END2971 – Advanced Computer Programming

Mehmet G. Güler -
Ebru Geçici

First Due 12.11.2023 @23.45 (Sharp)
Second Due 14.11.2023 @23.45 (Sharp)

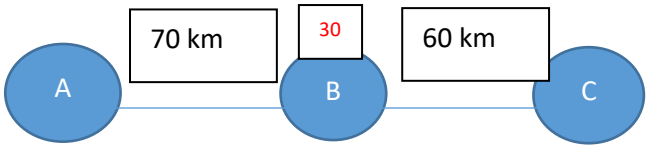
Assignment 1 (10%)

Explanation:

*You can work on the project with groups of at most 3 students. No, you cannot make it with more than3 students. This is the first assignment of END2971 Advanced Computer Programming lecture. You should submit it through online.yildiz.edu.tr by zipping your code, otherwise probably you will not be able to upload it. Every group must submit only one assignment and all of the students name should be written at the top of the jupiter file. Otherwise those who do not have their names cannot get any points. Late assignments are accepted at the cost of 20 points per days. For example if you submit at 23.**5**5 on 12.11.2023, you will get at max 80 points. You can submit at most 2 days late. After two days, the system will be closed. That is you cannot submit any homework after 14.11.2023 – 23:45*

Problem Definition:

One of the big challenges for the electrical vehicles (EV)to become widespread is the charging stations. It's important to find the optimal locations of charging stations. One criterion is to locate them so that an EV is able to go from an origin to a destination. For example, if the distance between A and B is 70 km and the distance between B and C is 60 km and the range of an EV is 100 km, then a fully charged EV is able to go from A to C only if you have a charging station in B.



In this project, we will seek answers to the question of where EV charging stations can potentially be installed in certain districts of Istanbul. For this, let's first examine the following tables:

1. **Distance matrix**: It gives the distance between some 13 selected districts of Istanbul.

District	Atasehir	Beykoz	Cekmekoy	Kadikoy	Kartal	Maltepe	Pendik	S.tepe	S.beyli	Sile	Tuzla	Umraniye	Uskudar
Atasehir	0	24	34	9	17	11	33	29	14	63	26	7	14
Beykoz	24	0	24	29	37	34	60	35	34	73	47	22	26
Cekmekoy	34	24	0	38	33	27	43	14	21	36	33	29	36
Kadikoy	9	29	38	0	19	13	20	24	18	63	30	12	11
Kartal	17	37	33	19	0	13	7	21	18	60	17	22	27
Maltepe	11	34	27	13	13	0	16	16	11	56	27	19	20
Pendik	33	60	43	20	7	16	0	29	23	71	16	27	29
Sancaktepe	29	35	14	24	21	16	29	0	10	41	21	19	28
Sultanbeyli	14	34	21	18	18	11	23	10	0	50	16	20	25
Sile	63	73	36	63	60	56	71	41	50	0	63	57	63
Tuzla	26	47	33	30	17	27	16	21	16	63	0	32	36
Umraniye	7	22	29	12	22	19	27	19	20	57	32	0	12
Uskudar	14	26	36	11	27	20	29	28	25	63	36	12	0

2. **Routes:** Shows the routes followed by vehicles and the IDs of these routes. For example, on the first route, vehicles leave Ataşehir and go to Ümraniye. On the second route, vehicles leave Ataşehir and go to Çekmeköy via Beykoz. Note that there can be many different routes on these 13 districts but the table gives only 6 of them.

ID	Routes
R1	Atasehir-Umraniye
R2	Atasehir - Beykoz-Cekmekoy
R3	Atasehir-Umraniye-Uskudar
R4	Beykoz-Cekmekoy-Sancaktepe
R5	Beykoz-Uskudar-Umraniye-Atasehir-Kadikoy-Maltepe-Kartal
R6	Beykoz-Cekmekoy-Sancaktepe-Sultanbeyli-Maltepe-Kartal-Pendik

3. **Combinations:** In operations research models, instead of deciding whether to open the charging stations individually, researchers study several combinations of these stations to be opened or not. As an example, 8 combinations are given below. Accordingly, if it is decided to open stations of first combination, one EV charging station will be opened in Ataşehir and one in Kadıköy. Similarly, if the 6th Combination is selected, totally three EV charging stations will be opened in Üsküdar, Kadıköy and Beykoz.

ID	Districts
C1	Atasehir – Kadikoy
C2	Beykoz – Cekmekoy
C3	Beykoz - Sancaktepe
C4	Cekmekoy-Sultanbeyli
C5	Maltepe- Kadıkoy
C6	Uskudar-Kadikoy-Beykoz
C7	Beykoz – Sultanbeyli – Kartal
C8	Beykoz – Ataşehir - Maltepe

4. **The range of a fully charged EV: Take it as 40 km for the example given below.**

The problem is to find which combinations are feasible for the given routes. For example, assume the range is 40 km. The first combination (C1) says that there will be one charging station at Ataşehir and one at Kadıköy. Now check the first route (R1) which is from Ataşehir to Ümraniye. Since there is a station in Ataşehir, the EV is fully charged, and its range becomes 40 km[1]. Then it goes from Ataşehir to Ümraniye which is 7 km. There is no charging statin in Ümraniye, hence at the end of the trip the amount of charge left is 33 km. So this combination is feasible for this route, i.e., if charging stations are opened in this combination, then an EV car can go through this path.

[1] While doing this project, we will make two assumptions about charging, one of which is that if a vehicle comes to a charging station, the vehicle is 100% charged.

For the same combination C1 (Ataşehir and Kadıköy), consider the second route R2 (Atasehir, Beykoz, Cekmekoy). The EV car starts full at Atasehir and goes to Beykoz. Since the distance is 24 km, the left range is 40-24 = 16 km. Then the EV car goes from Beykoz to Cekmekoy. However the distance between Beykoz and Cekmekoy is 24 km and but you have 16km of range. Hence it's not possible to go through the second route with the first combination.

If there is no station at the starting district of the route, you can assume that your EV is 50% charged, i.e., your starting range is ***(range/2)*** kilometers. This is our second assumption. Please note that your code should work for any given range, i.e., not only for 40 km.

To Do:

1. (50pt) Write a function called `check_feasibility` which takes two input as tuples: a route and a combination. It should return whether the combination is feasible for the route (True) or not (False). Hence the function will return a boolean that shows the feasibility of that route and combination. For example the function can read like the following

```
route = ("Beykoz", "Cekmekoy", "Sancaktepe")
combination = ("Uskudar", "Kadikoy")
def check_feasibility(route, combination)
    ……
```

2. (30pt) Write a function called `find_feasible_route` that takes (at least) two parameters. One of them is a list of routes and the other parameter is a list of combinations. It returns a dictionary which gives the feasible combinations for all of the given routes. Please use ID's to generate the dictionaries. The output will be something like this

```
LofC = [C1,C3, C5]
LofR = [R2, R4]

def find_feasible_route(LofR, LofC, …)
    ……
    ……

find_feasible_route(LofR, LofC)
```

>> {R2: (C1,C3), R4: (C3,C5) }

Note that the output is just for illustrative purposes, i.e., C1 may not be feasible for R2, for example.

3. (2opt)Write a function called `find_feasible_route_ex` that takes a list of routes and a list of combinations and returns a dictionary which gives the feasible combinations for all of the given routes. Now you will give the routes and combinations explicitly, i.e., you won't use the ID's given in the table. Assume that this function will be used for those routes and combinations which are not given in the tables above. The output will be something like this:

```
LofCex=[(Atasehir, Kadikoy),(Beykoz, Cekmekoy),(Cekmekoy,
Sultanbeyli) ]

LofRex = [(Atasehir, Umraniye),( Beykoz,Cekmekoy,Sancaktepe)]

def find_feasible_route_ex(LofRex, LofCex, …)
      ……
      ……

find_feasible_route_ex(LofRex, LofCex)

>> {(Atasehir, Umraniye): ((Atasehir, Kadikoy),), (Beykoz,
Cekmekoy, Sancaktepe): ((Beykoz, Cekmekoy),(Cekmekoy,
Sultanbeyli)) }
```

Note that the output is just for illustrative purposes, i.e., the routes or combinations may exist in the tables above.

4. You won't give any reports but you will submit jupyter file. Make sure that you write necessary comments for me to understand what you have done. I won't grade if I don't understand

Assume that all of the combinations and routes will be given from the lists above.

Good luck.