(num_vehicle):
    memo[i] = vehicle[i]
    print(f"Vehicle {i + 1}: {vehicle[i]}")

print(f"The dictionary of all the vehicle and their carriage: {memo}")
```

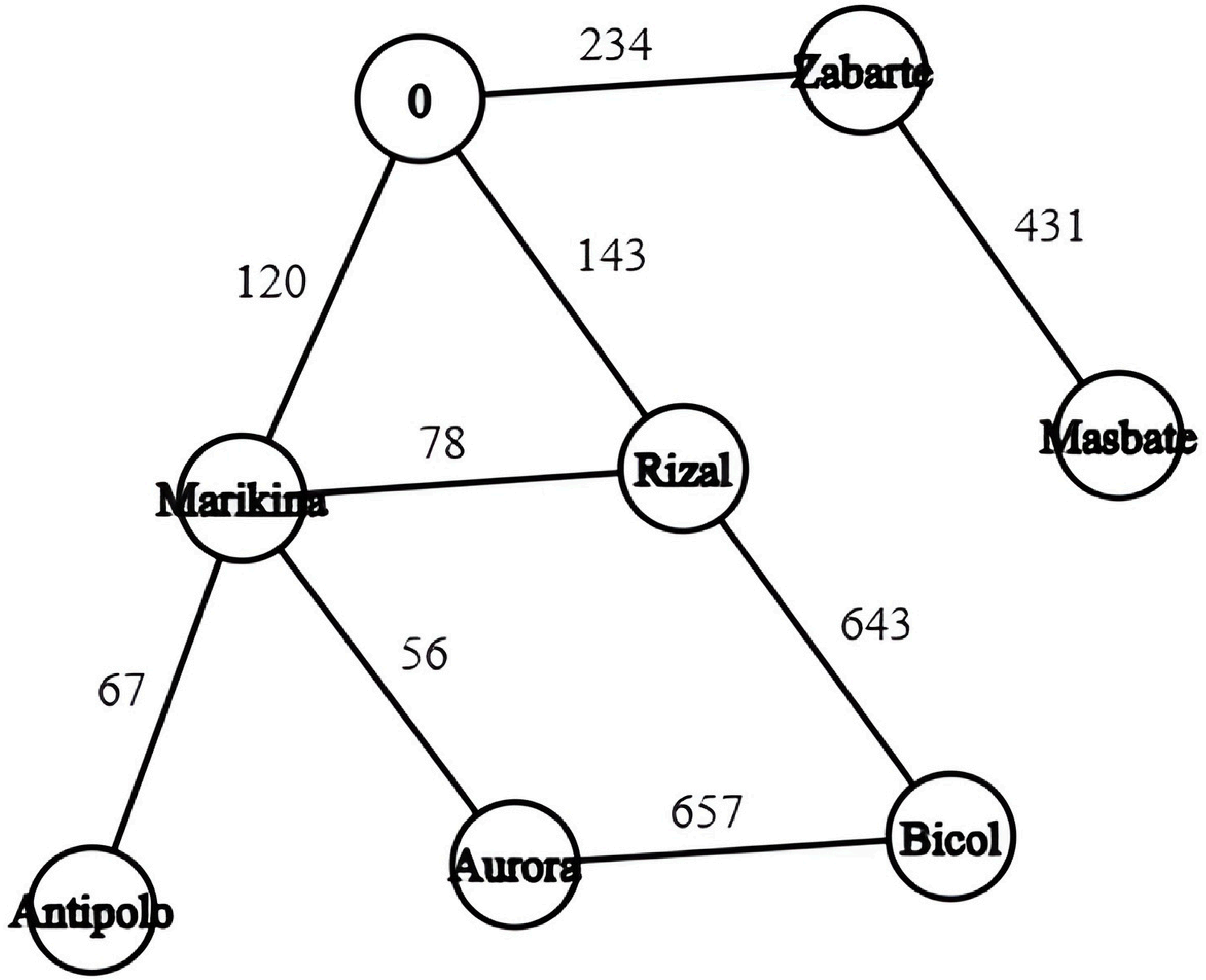


```
Number of vehicle required in First Fit Decreasing: 4
Items in each vehicle:
Vehicle 1: [88]
Vehicle 2: [64, 24]
Vehicle 3: [51, 43]
Vehicle 4: [23, 17]
The dictionary of all the vehicle and their carriage: {0: [88], 1: [64, 24], 2: [51, 43], 3: [23, 17]}
```

Graph:

Iteration 2

What is the optimal route of each ground transport chosen?

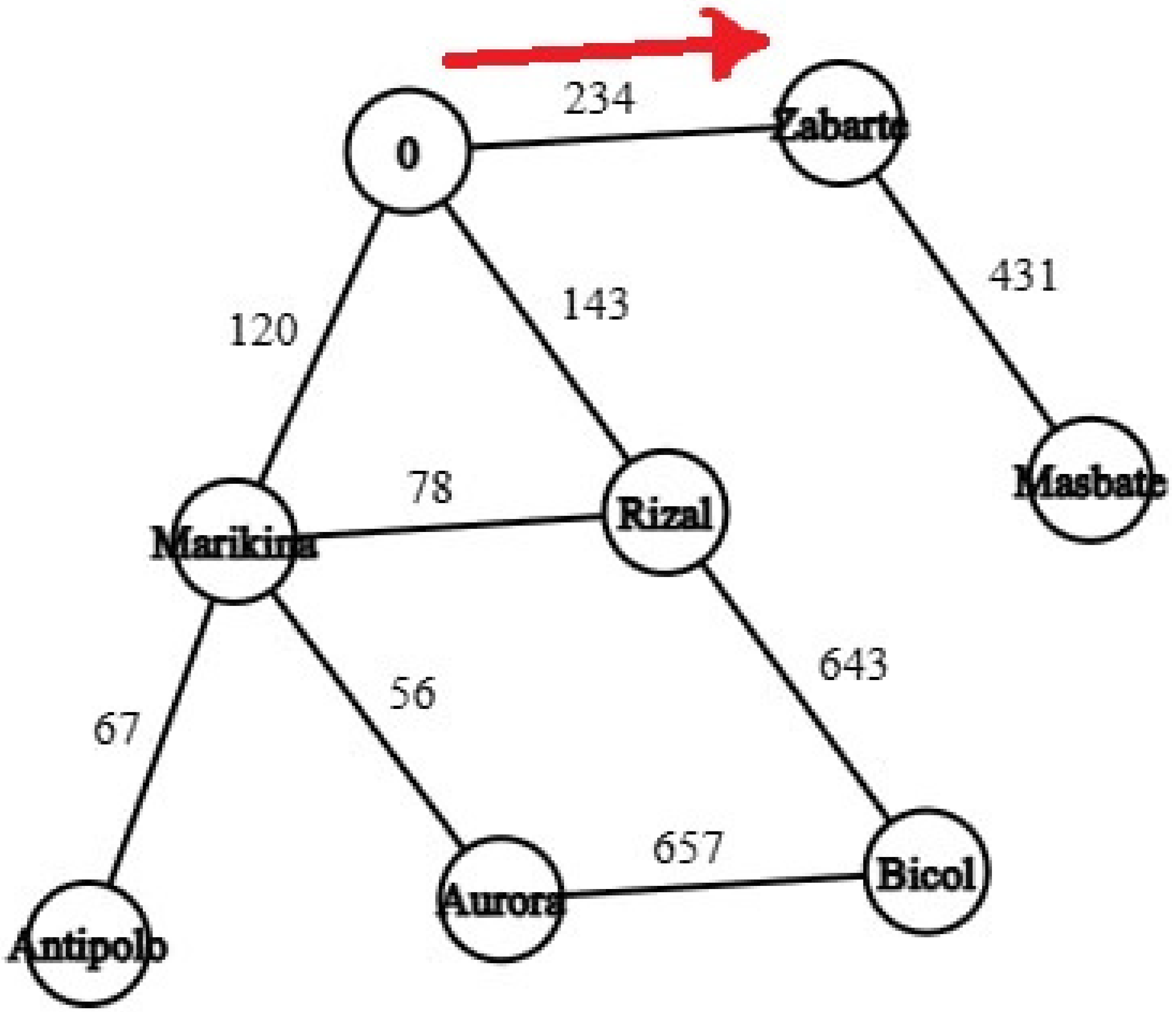


Graph:

Iteration 2

What is the optimal route of each ground transport chosen?

