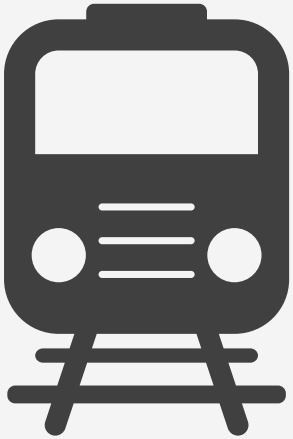
CASE STUDY 1: SOLVING REAL-WORLD PROBLEMS USING COMPUTATIONAL THINKING

# TRAIN SYSTEM



**TEAM 6**

ALORRO, JAY-ANN
CABOS, JHILLIAN
GALAPIA, XANDER SAM

**PROBLEM:**

THE DIFFICULTY TO USE THE METRO MANILA RAILWAY TRANSIT SYSTEM.

# ITERATION 01

## PROBLEM IDENTIFICATION

There are 3 different railway transits in Metro Manila.

## DECOMPOSITION

HOW WOULD YOU BREAK DOWN YOUR PROBLEM INTO SUB- PROBLEMS?

- Listing down the routes that each LRT/MRT station takes

## PATTERN RECOGNITION

ARE THERE RELATED SOLUTIONS TO DRAW ON?

- Each line have different start and end station
- There is one station that is near with one another
- Each station are in different areas

## ABSTRACTION

HOW WOULD YOU ABSTRACT THIS PROBLEM?

- What station am I in and where?
- Is there other railway transit available in the station I'm in?

# ITERATION 02

## PROBLEM IDENTIFICATION

Finding the most efficient route (via Train) and determining the fare between the designated train stations.

## DECOMPOSITION

HOW WOULD YOU BREAK DOWN YOUR PROBLEM INTO SUB- PROBLEMS?

- Identifying all possible routes between those 3 stations
- Fare depending on the Beep Card
- Pricing of the Beep Card

## PATTERN RECOGNITION

ARE THERE RELATED SOLUTIONS TO DRAW ON?

- Some routes may involve transfer between lines.
- Some routes may have faster travel time.
- More stops increase the price of the fare.
- When transferring between trains effects the price of the fare

## ABSTRACTION

HOW WOULD YOU ABSTRACT THIS PROBLEM?

- Knowing the Beep card pricing and fare structure to provide accurate fare pricing
- Number of transfers between lines.
- Total travel journey.
- Efficiency in terms of fare price and direction.

# ITERATION 03

## PROBLEM IDENTIFICATION

Developing an algorithm that finds the shortest path.

## DECOMPOSITION

HOW WOULD YOU BREAK DOWN YOUR PROBLEM INTO SUB- PROBLEMS?

- Identifying the start and end stations
- Finding all possible routes
- Evaluate each route found
- Selecting the shortest one

## PATTERN RECOGNITION

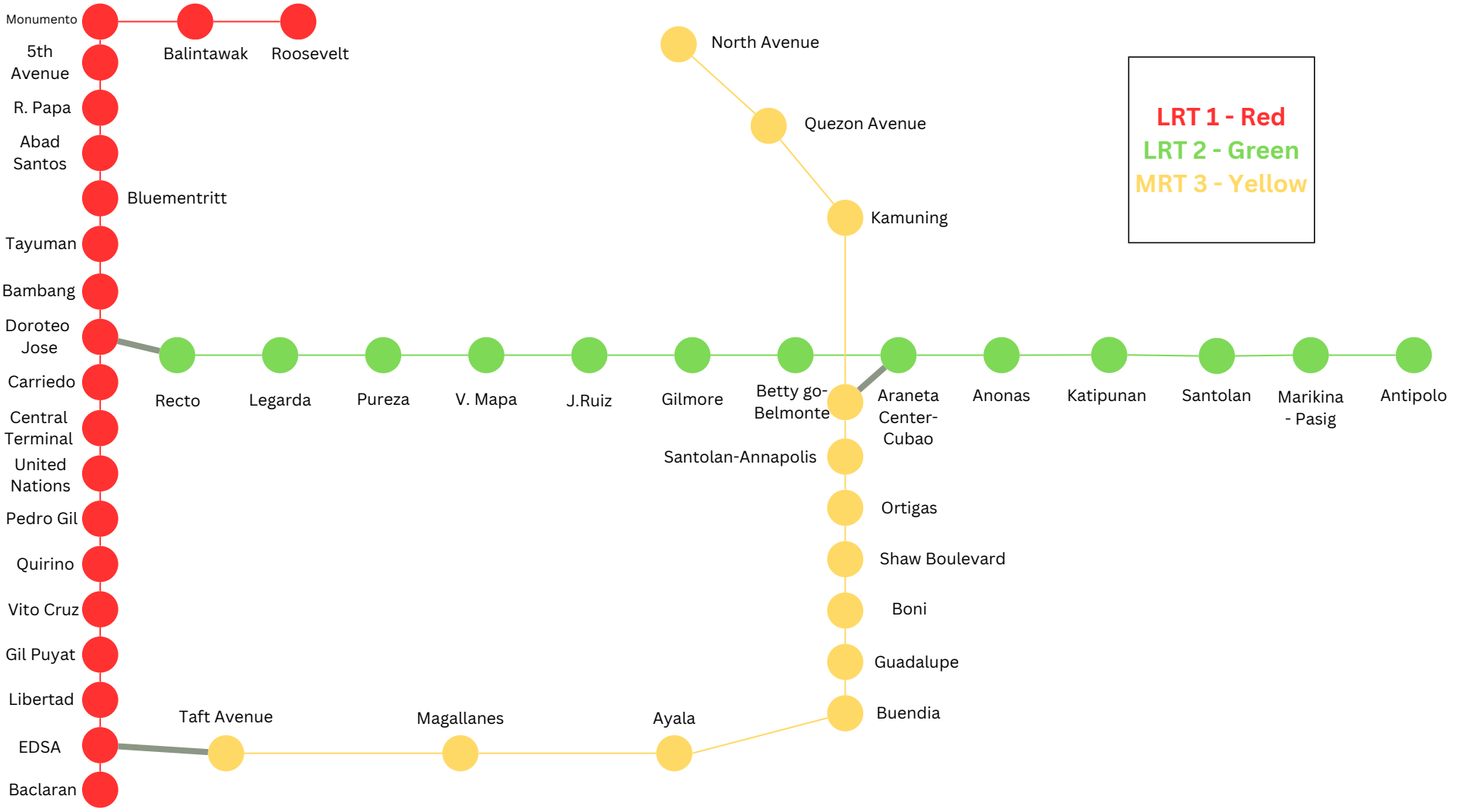
ARE THERE RELATED SOLUTIONS TO DRAW ON?

- Implementation of a path finding algorithm (DFS)
- Considering current-time fare prices

## ABSTRACTION

HOW WOULD YOU ABSTRACT THIS PROBLEM?

- Train network that includes the stations, lines, and their connections (Graph)
- Implementing the algorithm to find the shortest route.


Monumento
Balintawak
Roosevelt
5th Avenue
R. Papa
Abad Santos
Bluementritt
Tayuman
Bambang
Doroteo Jose
Carriedo
Central Terminal
United Nations
Pedro Gil
Quirino
Vito Cruz
Gil Puyat
Libertad
EDSA
Baclaran
North Avenue
Quezon Avenue
Kamuning
Recto
Legarda
Pureza
V. Mapa
J.Ruiz
Gilmore
Betty go-Belmonte
Araneta Center-Cubao
Anonas
Katipunan
Santolan
Marikina - Pasig
Antipolo
Santolan-Annapolis
Ortigas
Shaw Boulevard
Boni
Guadalupe
Buendia
Ayala
Magallanes
Taft Avenue
LRT 1 - Red
LRT 2 - Green
MRT 3 - Yellow

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

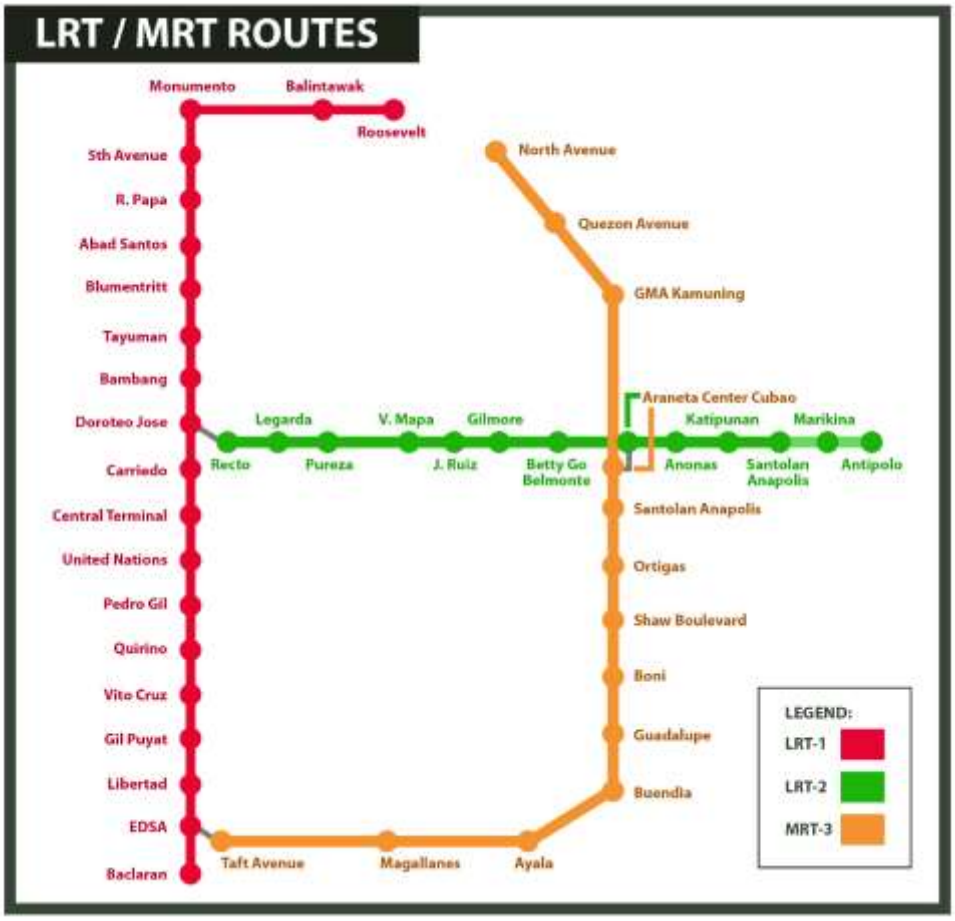
Members:

- Alorro, Jay-ann
- Cabos, Jhillian
- Galapia, Xander Sam

**Commuting via the Metro Manila Railway Transit (Train System)**

This program helps us select and organize a route that gives us the shortest path to arrive at our location using the railway transits found in metro manila (LRT 1 , LRT2, MRT 3).

*Documentation:*




- Each railway transit represented as a graph.

In [ ]:
```
lrt1 = {
    'Roosevelt': ['Balintawak'],
```

```
    'Balintawak': ['Roosevelt', 'Monumento'],
    'Monumento': ['Balintawak', '5th Avenue'],
    '5th Avenue': ['Monumento', 'R. Papa'],
    'R. Papa': ['5th Avenue', 'Abad Santos'],
    'Abad Santos': ['R. Papa', 'Blumetritt'],
    'Blumentritt': ['Abad Santos', 'Tayuman'],
    'Tayuman': ['Blumentritt', 'Bambang'],
    'Bambang': ['Tayuman', 'Donoteo Jose'],
    'Donoteo Jose': ['Bambang', 'Carriedo', 'Recto'], # recto from lrt2
    'Carriedo': ['Donoteo Jose', 'Central Terminal'],
    'Central Terminal': ['Corriedo', 'United Nations'],
    'United Nations': ['Central Terminal', 'Pedro Gil'],
    'Pedro Gil': ['United Nations', 'Quirino'],
    'Quirino': ['Pedro Gil', 'Vito Cruz'],
    'Vito Cruz': ['Quirino', 'Gil Puyat'],
    'Gil Puyat': ['Vito Cruz', 'Libertad'],
    'Libertad': ['Gil Puyat', 'EDSA'],
    'EDSA': ['Libertad', 'Baclaran'],
    'Baclaran': ['EDSA'],
}
```

In [ ]:
```
lrt2={
    'Santolan': ['Katipunan'],
    'Katipunan': ['Santolan', 'Anonas'],
    'Anonas': ['Katipunan', 'Araneta Center-Cubao'],
    'Araneta Center-Cubao': ['Anonas', 'Betty Go-Belmonte', 'Kamuning', 'Araneta Cente
    'Betty Go-Belmonte': ['Araneta Center-Cubao', 'Gilmore'],
    'Gilmore': ['Betty Go-Belmonte', 'J. Ruiz'],
    'J. Ruiz': ['Gilmore', 'V. Mapa'],
    'V. Mapa': ['J. Ruiz', 'Pureza'],
    'Pureza': ['V. Mapa', 'Legarda'],
    'Legarda': ['Pureza', 'Recto'],
    'Recto': ['Donoteo Jose','Legarda'] # donoteo jose from lrt1
}
```

In [ ]:
```
mrt3={
    'North Avenue': ['Quezon Avenue'],
    'Quezon Avenue': ['North Avenue', 'Kamuning'],
    'Kamuning': ['Quezon Avenue', 'Araneta Center-Cubao'],
    'Araneta Center-Cubao': ['Kamuning', 'Santolan-Annapolis', 'Araneta Center-Cubao']
    'Santolan-Annapolis': ['Araneta Center-Cubao', 'Ortigas'],
    'Ortigas': ['Santolan-Annapolis', 'Shaw Boulevard'],
    'Shaw Boulevard': ['Ortigas', 'Boni'],
    'Boni': ['Shaw Boulevard', 'Guadalupe'],
    'Guadalupe': ['Boni', 'Buendia'],
    'Buendia': ['Guadalupe', 'Ayala'],
    'Ayala': ['Buendia', 'Magallanes'],
    'Magallanes': ['Ayala', 'Taft Avenue'],
    'Taft Avenue': ['Magallanes', 'EDSA'] # edsa from lrt1
}
```

- All railway transit stations compiled into one graph.

In [ ]:
```
trainsG = {
    # lrt1
    'Roosevelt': ['Balintawak'],
    'Balintawak': ['Roosevelt', 'Monumento'],
```

```
    'Monumento': ['Balintawak', '5th Avenue'],
    '5th Avenue': ['Monumento', 'R. Papa'],
    'R. Papa': ['5th Avenue', 'Abad Santos'],
    'Abad Santos': ['R. Papa', 'Blumentritt'],
    'Blumentritt': ['Abad Santos', 'Tayuman'],
    'Tayuman': ['Blumentritt', 'Bambang'],
    'Bambang': ['Tayuman', 'Donoteo Jose'],
    'Donoteo Jose': ['Bambang', 'Carriedo', 'Recto'], # recto from lrt2
    'Carriedo': ['Donoteo Jose', 'Central Terminal'],
    'Central Terminal': ['Carriedo', 'United Nations'],
    'United Nations': ['Central Terminal', 'Pedro Gil'],
    'Pedro Gil': ['United Nations', 'Quirino'],
    'Quirino': ['Pedro Gil', 'Vito Cruz'],
    'Vito Cruz': ['Quirino', 'Gil Puyat'],
    'Gil Puyat': ['Vito Cruz', 'Libertad'],
    'Libertad': ['Gil Puyat', 'EDSA'],
    'EDSA': ['Libertad', 'Baclaran'],
    'Baclaran': ['EDSA'],

    # lrt2
    'Santolan': ['Katipunan'],
    'Katipunan': ['Santolan', 'Anonas'],
    'Anonas': ['Katipunan', 'Araneta Center-Cubao'],
    'Araneta Center-Cubao': ['Anonas', 'Betty Go-Belmonte', 'Kamuning', 'Araneta Cente
    'Betty Go-Belmonte': ['Araneta Center-Cubao', 'Gilmore'],
    'Gilmore': ['Betty Go-Belmonte', 'J. Ruiz'],
    'J. Ruiz': ['Gilmore', 'V. Mapa'],
    'V. Mapa': ['J. Ruiz', 'Pureza'],
    'Pureza': ['V. Mapa', 'Legarda'],
    'Legarda': ['Pureza', 'Recto'],
    'Recto': ['Donoteo Jose','Legarda'], # donoteo jose from lrt1

    # mrt3
    'North Avenue': ['Quezon Avenue'],
    'Quezon Avenue': ['North Avenue', 'Kamuning'],
    'Kamuning': ['Quezon Avenue', 'Araneta Center-Cubao'],
    'Araneta Center-Cubao': ['Kamuning', 'Santolan-Annapolis', 'Araneta Center-Cubao',
    'Santolan-Annapolis': ['Araneta Center-Cubao', 'Ortigas'],
    'Ortigas': ['Santolan-Annapolis', 'Shaw Boulevard'],
    'Shaw Boulevard': ['Ortigas', 'Boni'],
    'Boni': ['Shaw Boulevard', 'Guadalupe'],
    'Guadalupe': ['Boni', 'Buendia'],
    'Buendia': ['Guadalupe', 'Ayala'],
    'Ayala': ['Buendia', 'Magallanes'],
    'Magallanes': ['Ayala', 'Taft Avenue'],
    'Taft Avenue': ['Magallanes', 'EDSA'] # edsa from lrt1
}
```

- Algorthm used to produced the shortest path from one station to another.

In [ ]:
```
def sp(network, start, end, path=[]):
    path = path + [start]
    if start == end:
        return path
    if start not in network:
        return None
    shortest = None
    for station in network[start]:
```

```
        if station not in path:
            newpath = sp(network, station, end, path)
            if newpath:
                if not shortest or len(newpath) < len(shortest):
                    shortest = newpath
    return shortest

path = sp(trainsG, 'Anonas', 'Taft Avenue')
print('Path:', ' -> '.join(path) if path else "No path found.")
```

```
Path: Anonas -> Araneta Center-Cubao -> Santolan-Annapolis -> Ortigas -> Shaw Bouleva
rd -> Boni -> Guadalupe -> Buendia -> Ayala -> Magallanes -> Taft Avenue
```

- Checks the current station and returns the railway transit it belongs.

In [ ]:

```
# with directions
def get_train(station):
    if station in lrt1_stations:
        return 'LRT1'
    elif station in lrt2_stations:
        return 'LRT2'
    elif station in mrt3_stations:
        return 'MRT3'
    else:
        return None

# set instead of lists for a constant time complexity of O(1)
lrt1_stations = set([
    'Roosevelt', 'Balintawak', 'Monumento', '5th Avenue', 'R. Papa', 'Abad Santos',
    'Blumentritt', 'Tayuman', 'Bambang', 'Donoteo Jose', 'Carriedo', 'Central Terminal
    'United Nations', 'Pedro Gil', 'Quirino', 'Vito Cruz', 'Gil Puyat', 'Libertad',
    'EDSA', 'Baclaran'
])

lrt2_stations = set([
    'Santolan', 'Katipunan', 'Anonas', 'Araneta Center-Cubao', 'Betty Go-Belmonte',
    'Gilmore', 'J. Ruiz', 'V. Mapa', 'Pureza', 'Legarda', 'Recto'
])

mrt3_stations = set([
    'North Avenue', 'Quezon Avenue', 'Kamuning', 'Araneta Center-Cubao',
    'Santolan-Annapolis', 'Ortigas', 'Shaw Boulevard', 'Boni', 'Guadalupe',
    'Buendia', 'Ayala', 'Magallanes', 'Taft Avenue'
])

start_station = 'Kamuning'
end_station = 'Gilmore'

path = sp(trainsG, start_station, end_station)
if path:
    print("Shortest path from", start_station, "to", end_station, ":")
    for i in range(len(path)):
        station = path[i]
        train = get_train(station)
        print(f"{station} ({train})", end=" -> " if i < len(path) - 1 else "\n")
else:
    print("No path found from", start_station, "to", end_station)
```

```
Shortest path from Kamuning to Gilmore :
Kamuning (MRT3) -> Araneta Center-Cubao (LRT2) -> Betty Go-Belmonte (LRT2) -> Gilmore
(LRT2)
```

- OOP Implementation

In [ ]:

```
class TrainSystem:
    def __init__(location):
        location.trainsG = {
            # lrt1
            'Roosevelt': ['Balintawak'],
            'Balintawak': ['Roosevelt', 'Monumento'],
            'Monumento': ['Balintawak', '5th Avenue'],
            '5th Avenue': ['Monumento', 'R. Papa'],
            'R. Papa': ['5th Avenue', 'Abad Santos'],
            'Abad Santos': ['R. Papa', 'Blumentritt'],
            'Blumentritt': ['Abad Santos', 'Tayuman'],
            'Tayuman': ['Blumentritt', 'Bambang'],
            'Bambang': ['Tayuman', 'Donoteo Jose'],
            'Donoteo Jose': ['Bambang', 'Carriedo', 'Recto'],
            'Carriedo': ['Donoteo Jose', 'Central Terminal'],
            'Central Terminal': ['Carriedo', 'United Nations'],
            'United Nations': ['Central Terminal', 'Pedro Gil'],
            'Pedro Gil': ['United Nations', 'Quirino'],
            'Quirino': ['Pedro Gil', 'Vito Cruz'],
            'Vito Cruz': ['Quirino', 'Gil Puyat'],
            'Gil Puyat': ['Vito Cruz', 'Libertad'],
            'Libertad': ['Gil Puyat', 'EDSA'],
            'EDSA': ['Libertad', 'Baclaran'],
            'Baclaran': ['EDSA'],

            # lrt2
            'Antipolo':['Marikina- Pasig'],
            'Marikina- Pasig': ['Santolan', 'Antipolo'],
            'Santolan': ['Katipunan'],
            'Katipunan': ['Santolan', 'Anonas'],
            'Anonas': ['Katipunan', 'Araneta Center-Cubao'],
            'Araneta Center-Cubao': ['Anonas', 'Betty Go-Belmonte', 'Kamuning', 'Arane
            'Betty Go-Belmonte': ['Araneta Center-Cubao', 'Gilmore'],
            'Gilmore': ['Betty Go-Belmonte', 'J. Ruiz'],
            'J. Ruiz': ['Gilmore', 'V. Mapa'],
            'V. Mapa': ['J. Ruiz', 'Pureza'],
            'Pureza': ['V. Mapa', 'Legarda'],
            'Legarda': ['Pureza', 'Recto'],
            'Recto': ['Donoteo Jose', 'Legarda'],

            # mrt3
            'North Avenue': ['Quezon Avenue'],
            'Quezon Avenue': ['North Avenue', 'Kamuning'],
            'Kamuning': ['Quezon Avenue', 'Araneta Center-Cubao'],
            'Araneta Center-Cubao': ['Kamuning', 'Santolan-Annapolis', 'Araneta Center
            'Santolan-Annapolis': ['Araneta Center-Cubao', 'Ortigas'],
            'Ortigas': ['Santolan-Annapolis', 'Shaw Boulevard'],
            'Shaw Boulevard': ['Ortigas', 'Boni'],
            'Boni': ['Shaw Boulevard', 'Guadalupe'],
            'Guadalupe': ['Boni', 'Buendia'],
            'Buendia': ['Guadalupe', 'Ayala'],
            'Ayala': ['Buendia', 'Magallanes'],
            'Magallanes': ['Ayala', 'Taft Avenue'],
```

```
            'Taft Avenue': ['Magallanes', 'EDSA']
        }

    def sp(self, network, start, end, path=[]):
        path = path + [start]
        if start == end:
            return path
        if start not in network:
            return None
        shortest = None
        for station in network[start]:
            if station not in path:
                newpath = self.sp(network, station, end, path)
                if newpath:
                    if not shortest or len(newpath) < len(shortest):
                        shortest = newpath
        return shortest

    def trainArrival(self, station):
        if station in lrt1_stations:
            return 'LRT1'
        elif station in lrt2_stations:
            return 'LRT2'
        elif station in mrt3_stations:
            return 'MRT3'
        else:
            return None

    def find_shortest_path(self, start_station, end_station):
        path = self.sp(self.trainsG, start_station, end_station)
        return path
```

```
When entering a location ensure that it's the full name of the station
and also the first letter should be in Capital

Ex: Pasig-Marikina or Anonas

Enter start station: Kamuning
Enter end station: Taft Avenue

Shortest path from Kamuning to Taft Avenue :
Kamuning (MRT3) -> Araneta Center-Cubao (LRT2) -> Santolan-Annapolis (MRT3) -> Ortiga
s (MRT3) -> Shaw Boulevard (MRT3) -> Boni (MRT3) -> Guadalupe (MRT3) -> Buendia (MRT
3) -> Ayala (MRT3) -> Magallanes (MRT3) -> Taft Avenue (MRT3)
```

In [ ]:

```
def RunTS():
    train_system = TrainSystem()
    start_station = input("When entering a location ensure that it's the full name of
    end_station = input("Enter end station: ")
    path = train_system.find_shortest_path(start_station, end_station)
    if path:
        print("\nShortest path from", start_station, "to", end_station, ":")
        for i in range(len(path)):
            station = path[i]
            train = train_system.trainArrival(station)
            print(f"{station} ({train})", end=" -> " if i < len(path) - 1 else "\n")
    else:
        print("No path found from", start_station, "to", end_station)

lrt1_stations = set([
```

```
    'Roosevelt', 'Balintawak', 'Monumento', '5th Avenue', 'R. Papa', 'Abad Santos',
    'Blumentritt', 'Tayuman', 'Bambang', 'Donoteo Jose', 'Carriedo', 'Central Terminal
    'United Nations', 'Pedro Gil', 'Quirino', 'Vito Cruz', 'Gil Puyat', 'Libertad',
    'EDSA', 'Baclaran'
])

lrt2_stations = set([
   'Antipolo','Marikina- Pasig', 'Santolan', 'Katipunan', 'Anonas', 'Araneta Center-Cu
    'Gilmore', 'J. Ruiz', 'V. Mapa', 'Pureza', 'Legarda', 'Recto'
])

mrt3_stations = set([
    'North Avenue', 'Quezon Avenue', 'Kamuning', 'Araneta Center-Cubao',
    'Santolan-Annapolis', 'Ortigas', 'Shaw Boulevard', 'Boni', 'Guadalupe',
    'Buendia', 'Ayala', 'Magallanes', 'Taft Avenue'
])

RunTS()
```

```
When entering a location ensure that it's the full name of the station
and also the first letter should be in Capital

Ex: Pasig-Marikina or Anonas

Enter start station: Kamuning
Enter end station: Ortigas

Shortest path from Kamuning to Ortigas :
Kamuning (MRT3) -> Araneta Center-Cubao (LRT2) -> Santolan-Annapolis (MRT3) -> Ortiga
s (MRT3)
```

# Final Work:

In [ ]:

```
class TrainSystem:
    def __init__(location):
        location.trainsG = {
            # lrt1
            'Roosevelt': ['Balintawak'],
            'Balintawak': ['Roosevelt', 'Monumento'],
            'Monumento': ['Balintawak', '5th Avenue'],
            '5th Avenue': ['Monumento', 'R. Papa'],
            'R. Papa': ['5th Avenue', 'Abad Santos'],
            'Abad Santos': ['R. Papa', 'Blumentritt'],
            'Blumentritt': ['Abad Santos', 'Tayuman'],
            'Tayuman': ['Blumentritt', 'Bambang'],
            'Bambang': ['Tayuman', 'Donoteo Jose'],
            'Donoteo Jose': ['Bambang', 'Carriedo', 'Recto'],
            'Carriedo': ['Donoteo Jose', 'Central Terminal'],
            'Central Terminal': ['Carriedo', 'United Nations'],
            'United Nations': ['Central Terminal', 'Pedro Gil'],
            'Pedro Gil': ['United Nations', 'Quirino'],
            'Quirino': ['Pedro Gil', 'Vito Cruz'],
            'Vito Cruz': ['Quirino', 'Gil Puyat'],
            'Gil Puyat': ['Vito Cruz', 'Libertad'],
            'Libertad': ['Gil Puyat', 'EDSA'],
            'EDSA': ['Libertad', 'Baclaran'],
            'Baclaran': ['EDSA'],
```

```
            # lrt2
            'Antipolo': ['Marikina- Pasig'],
            'Marikina- Pasig': ['Santolan', 'Antipolo'],
            'Santolan': ['Katipunan'],
            'Katipunan': ['Santolan', 'Anonas'],
            'Anonas': ['Katipunan', 'Araneta Center-Cubao'],
            'Araneta Center-Cubao': ['Anonas', 'Betty Go-Belmonte', 'Kamuning', 'Arane
            'Betty Go-Belmonte': ['Araneta Center-Cubao', 'Gilmore'],
            'Gilmore': ['Betty Go-Belmonte', 'J. Ruiz'],
            'J. Ruiz': ['Gilmore', 'V. Mapa'],
            'V. Mapa': ['J. Ruiz', 'Pureza'],
            'Pureza': ['V. Mapa', 'Legarda'],
            'Legarda': ['Pureza', 'Recto'],
            'Recto': ['Donoteo Jose', 'Legarda'],

            # mrt3
            'North Avenue': ['Quezon Avenue'],
            'Quezon Avenue': ['North Avenue', 'Kamuning'],
            'Kamuning': ['Quezon Avenue', 'Araneta Center-Cubao'],
            'Araneta Center-Cubao': ['Kamuning', 'Santolan-Annapolis', 'Araneta Center
            'Santolan-Annapolis': ['Araneta Center-Cubao', 'Ortigas'],
            'Ortigas': ['Santolan-Annapolis', 'Shaw Boulevard'],
            'Shaw Boulevard': ['Ortigas', 'Boni'],
            'Boni': ['Shaw Boulevard', 'Guadalupe'],
            'Guadalupe': ['Boni', 'Buendia'],
            'Buendia': ['Guadalupe', 'Ayala'],
            'Ayala': ['Buendia', 'Magallanes'],
            'Magallanes': ['Ayala', 'Taft Avenue'],
            'Taft Avenue': ['Magallanes', 'EDSA']
        }
        location.stored_value_fare = 13  # stored value beep card
        location.single_journey_fare = 15  # single journey beep card

    def sp(self, network, start, end, path=[]):
        path = path + [start]
        if start == end:
            return path
        if start not in network:
            return None
        shortest = None
        for station in network[start]:
            if station not in path:
                newpath = self.sp(network, station, end, path)
                if newpath:
                    if not shortest or len(newpath) < len(shortest):
                        shortest = newpath
        return shortest

    def trainArrival(self, station):
        if station in lrt1_stations:
            return 'LRT1'
        elif station in lrt2_stations:
            return 'LRT2'
        elif station in mrt3_stations:
            return 'MRT3'
        else:
            return None
    def calculate_fare(self, stations_passed, card_type, trains_taken):
      fare = 0
      if card_type == 'a':
```

```
            fare += self.stored_value_fare * len(trains_taken)
            fare += (stations_passed - 1) * len(trains_taken)  # Adding 1 peso for each
        elif card_type =='b':
            fare += self.single_journey_fare * len(trains_taken)
            fare += (stations_passed // 2) * len(trains_taken) * 5  # Adding 5 pesos for
        return fare


    def find_shortest_path(self, start_station, end_station):
        path = self.sp(self.trainsG, start_station, end_station)
        return path
```

In [ ]:

```
def RunTS():
    line = "======================================================"
    train_system = TrainSystem()
    start_station = input("When entering a location ensure that it's the full name of
    end_station = input("Enter end station: ")
    print(line)
    card_type = input("\nEnter type of beep card:(Press the either a or b)\nStored val
    print (line)
    path = train_system.find_shortest_path(start_station, end_station)
    if path:
        print("Shortest path from", start_station, "to", end_station, ":")
        stations_passed = len(path)
        trains_taken = set()
        for i in range(len(path)):
            station = path[i]
            train = train_system.trainArrival(station)
            if train:
                trains_taken.add(train)
            print(f"{station} ({train})", end=" -> " if i < len(path) - 1 else "\n")

        total_fare = train_system.calculate_fare(stations_passed, card_type, trains_ta
        print(line+(f"\nStations passed: {stations_passed}"))
        print(line+(f"\nDifferent trains taken: {len(trains_taken)}"))
        print(line+(f"\nTotal fare: {total_fare} pesos"))
    else:
        print(line +"\nNo path found from", start_station, "to", end_station)

lrt1_stations = set([
    'Roosevelt', 'Balintawak', 'Monumento', '5th Avenue', 'R. Papa', 'Abad Santos',
    'Blumentritt', 'Tayuman', 'Bambang', 'Doroteo Jose', 'Carriedo', 'Central Terminal
    'United Nations', 'Pedro Gil', 'Quirino', 'Vito Cruz', 'Gil Puyat', 'Libertad',
    'EDSA', 'Baclaran'
])

lrt2_stations = set([
   'Antipolo','Marikina- Pasig', 'Santolan', 'Katipunan', 'Anonas', 'Araneta Center-Cu
    'Gilmore', 'J. Ruiz', 'V. Mapa', 'Pureza', 'Legarda', 'Recto'
])

mrt3_stations = set([
    'North Avenue', 'Quezon Avenue', 'Kamuning', 'Araneta Center-Cubao',
    'Santolan-Annapolis', 'Ortigas', 'Shaw Boulevard', 'Boni', 'Guadalupe',
    'Buendia', 'Ayala', 'Magallanes', 'Taft Avenue'
])
```

In [ ]:

```
RunTS()
```

```
When entering a location ensure that it's the full name of the station
and also the first letter should be in Capital

Ex: Pasig-Marikina or Anonas
==================================================
Enter start station: Boni
Enter end station: Anonas
==================================================

Enter type of beep card:(Press the either a or b)
Stored value (a)  Single Journey (b): a
==================================================
Shortest path from Boni to Anonas :
Boni (MRT3) -> Shaw Boulevard (MRT3) -> Ortigas (MRT3) -> Santolan-Annapolis (MRT3) -
> Araneta Center-Cubao (LRT2) -> Anonas (LRT2)
==================================================
Stations passed: 6
==================================================
Different trains taken: 2
==================================================
Total fare: 36 pesos
```

In [ ]: