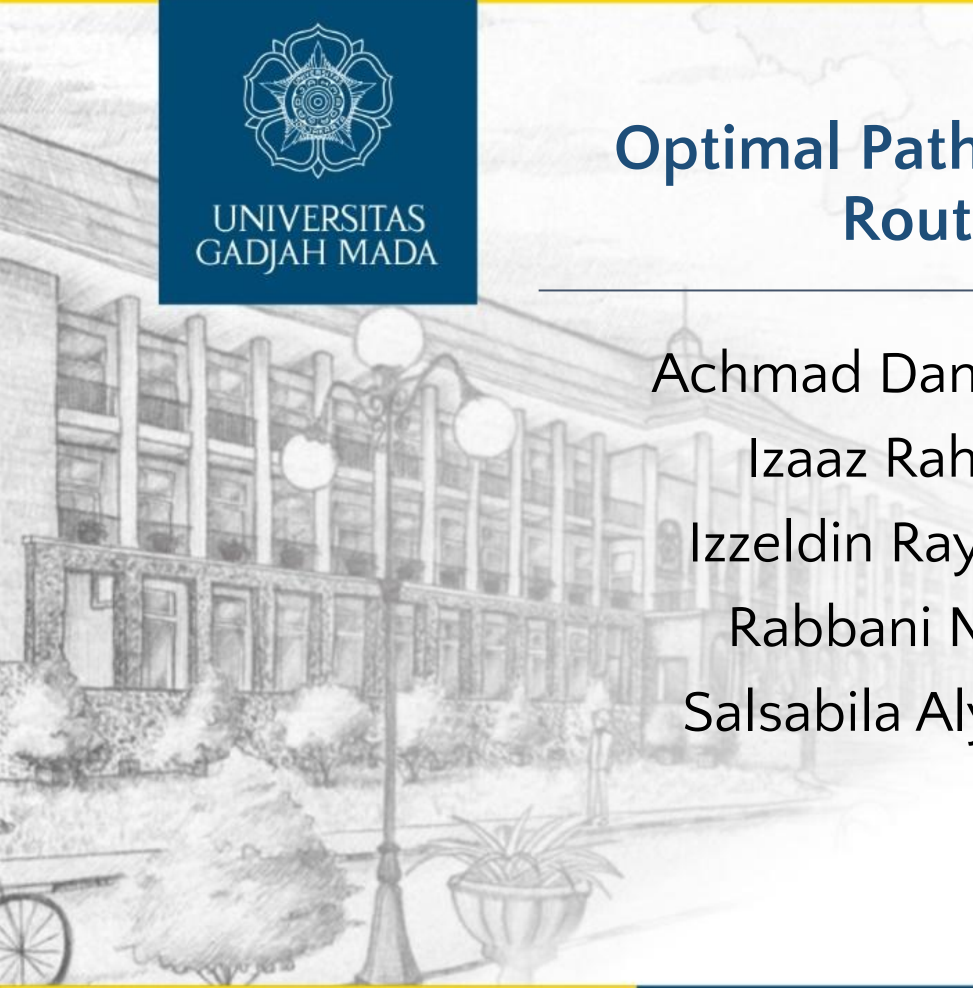
UNIVERSITAS
GADJAH MADA



Optimal Path Selection of Hong Kong's MTR Route Based on Dijkstra's Algorithm

Achmad Dani Nursanto (21/475048/PA/20520)
Izaaz Rahman Akbar (21/472855/PA/20348)
Izzeldin Rayyan Bastian (21/477936/PA/20718)
Rabbani Nur Kumoro (21/472599/PA/20310)
Salsabila Alyanitazahra (21/472702/PA/20324)

BACKGROUND

- ❖ Mass Transit Railroad (MTR) is the leading public transport network utilized in Hong Kong.
- ❖ Tourism is a huge industry within itself in Hong Kong and public transportation is a common amenity that tourists use.
- ❖ However, the MTR might confuse common tourists as it offers a wide variety of routes.

GOAL

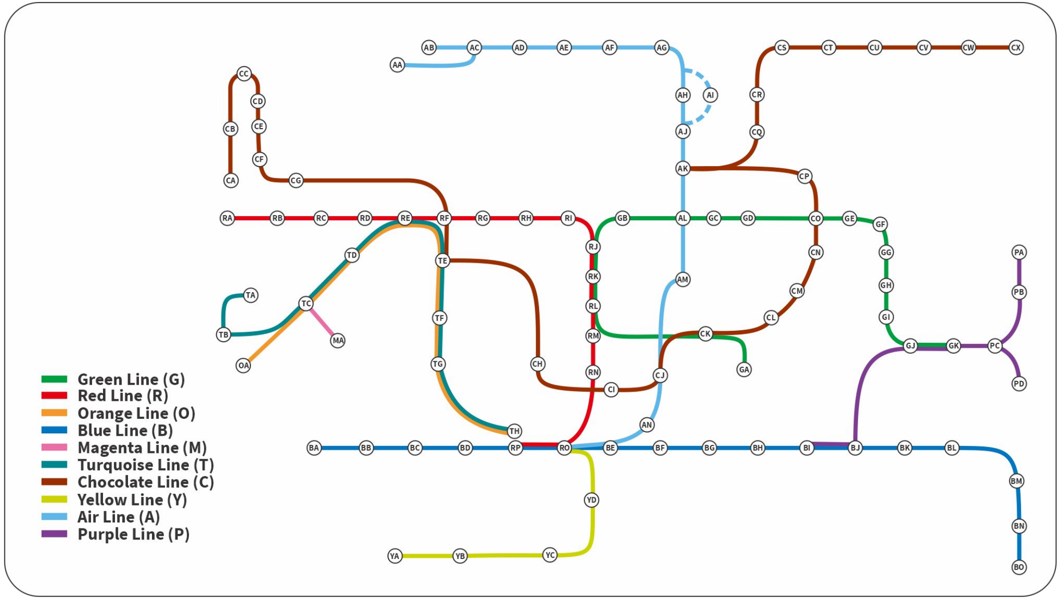
- ➔ Help a group of tourists planning their visit to Hong kong Disneyland using the MTR
- ➔ Find the shortest path, alternative path, route length, and estimated time

Destination: Kennedy Town Station to Disneyland Park

SOLUTION

Defining the problem into a graph:

- ➔ Construct the graph by declaring every node, path, and connected node.
- ➔ Each node represents a station in the MTR map.



SOLUTION

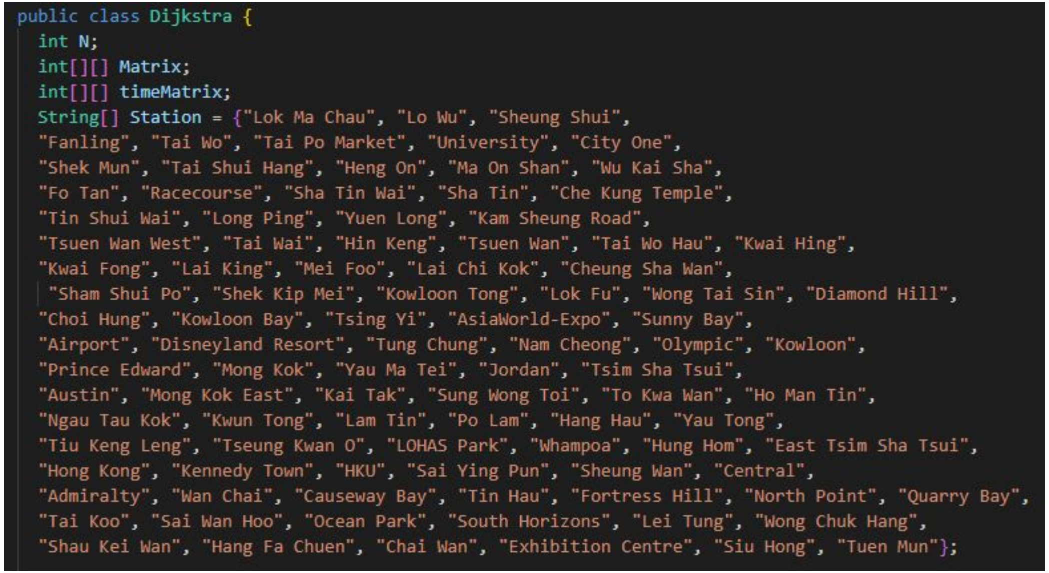
Java programming language will be used

CODE FUNCTION

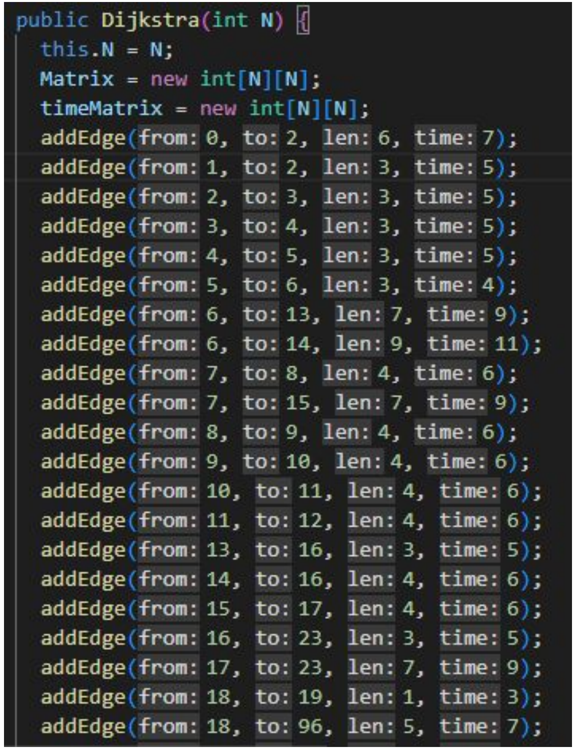
- Classes: djikstra and Main
- Functions:
  - ➔ dijkstra method
  - ➔ void addEdge
  - ➔ void dijkstraAlg
  - ➔ int minIndex
  - ➔ main method

CODE FUNCTIONS

dijkstra method



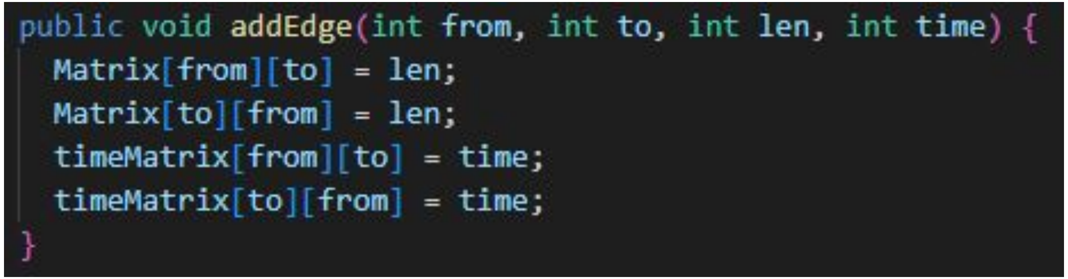
```
public class Dijkstra {
  int N;
  int[][] Matrix;
  int[][] timeMatrix;
  String[] Station = {"Lok Ma Chau", "Lo Wu", "Sheung Shui",
  "Fanling", "Tai Wo", "Tai Po Market", "University", "City One",
  "Shek Mun", "Tai Shui Hang", "Heng On", "Ma On Shan", "Wu Kai Sha",
  "Fo Tan", "Racecourse", "Sha Tin Wai", "Sha Tin", "Che Kung Temple",
  "Tin Shui Wai", "Long Ping", "Yuen Long", "Kam Sheung Road",
  "Tsuen Wan West", "Tai Wai", "Hin Keng", "Tsuen Wan", "Tai Wo Hau", "Kwai Hing",
  "Kwai Fong", "Lai King", "Mei Foo", "Lai Chi Kok", "Cheung Sha Wan",
   "Sham Shui Po", "Shek Kip Mei", "Kowloon Tong", "Lok Fu", "Wong Tai Sin", "Diamond Hill",
  "Choi Hung", "Kowloon Bay", "Tsing Yi", "AsiaWorld-Expo", "Sunny Bay",
  "Airport", "Disneyland Resort", "Tung Chung", "Nam Cheong", "Olympic", "Kowloon",
  "Prince Edward", "Mong Kok", "Yau Ma Tei", "Jordan", "Tsim Sha Tsui",
  "Austin", "Mong Kok East", "Kai Tak", "Sung Wong Toi", "To Kwa Wan", "Ho Man Tin",
  "Ngau Tau Kok", "Kwun Tong", "Lam Tin", "Po Lam", "Hang Hau", "Yau Tong",
  "Tiu Keng Leng", "Tseung Kwan O", "LOHAS Park", "Whampoa", "Hung Hom", "East Tsim Sha Tsui",
  "Hong Kong", "Kennedy Town", "HKU", "Sai Ying Pun", "Sheung Wan", "Central",
  "Admiralty", "Wan Chai", "Causeway Bay", "Tin Hau", "Fortress Hill", "North Point", "Quarry Bay",
  "Tai Koo", "Sai Wan Hoo", "Ocean Park", "South Horizons", "Lei Tung", "Wong Chuk Hang",
  "Shau Kei Wan", "Hang Fa Chuen", "Chai Wan", "Exhibition Centre", "Siu Hong", "Tuen Mun"};
```



```
public Dijkstra(int N) {
  this.N = N;
  Matrix = new int[N][N];
  timeMatrix = new int[N][N];
  addEdge(from: 0, to: 2, len: 6, time: 7);
  addEdge(from: 1, to: 2, len: 3, time: 5);
  addEdge(from: 2, to: 3, len: 3, time: 5);
  addEdge(from: 3, to: 4, len: 3, time: 5);
  addEdge(from: 4, to: 5, len: 3, time: 5);
  addEdge(from: 5, to: 6, len: 3, time: 4);
  addEdge(from: 6, to: 13, len: 7, time: 9);
  addEdge(from: 6, to: 14, len: 9, time: 11);
  addEdge(from: 7, to: 8, len: 4, time: 6);
  addEdge(from: 7, to: 15, len: 7, time: 9);
  addEdge(from: 8, to: 9, len: 4, time: 6);
  addEdge(from: 9, to: 10, len: 4, time: 6);
  addEdge(from: 10, to: 11, len: 4, time: 6);
  addEdge(from: 11, to: 12, len: 4, time: 6);
  addEdge(from: 13, to: 16, len: 3, time: 5);
  addEdge(from: 14, to: 16, len: 4, time: 6);
  addEdge(from: 15, to: 17, len: 4, time: 6);
  addEdge(from: 16, to: 23, len: 3, time: 5);
  addEdge(from: 17, to: 23, len: 7, time: 9);
  addEdge(from: 18, to: 19, len: 1, time: 3);
  addEdge(from: 18, to: 96, len: 5, time: 7);
```

CODE FUNCTIONS

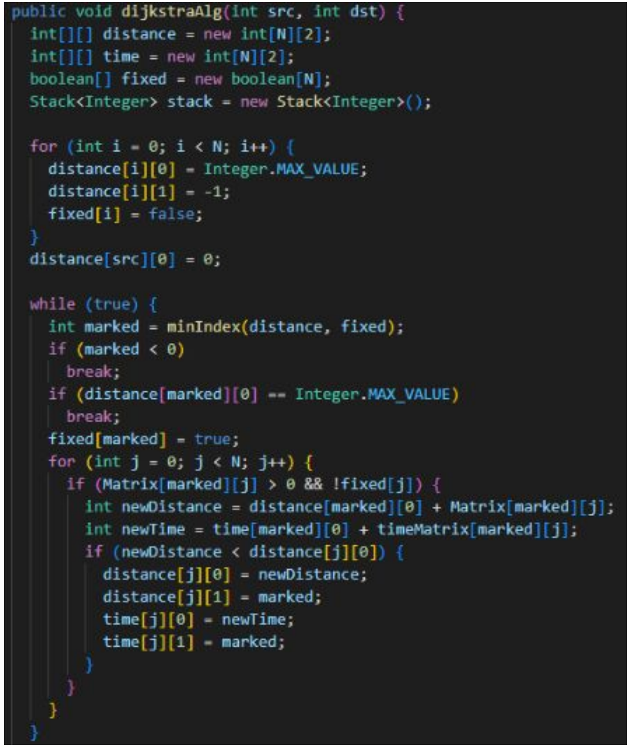
void addEdge



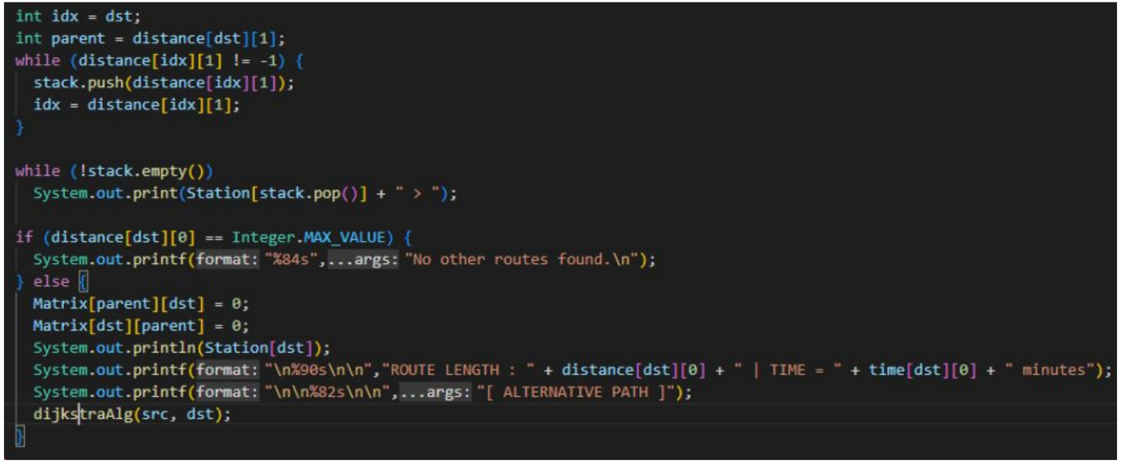
```
public void addEdge(int from, int to, int len, int time) {
  Matrix[from][to] = len;
  Matrix[to][from] = len;
  timeMatrix[from][to] = time;
  timeMatrix[to][from] = time;
}
```

CODE FUNCTIONS

void dijkstraAlg



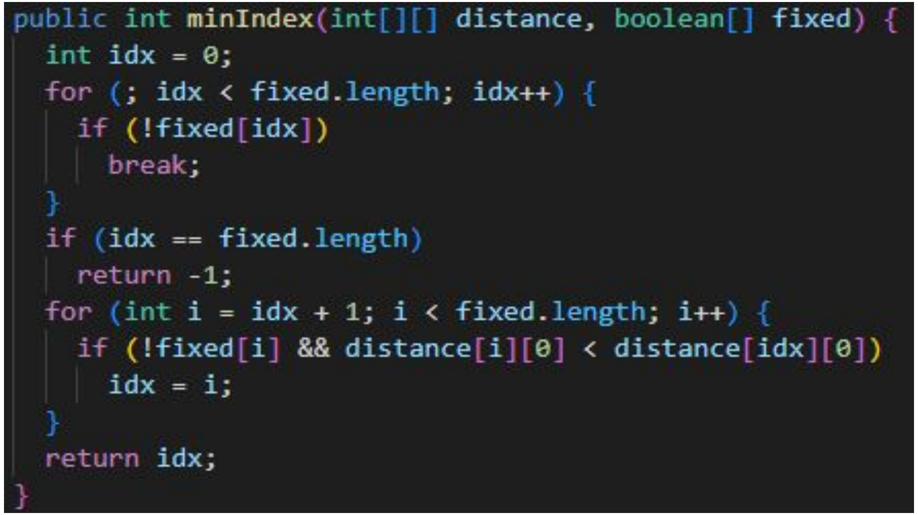
```
public void dijkstraAlg(int src, int dst) {
  int[][] distance = new int[N][2];
  int[][] time = new int[N][2];
  boolean[] fixed = new boolean[N];
  Stack<Integer> stack = new Stack<Integer>();

  for (int i = 0; i < N; i++) {
    distance[i][0] = Integer.MAX_VALUE;
    distance[i][1] = -1;
    fixed[i] = false;
  }
  distance[src][0] = 0;

  while (true) {
    int marked = minIndex(distance, fixed);
    if (marked < 0)
      break;
    if (distance[marked][0] == Integer.MAX_VALUE)
      break;
    fixed[marked] = true;
    for (int j = 0; j < N; j++) {
      if (Matrix[marked][j] > 0 && !fixed[j]) {
        int newDistance = distance[marked][0] + Matrix[marked][j];
        int newTime = time[marked][0] + timeMatrix[marked][j];
        if (newDistance < distance[j][0]) {
          distance[j][0] = newDistance;
          distance[j][1] = marked;
          time[j][0] = newTime;
          time[j][1] = marked;
        }
      }
    }
  }
```



```
int idx = dst;
int parent = distance[dst][1];
while (distance[idx][1] != -1) {
  stack.push(distance[idx][1]);
  idx = distance[idx][1];
}

while (!stack.empty())
  System.out.print(Station[stack.pop()] + " > ");

if (distance[dst][0] == Integer.MAX_VALUE) {
  System.out.printf(format: "%84s",...args: "No other routes found.\n");
} else {
  Matrix[parent][dst] = 0;
  Matrix[dst][parent] = 0;
  System.out.println(Station[dst]);
  System.out.printf(format: "\n%90s\n\n","ROUTE LENGTH : " + distance[dst][0] + " | TIME = " + time[dst][0] + " minutes");
  System.out.printf(format: "\n\n%82s\n\n",...args: "[ ALTERNATIVE PATH ]");
  dijkstraAlg(src, dst);
}
```

CODE FUNCTIONS

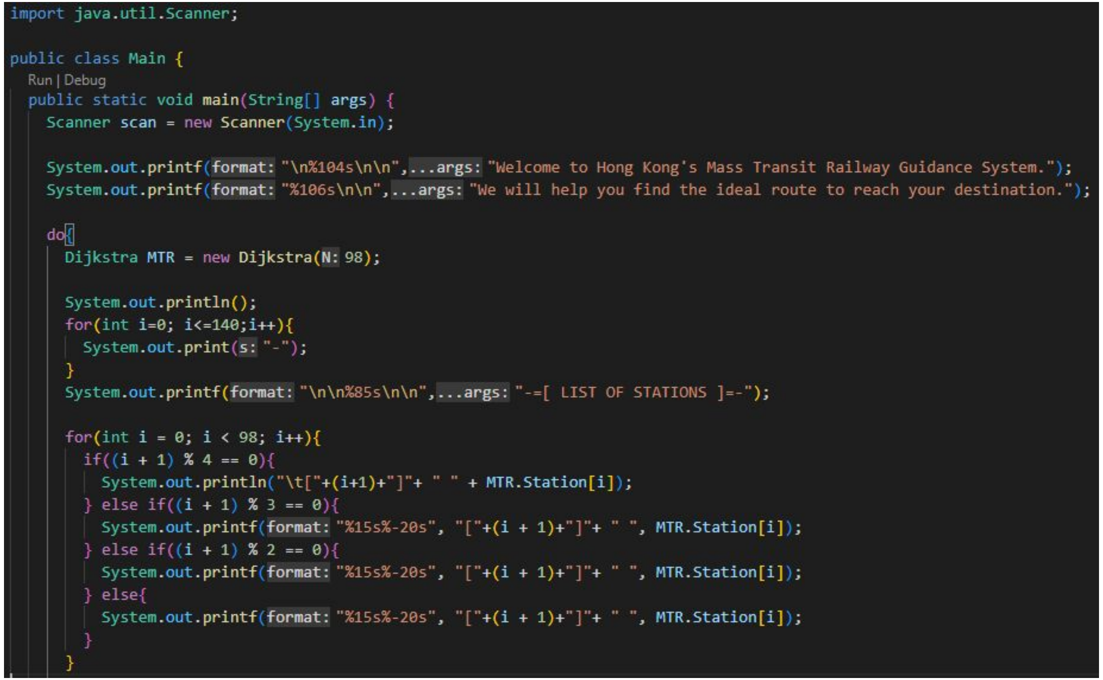
int minIndex



```
public int minIndex(int[][] distance, boolean[] fixed) {
  int idx = 0;
  for (; idx < fixed.length; idx++) {
    if (!fixed[idx])
      break;
  }
  if (idx == fixed.length)
    return -1;
  for (int i = idx + 1; i < fixed.length; i++) {
    if (!fixed[i] && distance[i][0] < distance[idx][0])
      idx = i;
  }
  return idx;
}
```

CODE FUNCTIONS

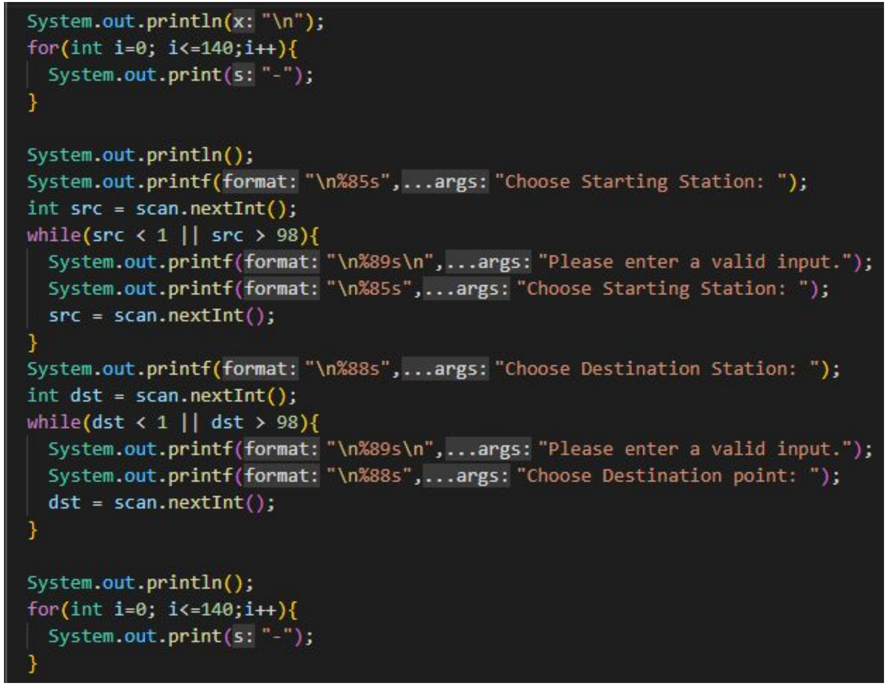
main method



```java
import java.util.Scanner;

public class Main {
  Run | Debug
  public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);

    System.out.printf(format: "\n%104s\n\n",...args: "Welcome to Hong Kong's Mass Transit Railway Guidance System.");
    System.out.printf(format: "%106s\n\n",...args: "We will help you find the ideal route to reach your destination.");

    do{
      Dijkstra MTR = new Dijkstra(N: 98);

      System.out.println();
      for(int i=0; i<=140;i++){
        System.out.print(s: "-");
      }
      System.out.printf(format: "\n\n%85s\n\n",...args: "-=[ LIST OF STATIONS ]=-");

      for(int i = 0; i < 98; i++){
        if((i + 1) % 4 == 0){
          System.out.println("\t["+(i+1)+"]"+ " " + MTR.Station[i]);
        } else if((i + 1) % 3 == 0){
          System.out.printf(format: "%15s%-20s", "["+(i + 1)+"]"+ " ", MTR.Station[i]);
        } else if((i + 1) % 2 == 0){
          System.out.printf(format: "%15s%-20s", "["+(i + 1)+"]"+ " ", MTR.Station[i]);
        } else{
          System.out.printf(format: "%15s%-20s", "["+(i + 1)+"]"+ " ", MTR.Station[i]);
        }
      }
```

CODE FUNCTIONS

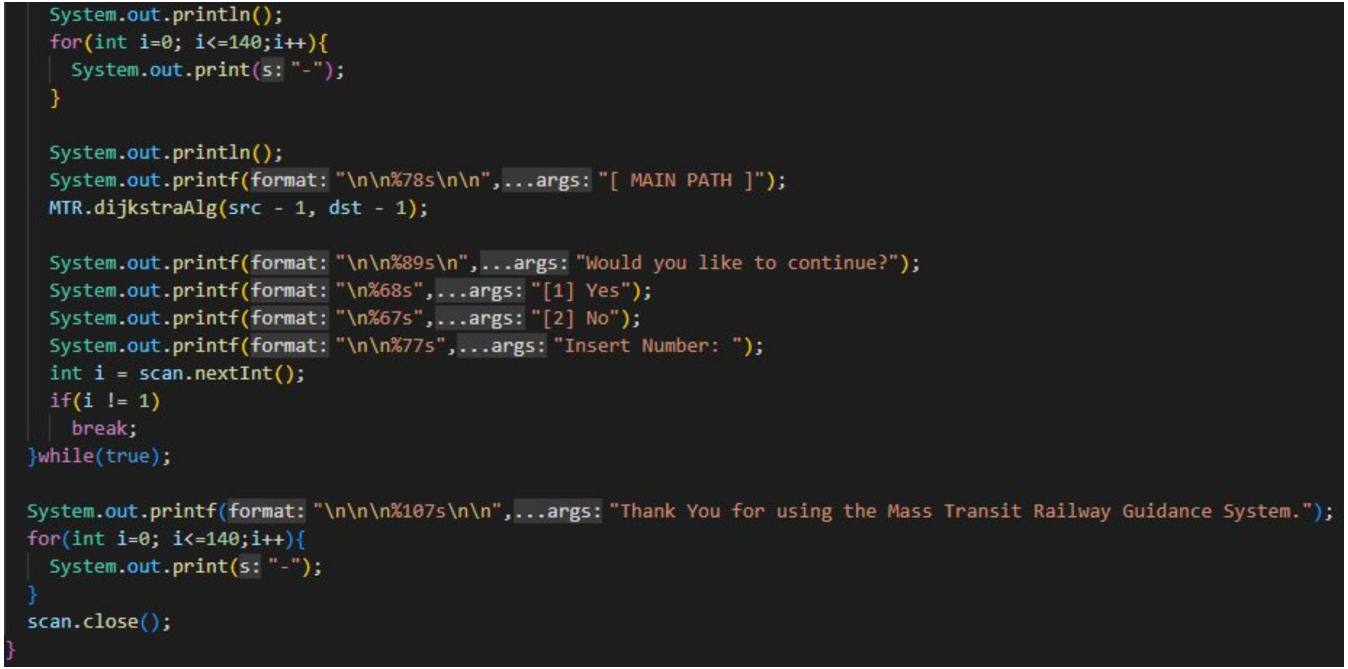
main method



```
System.out.println(x: "\n");
for(int i=0; i<=140;i++){
  System.out.print(s: "-");
}

System.out.println();
System.out.printf(format: "\n%85s",...args: "Choose Starting Station: ");
int src = scan.nextInt();
while(src < 1 || src > 98){
  System.out.printf(format: "\n%89s\n",...args: "Please enter a valid input.");
  System.out.printf(format: "\n%85s",...args: "Choose Starting Station: ");
  src = scan.nextInt();
}
System.out.printf(format: "\n%88s",...args: "Choose Destination Station: ");
int dst = scan.nextInt();
while(dst < 1 || dst > 98){
  System.out.printf(format: "\n%89s\n",...args: "Please enter a valid input.");
  System.out.printf(format: "\n%88s",...args: "Choose Destination point: ");
  dst = scan.nextInt();
}

System.out.println();
for(int i=0; i<=140;i++){
  System.out.print(s: "-");
}
```

CODE FUNCTIONS

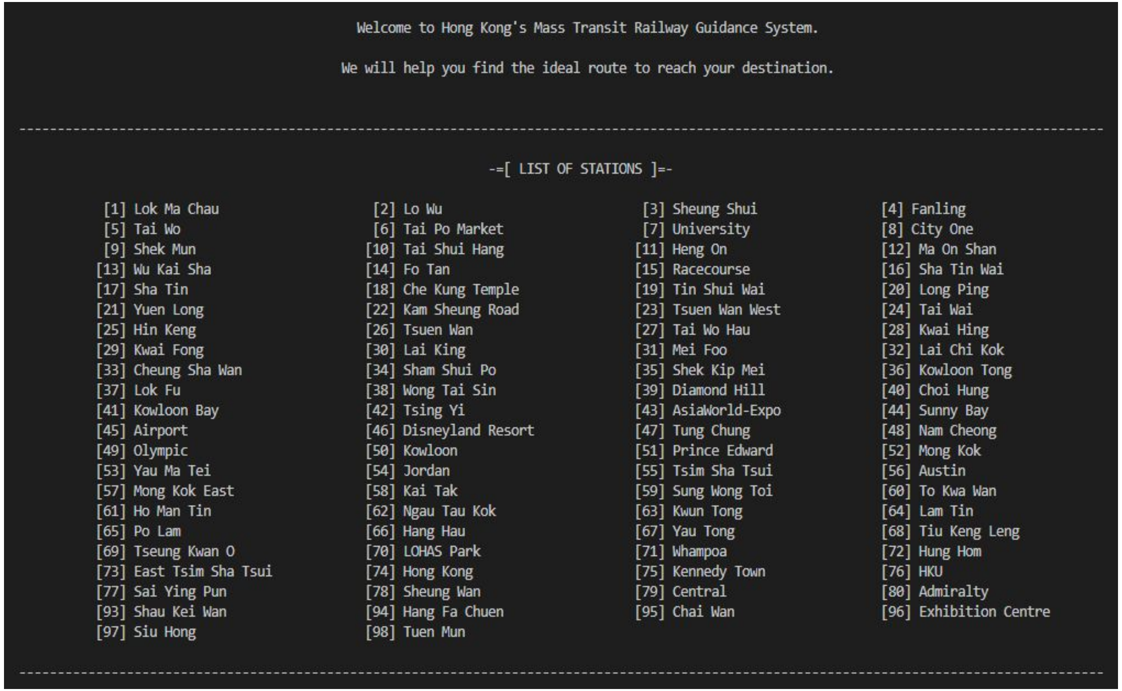
main method



```
        System.out.println();
        for(int i=0; i<=140;i++){
            System.out.print(s: "-");
        }

        System.out.println();
        System.out.printf(format: "\n\n%78s\n\n",...args: "[ MAIN PATH ]");
        MTR.dijkstraAlg(src - 1, dst - 1);

        System.out.printf(format: "\n\n%89s\n",...args: "Would you like to continue?");
        System.out.printf(format: "\n%68s",...args: "[1] Yes");
        System.out.printf(format: "\n%67s",...args: "[2] No");
        System.out.printf(format: "\n\n%77s",...args: "Insert Number: ");
        int i = scan.nextInt();
        if(i != 1)
            break;
    }while(true);

    System.out.printf(format: "\n\n\n%107s\n\n",...args: "Thank You for using the Mass Transit Railway Guidance System.");
    for(int i=0; i<=140;i++){
        System.out.print(s: "-");
    }
    scan.close();
}
```

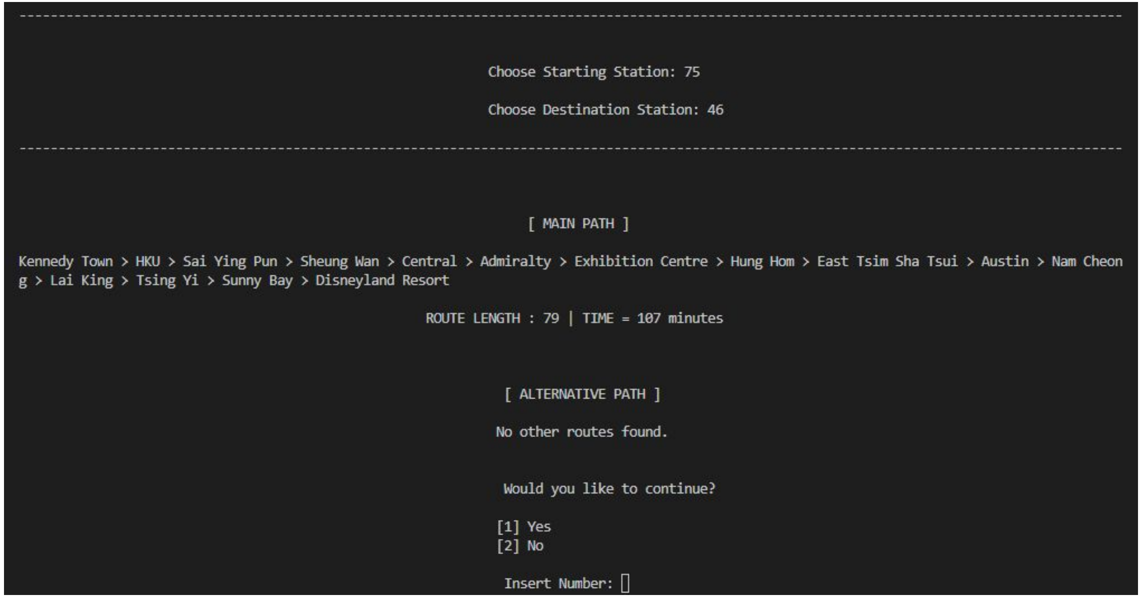
DEMO

OUTPUT



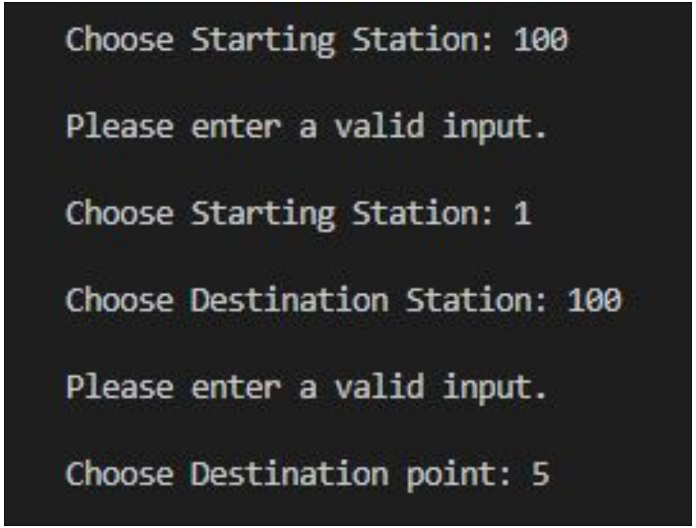
```
                         Welcome to Hong Kong's Mass Transit Railway Guidance System.

                          We will help you find the ideal route to reach your destination.


----------------------------------------------------------------------------------------------------------------


                                              -=[ LIST OF STATIONS ]=-

          [1] Lok Ma Chau                [2] Lo Wu                     [3] Sheung Shui             [4] Fanling
          [5] Tai Wo                     [6] Tai Po Market             [7] University              [8] City One
          [9] Shek Mun                  [10] Tai Shui Hang            [11] Heng On                [12] Ma On Shan
         [13] Wu Kai Sha                [14] Fo Tan                   [15] Racecourse             [16] Sha Tin Wai
         [17] Sha Tin                   [18] Che Kung Temple          [19] Tin Shui Wai           [20] Long Ping
         [21] Yuen Long                 [22] Kam Sheung Road          [23] Tsuen Wan West         [24] Tai Wai
         [25] Hin Keng                  [26] Tsuen Wan                [27] Tai Wo Hau             [28] Kwai Hing
         [29] Kwai Fong                 [30] Lai King                 [31] Mei Foo                [32] Lai Chi Kok
         [33] Cheung Sha Wan            [34] Sham Shui Po             [35] Shek Kip Mei           [36] Kowloon Tong
         [37] Lok Fu                    [38] Wong Tai Sin             [39] Diamond Hill           [40] Choi Hung
         [41] Kowloon Bay               [42] Tsing Yi                 [43] AsiaWorld-Expo         [44] Sunny Bay
         [45] Airport                   [46] Disneyland Resort        [47] Tung Chung             [48] Nam Cheong
         [49] Olympic                   [50] Kowloon                  [51] Prince Edward          [52] Mong Kok
         [53] Yau Ma Tei                [54] Jordan                   [55] Tsim Sha Tsui          [56] Austin
         [57] Mong Kok East             [58] Kai Tak                  [59] Sung Wong Toi          [60] To Kwa Wan
         [61] Ho Man Tin                [62] Ngau Tau Kok             [63] Kwun Tong              [64] Lam Tin
         [65] Po Lam                    [66] Hang Hau                 [67] Yau Tong               [68] Tiu Keng Leng
         [69] Tseung Kwan O             [70] LOHAS Park               [71] Whampoa                [72] Hung Hom
         [73] East Tsim Sha Tsui        [74] Hong Kong                [75] Kennedy Town           [76] HKU
         [77] Sai Ying Pun              [78] Sheung Wan               [79] Central                [80] Admiralty
         [93] Shau Kei Wan              [94] Hang Fa Chuen            [95] Chai Wan               [96] Exhibition Centre
         [97] Siu Hong                  [98] Tuen Mun

----------------------------------------------------------------------------------------------------------------
```

OUTPUT



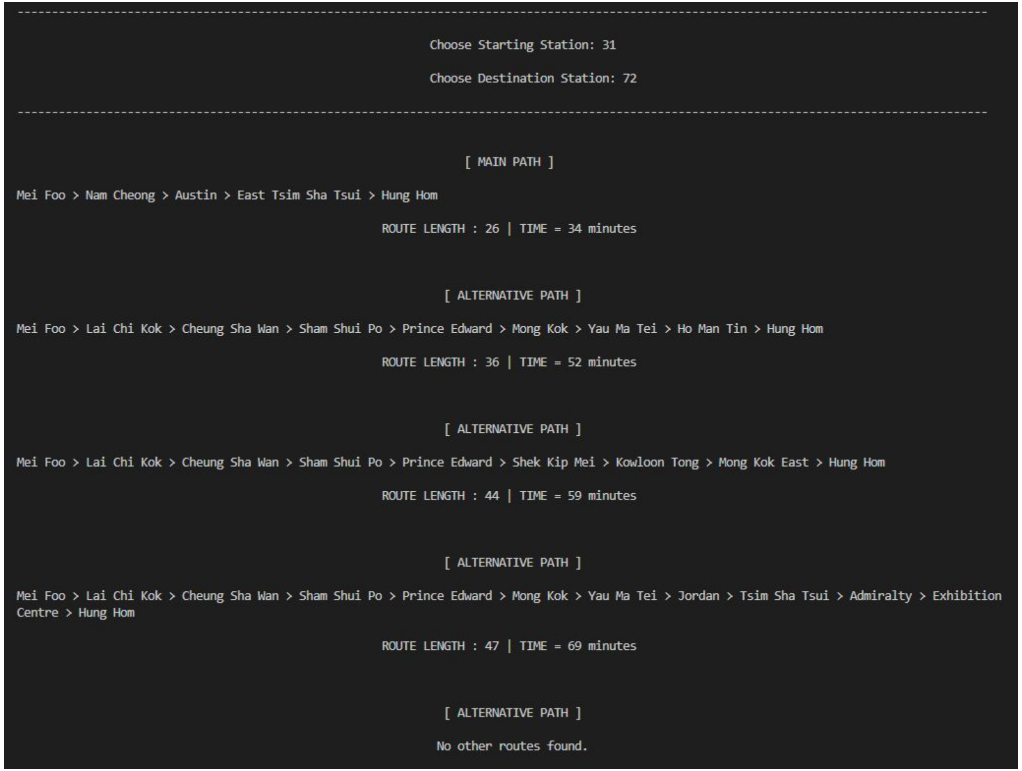
```
-------------------------------------------------------------------------------------------------------------------------------


                                          Choose Starting Station: 75

                                          Choose Destination Station: 46

-------------------------------------------------------------------------------------------------------------------------------



                                              [ MAIN PATH ]

Kennedy Town > HKU > Sai Ying Pun > Sheung Wan > Central > Admiralty > Exhibition Centre > Hung Hom > East Tsim Sha Tsui > Austin > Nam Cheon
g > Lai King > Tsing Yi > Sunny Bay > Disneyland Resort

                                     ROUTE LENGTH : 79 | TIME = 107 minutes



                                           [ ALTERNATIVE PATH ]

                                          No other routes found.


                                           Would you like to continue?

                                          [1] Yes
                                          [2] No

                                           Insert Number: 
```

OUTPUT



```
Choose Starting Station: 100

Please enter a valid input.

Choose Starting Station: 1

Choose Destination Station: 100

Please enter a valid input.

Choose Destination point: 5
```

OUTPUT



```
------------------------------------------------------------------------------------------------------------------------

                                         Choose Starting Station: 31

                                         Choose Destination Station: 72

------------------------------------------------------------------------------------------------------------------------


                                              [ MAIN PATH ]

Mei Foo > Nam Cheong > Austin > East Tsim Sha Tsui > Hung Hom

                                    ROUTE LENGTH : 26 | TIME = 34 minutes


                                           [ ALTERNATIVE PATH ]

Mei Foo > Lai Chi Kok > Cheung Sha Wan > Sham Shui Po > Prince Edward > Mong Kok > Yau Ma Tei > Ho Man Tin > Hung Hom

                                    ROUTE LENGTH : 36 | TIME = 52 minutes


                                           [ ALTERNATIVE PATH ]

Mei Foo > Lai Chi Kok > Cheung Sha Wan > Sham Shui Po > Prince Edward > Shek Kip Mei > Kowloon Tong > Mong Kok East > Hung Hom

                                    ROUTE LENGTH : 44 | TIME = 59 minutes


                                           [ ALTERNATIVE PATH ]

Mei Foo > Lai Chi Kok > Cheung Sha Wan > Sham Shui Po > Prince Edward > Mong Kok > Yau Ma Tei > Jordan > Tsim Sha Tsui > Admiralty > Exhibition
Centre > Hung Hom

                                    ROUTE LENGTH : 47 | TIME = 69 minutes


                                           [ ALTERNATIVE PATH ]

                                           No other routes found.
```

CONCLUSION

We can conclude that this program is a “Smart and Simple” navigation system for finding the best route to reach our chosen destination.

It came as a solution for the tourists to find the shortest path, alternative path, route length, and estimated time using Dijkstra’s algorithm for their vacation.

THANK YOU