Vehicle 1

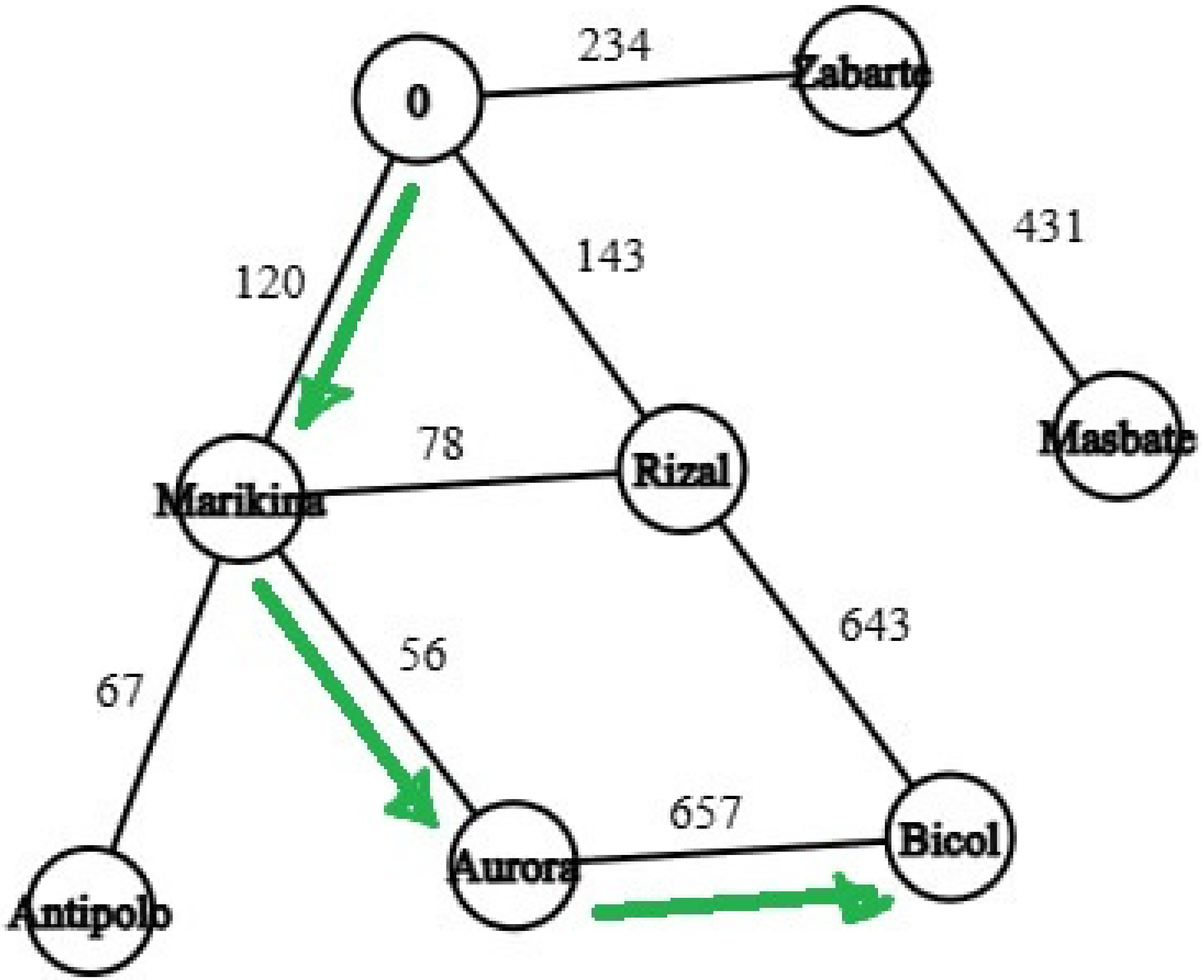


Graph:

Iteration 2

What is the optimal route of each ground transport chosen?

Vehicle 2

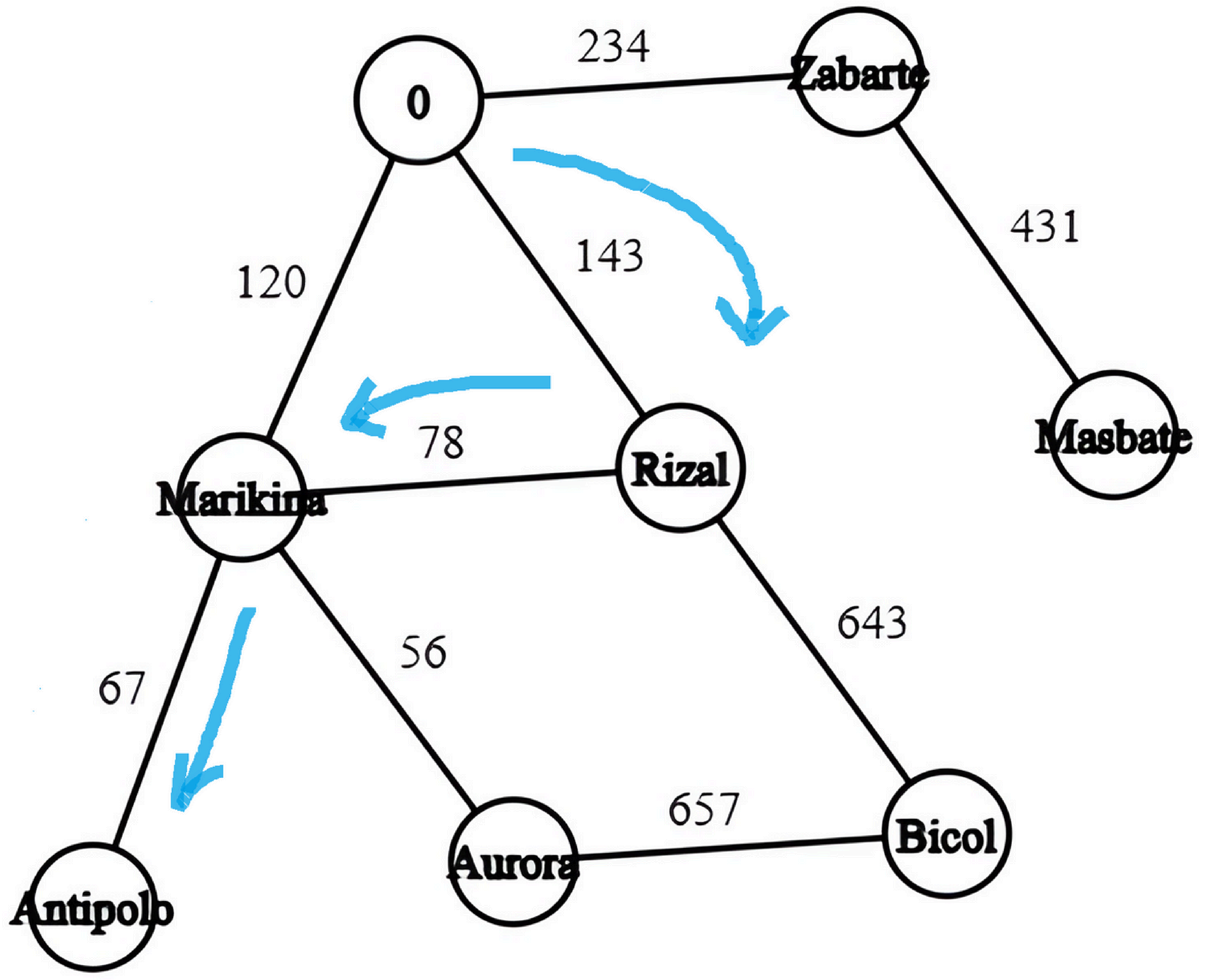


Graph:

Iteration 2

What is the optimal route of each ground transport chosen?

Vehicle 3

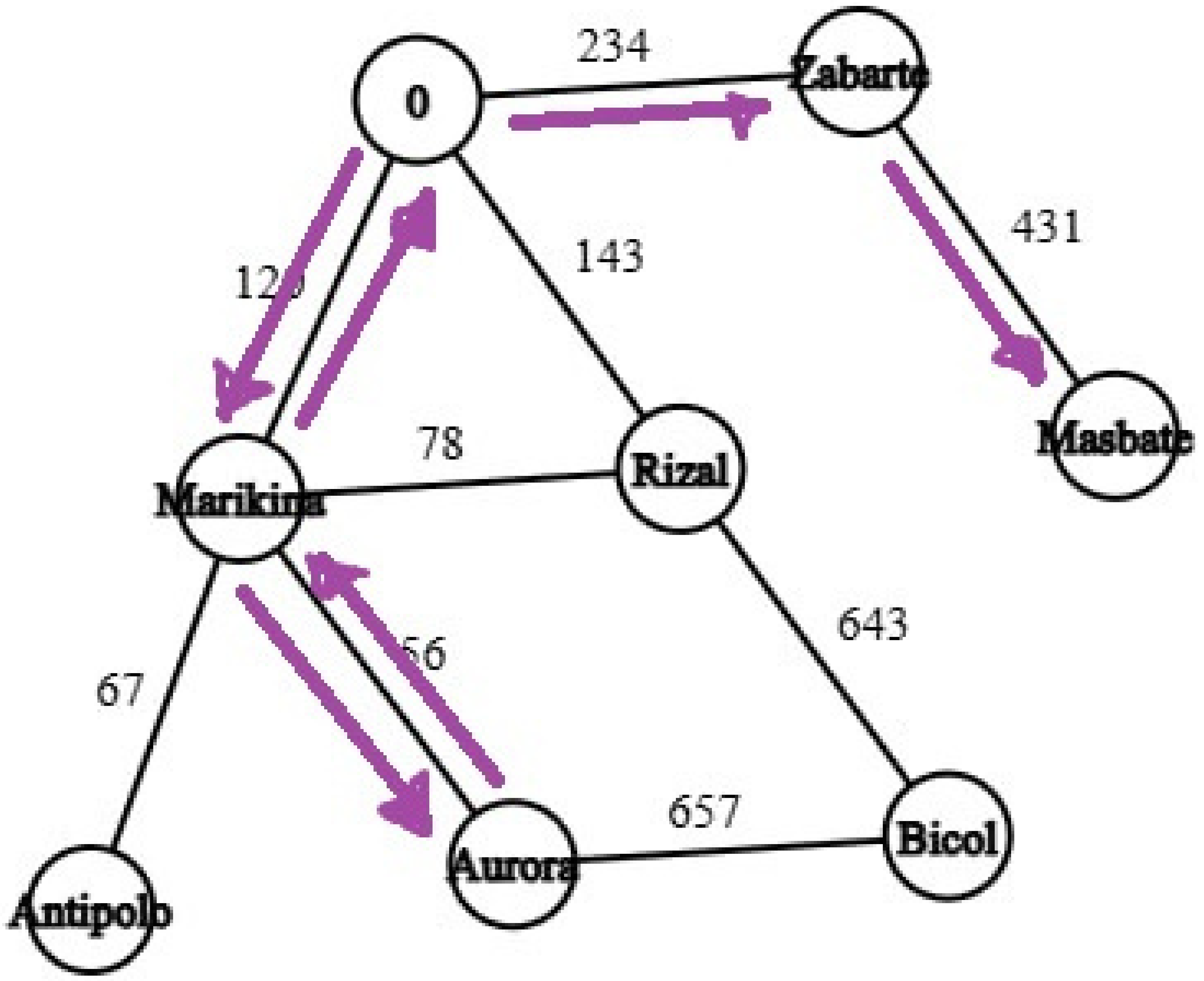


Graph:

Iteration 2

What is the optimal route of each ground transport chosen?

Vehicle 4


0
234
Zabarte
431
143
120
Masbate
78
Marikina
Rizal
643
66
67
657
Aurora
Bicol
Antipolo