

In my algorithm, I considered the followings:

- Each city has two stations even if it says that it has only one station in the input.
- There should be 1 list for each station and each index should indicate the distance between the current station and the station that corresponds to the current index. (For example in EXAMPLE 3, the 2nd index in istanbul_bus3 indicates that the distance between istanbul_bus3 and ankara_bus3 is 7.)
- If there is a 0 in the current index, it means that I am trying to go from x to x.
- If there is a -1 in the current index, it means that there is no direct transportation between the two stations. (Like there is no edge between the two nodes.)
- If there is a -2 in the current index, it means that there is no such station in the given input.
- If there is a number bigger than 0 in the current index, it means that there is direct transportation between the two stations.

FUNCTIONAL TESTING:

EXAMPLES 1, 2, 3

```
#EXAMPLE 1 - the set of cities is empty
twod_list1 = []

#EXAMPLE 2 - there is 1 item in the set of cities
istanbul_bus2 = [0, 3]
istanbul_train2 = [3, 0]
twod_list2 = []
twod_list2.append(istanbul_bus2)
twod_list2.append(istanbul_train2)

# #EXAMPLE 3 - there are 3 items in the set of cities, there are 6 valid stations
istanbul_bus3 = [0,3,7,-1,10,-1]
istanbul_train3 = [3,0,-1,8,-1,-1]
ankara_bus3 = [7,-1,0,5,-1,-1]
ankara_train3 = [-1,8,5,0,-1,6]
hatay_bus3 = [10,-1,-1,-1,0,5]
hatay_train3 = [-1,-1,-1,6,5,0]
twod_list3 = []
twod_list3.append(istanbul_bus3)
twod_list3.append(istanbul_train3)
twod_list3.append(ankara_bus3)
twod_list3.append(ankara_train3)
twod_list3.append(hatay_bus3)
twod_list3.append(hatay_train3)
```

EXAMPLES 4, 5

#EXAMPLE 4 - there are 5 items in the set of cities, there are 7 valid stations

```
istanbul_bus4 = [0,1,-1,-2,2,-2,-2,-1,13,-1]
istanbul_train4 = [1,0,-1,-2,-1,-2,-2,-1,-1,7]
ankara_bus4 = [-1,-1,0,-2,5,-2,-2,-1,-1,-1]
ankara_train4 = [-2,-2,-2,0,-2,-2,-2,-2,-2,-2]
bursa_bus4 = [2,-1,5,-2,0,-2,-2,-1,12,-1]
bursa_train4 = [-2,-2,-2,-2,-2,0,-2,-2,-2,-2]
eskisehir_bus4 = [-2,-2,-2,-2,-2,-2,0,-2,-2,-2]
eskisehir_train4 = [-1,-1,-1,-2,-1,-2,-2,0,-1,6]
hatay_bus4 = [13,-1,-1,-2,12,-2,-2,-1,0,2]
hatay_train4 = [-1,7,-1,-2,-1,-2,-2,6,2,0]
twod_list4 = []
twod_list4.append(istanbul_bus4)
twod_list4.append(istanbul_train4)
twod_list4.append(ankara_bus4)
twod_list4.append(ankara_train4)
twod_list4.append(bursa_bus4)
twod_list4.append(bursa_train4)
twod_list4.append(eskisehir_bus4)
twod_list4.append(eskisehir_train4)
twod_list4.append(hatay_bus4)
twod_list4.append(hatay_train4)
```

#EXAMPLE 5 - there are 5 items in the set of cities, there are 10 valid stations and there is a single itinerary as a line

```
istanbul_bus5 = [0,5,-1,-1,-1,-1,-1,-1,-1,-1]
istanbul_train5 = [5,0,-1,10,-1,-1,-1,-1,-1,-1]
rize_bus5 = [-1,-1,0,3,15,-1,-1,-1,-1,-1]
rize_train5 = [-1,10,3,0,-1,-1,-1,-1,-1,-1]
aydin_bus5 = [-1,-1,15,-1,0,2,-1,-1,-1,-1]
aydin_train5 = [-1,-1,-1,-1,2,0,-1,4,-1,-1]
mugla_bus5 = [-1,-1,-1,-1,-1,-1,0,1,20,-1]
mugla_train5 = [-1,-1,-1,-1,-1,4,1,0,-1,-1]
van_bus5 = [-1,-1,-1,-1,-1,-1,20,-1,0,6]
van_train5 = [-1,-1,-1,-1,-1,-1,-1,-1,6,0]
twod_list5 = []
twod_list5.append(istanbul_bus5)
twod_list5.append(istanbul_train5)
twod_list5.append(rize_bus5)
twod_list5.append(rize_train5)
twod_list5.append(aydin_bus5)
twod_list5.append(aydin_train5)
twod_list5.append(mugla_bus5)
twod_list5.append(mugla_train5)
twod_list5.append(van_bus5)
twod_list5.append(van_train5)
```

EXAMPLE 6

```
#EXAMPLE 6 - there are 7 items in the set of cities, there are 11 valid stations
istanbul_bus6 = [0,3,-1,-1,-1,-1,-1,-2,-1,-1,-2,-1,5,-2]
istanbul_train6 = [3,0,-1,-1,-1,-1,-1,-2,-1,4,-2,15,-1,-2]
ankara_bus6 = [-1,-1,0,2,-1,-1,-1,-2,-1,-1,-2,-1,3,-2]
ankara_train6 = [-1,-1,2,0,-1,12,-1,-2,-1,5,-2,10,-1,-2]
hatay_bus6 = [-1,-1,-1,-1,0,3,16,-2,8,-1,-2,-1,-1,-2]
hatay_train6 = [-1,-1,-1,12,3,0,-1,-2,-1,-1,-2,-1,-1,-2]
mugla_bus6 = [-1,-1,-1,-1,16,-1,0,-2,-1,-1,-2,-1,-1,-2]
mugla_train6 = [-2,-2,-2,-2,-2,-2,-2,0,-2,-2,-2,-2,-2,-2]
bursa_bus6 = [-1,-1,-1,-1,8,-1,-1,-2,0,2,-2,-1,-1,-2]
bursa_train6 = [-1,4,-1,5,-1,-1,-1,-2,2,0,-2,7,-1,-2]
rize_bus6 = [-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,0,-2,-2,-2]
rize_train6 = [-1,15,-1,10,-1,-1,-1,-2,-1,7,-2,0,-1,-2]
eskisehir_bus6 = [5,-1,3,-1,-1,-1,-1,-2,-1,-1,-2,-1,0,-2]
eskisehir_train6 = [-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2,0]
twod_list6 = []
twod_list6.append(istanbul_bus6)
twod_list6.append(istanbul_train6)
twod_list6.append(ankara_bus6)
twod_list6.append(ankara_train6)
twod_list6.append(hatay_bus6)
twod_list6.append(hatay_train6)
twod_list6.append(mugla_bus6)
twod_list6.append(mugla_train6)
twod_list6.append(bursa_bus6)
twod_list6.append(bursa_train6)
twod_list6.append(rize_bus6)
twod_list6.append(rize_train6)
twod_list6.append(eskisehir_bus6)
twod_list6.append(eskisehir_train6)
```

RESULTS

Each number corresponds to the quickest itinerary from the city with indexes 0, 1 to other cities such as the city with indexes 2, 3 or 4, 5.

For instance, in the 3rd example, the quickest itinerary from Istanbul to Ankara is 7, and the quickest itinerary from Istanbul to Hatay is 10.

```
EXAMPLE 1:
There is no city in the set
---
9.5367431640625e-07
---
EXAMPLE 2:
There is only one city in the set
---
1.33514404296875e-05
---
EXAMPLE 3:
7
10
---
9.417533874511719e-05
---
EXAMPLE 4:
7
2
14
8
---
0.0001919269561767578
---
EXAMPLE 5:
15
33
39
60
---
0.0002875328063964844
---
EXAMPLE 6:
8
17
33
7
14
5
---
0.00081634521484375
---
```

PERFORMANCE TESTING

The code that I have used for creating matrices with different input sizes:

```
from random import randrange

lengths = [5, 10, 15, 20, 25, 30, 35, 40]
for x in range(len(lengths)):
    length = lengths[x]

    twod_list_example = []
    for i in range(length):
        temp = []
        for j in range(length):
            temp.append(0)
        twod_list_example.append(temp)

    for i in range(length):
        for j in range(length):
            num = 0
            if i != j:
                num = randrange(-1,10)
                while num == 0:
                    num = randrange(-1,10)
            twod_list_example[i][j] = num
            twod_list_example[j][i] = num

    list_examples.append(twod_list_example)
```

The graph with the above examples:

