UNIVERSIDAD AUTONÓMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

COORDINACIÓN GENERAL DE CIENCIAS BÁSICAS

SUBJECT: **TSO**

SEMESTER: **JANUARY - JUNE 2024**

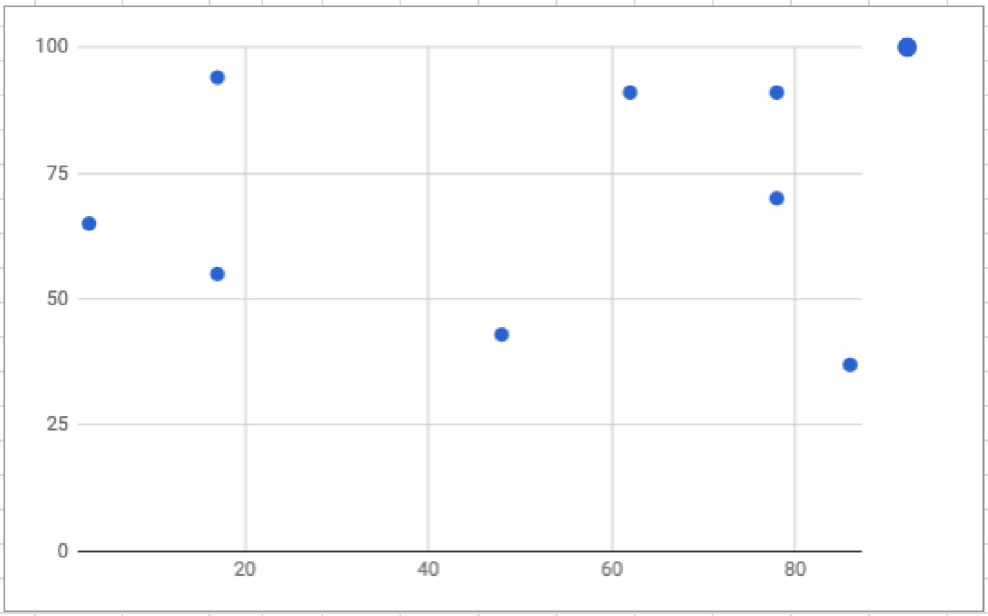
ACTIVITY: **Homework 3**

PROFESSOR: **Roger Zirahuén Ríos Mercados**

OP.	STUDENT ID	NAME	CLASS HOUR	Group	CAREER
1	1930968	Alejandro Sebastian Carranza Rodríguez	M4-M6	001	ITS

Data of the problem

	coordenadas										
	x	y		1	2	3	4	5	6	7	8
1	86	37	1	0							
2	17	94	2	89	0						
3	3	65	3	87	32	0					
4	48	43	4	38	59	50	0				
5	78	70	5	33	65	75	40	0			
6	17	55	6	71	39	17	33	62	0		
7	62	91	7	59	45	64	50	26	57	0	
8	78	91	8	54	61	79	56	21	70	16	0



The attached figure shows the coordinates and distance matrix of an 8-city symmetric TSP. Points are shown graphically. The distance matrix are integer values.

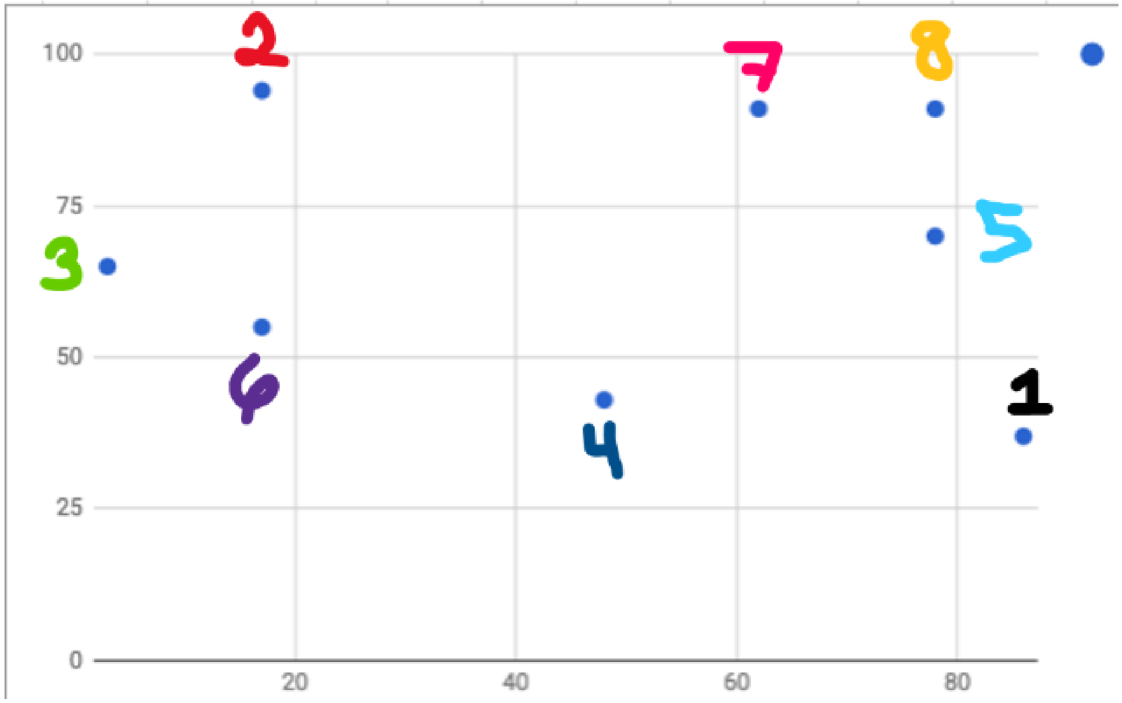
Nearest Neighbor Heuristic

To solve the problem using the nearest neighbor heuristic, we'll have to follow the next steps:

- Pick a random city.
- Go to the unvisited city closest to the last city picked.
- We do this until we finish the tour, then we calculate the total travel distance.
- Calculate all the possible instances and pick the one with the lowest total travel distance value.

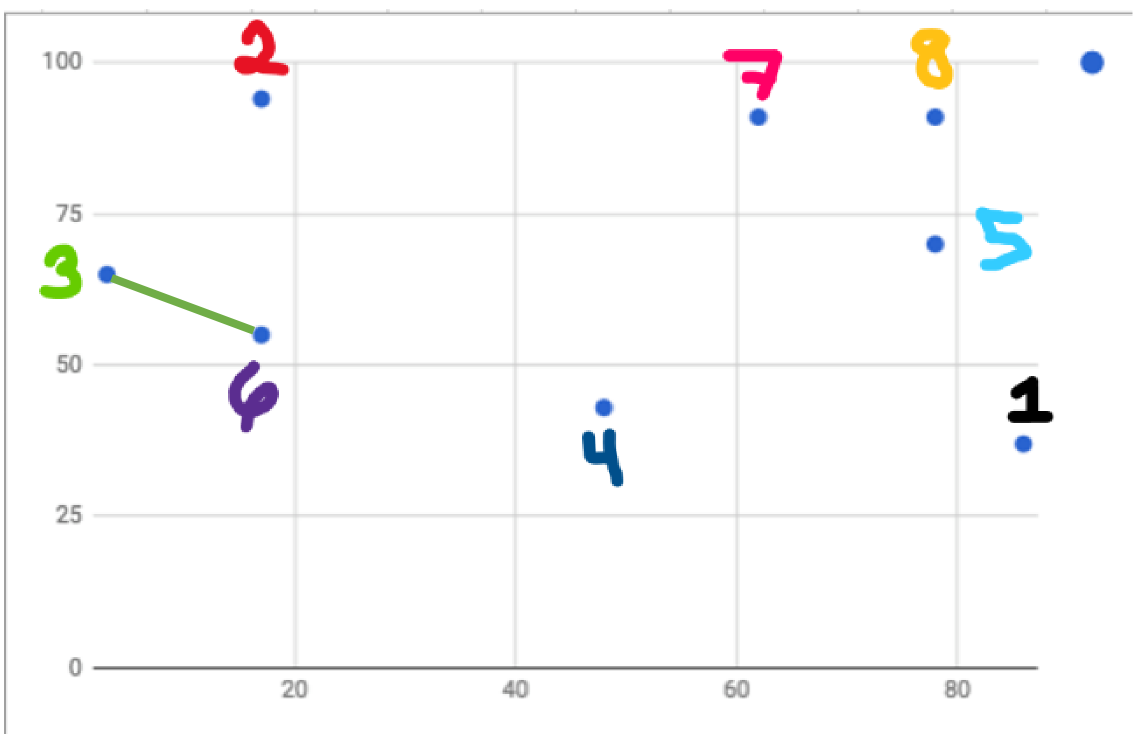
Development

First, it's important to mark the cities with their respective number id to get a better understanding of the problem and avoid mistakes, our "map" now would look like this:



First Iteration

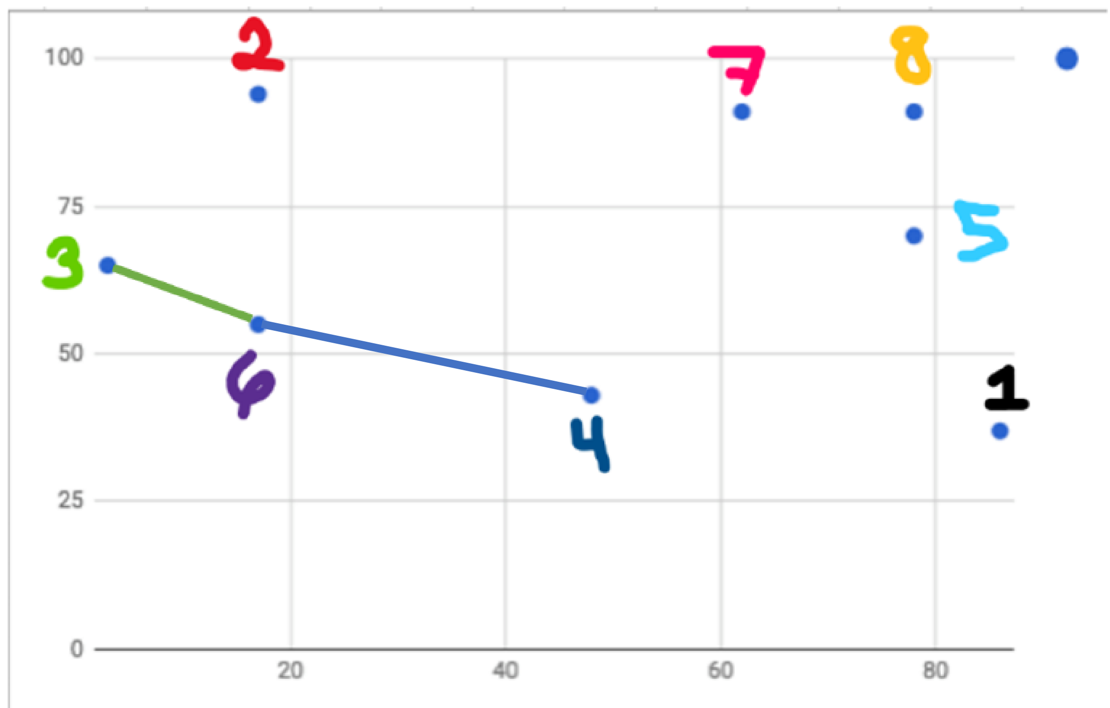
Now to start the algorithm we need to pick a start point city, it can be random the number, it doesn't matter so for this instance we'll pick the city **number 3**, so in base of the data that we knew the city **number 6** is the closest one, so we mark the route **3**-**6**.



Before ending this iteration, we calculated the objective function which is: **Travel distance** = 17.

Second Iteration

For the second iteration we repeated the process of the first iteration but in this case, we choose the closest city of the city **number 6** which is the city **number 4**, now our route will be **3**-**6**-**4**.

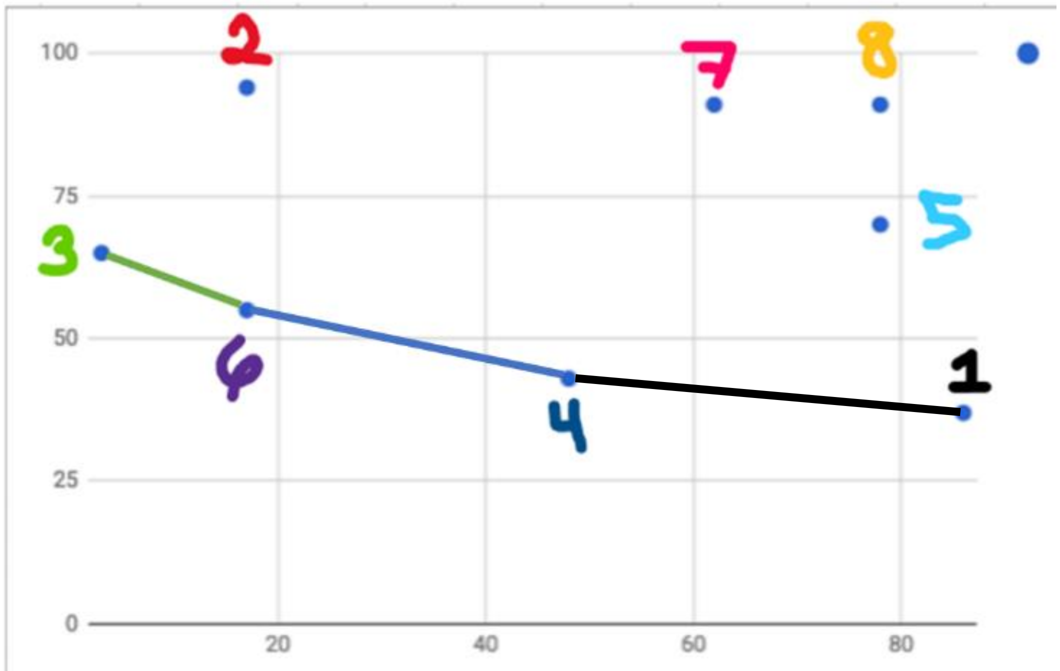


Before ending this iteration, we calculated the objective function which is: **Travel distance** = 17+33.

Third Iteration

For the third iteration we repeated the process of the first iteration but in this case, we choose the closest city of city the **number 4** which is the city **number 1**, now our route will be **3**-**6**-**4**-**1**.

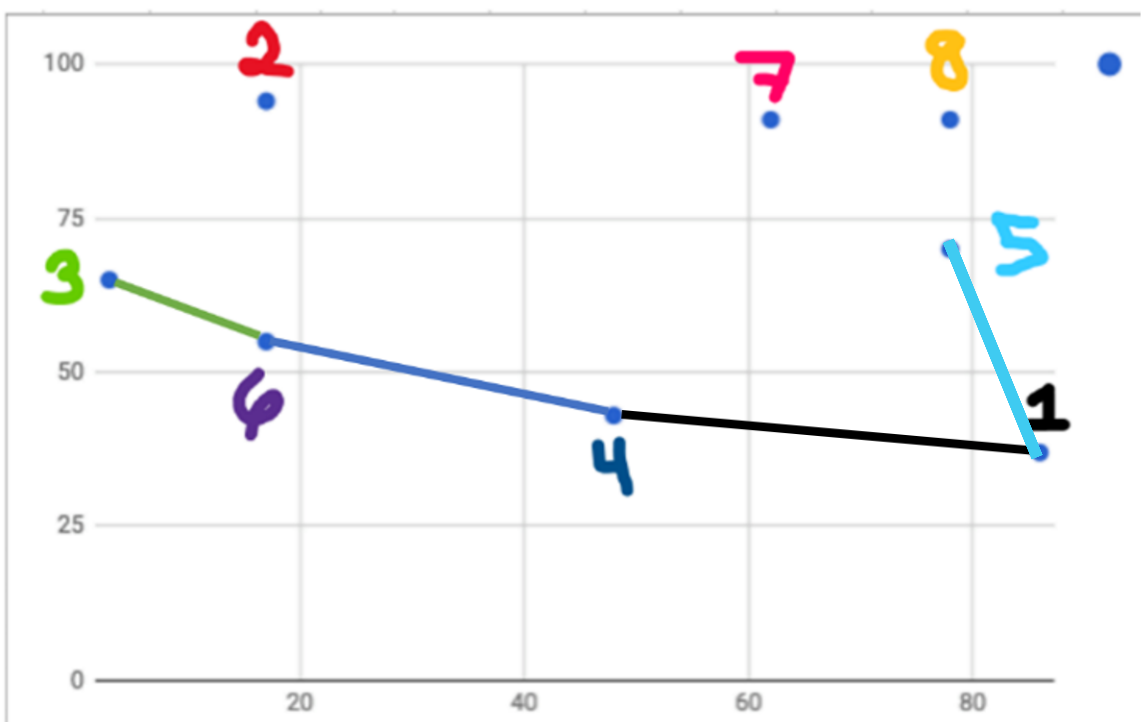
Before ending this iteration, we calculated the objective function which is: **Travel distance** = 17+33+38.



Fourth Iteration

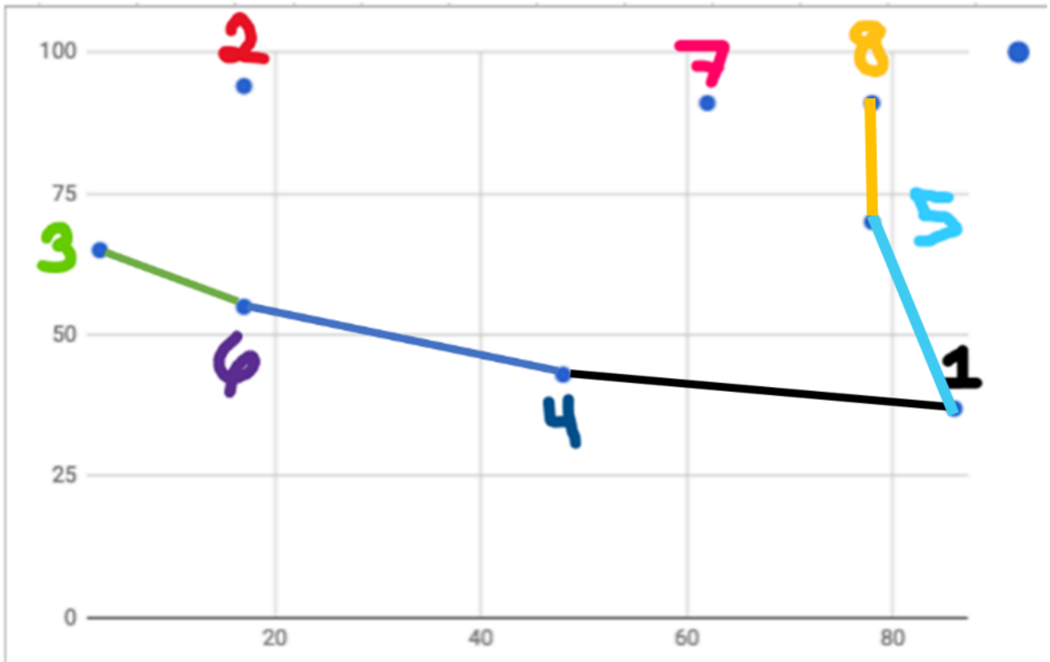
For the fourth iteration we repeated the process of the first iteration but in this case, we choose the closest city of city the **number 1** which is the city **number 5**, now our route will be **3-6-4-1-5**.

Before ending this iteration, we calculated the objective function which is:
Travel distance = 17+33+38+33.



Fifth Iteration

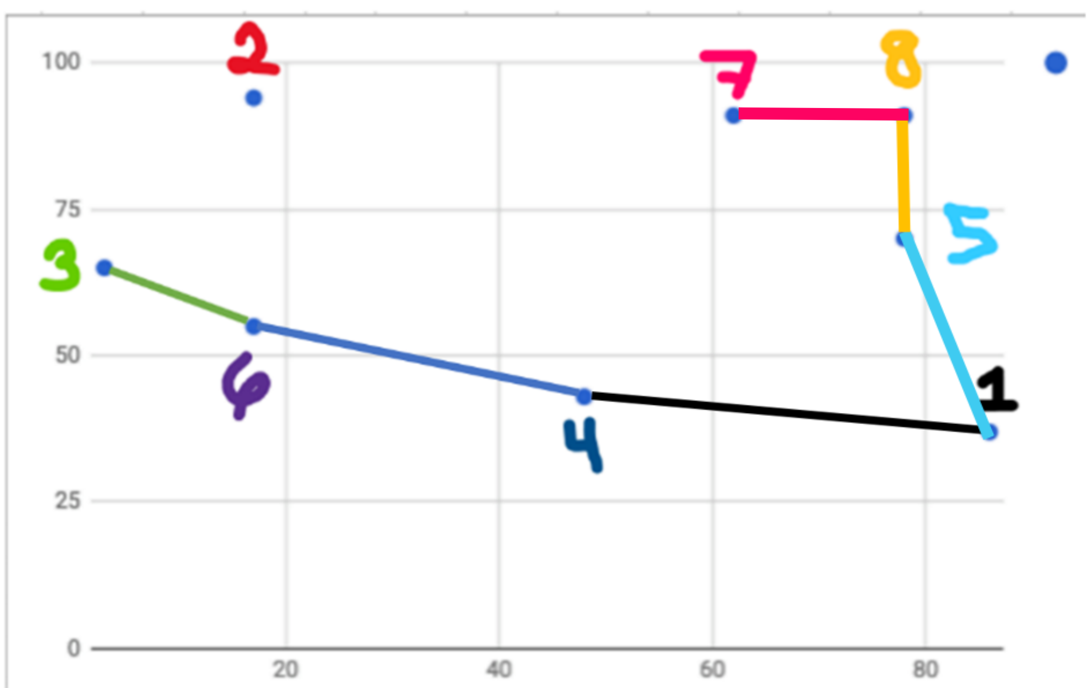
For the fifth iteration we repeated the process of the first iteration but in this case, we choose the closest city of city the **number 5** which is the city **number 8**, now our route will be **3**-**6**-**4**-**1**-**5**-**8**.



Before ending this iteration, we calculated the objective function which is:
Travel distance = 17+33+38+33+21.

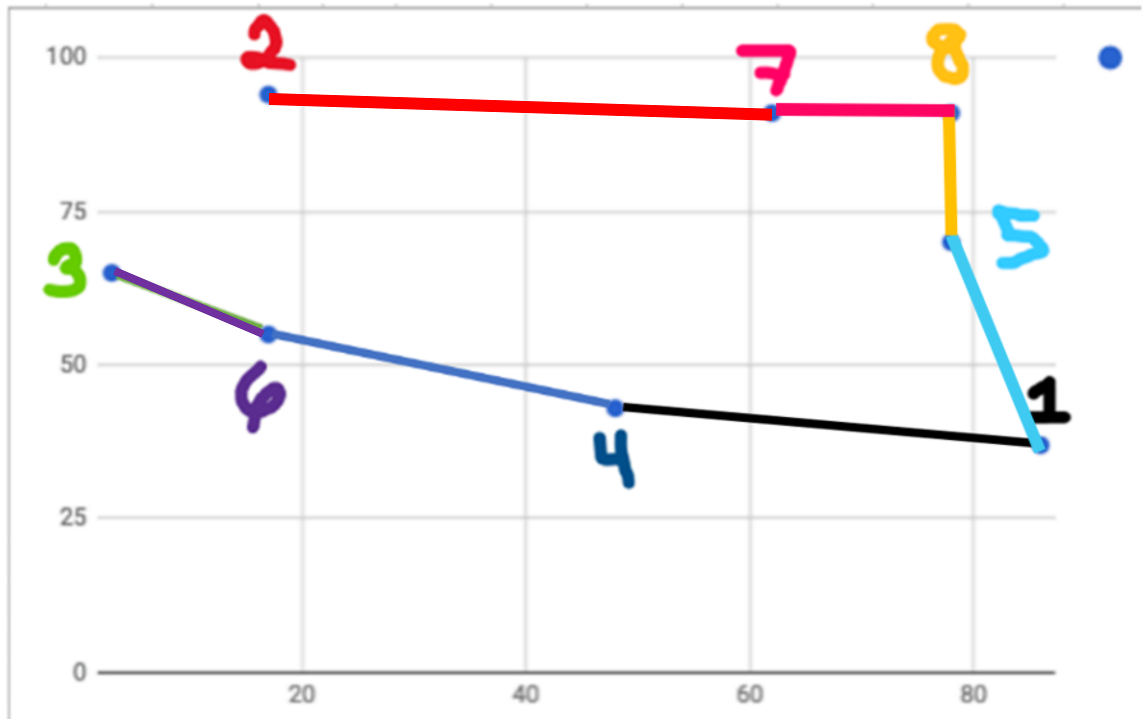
Sixth Iteration

For the sixth iteration we repeated the process of the first iteration but in this case, we choose the closest city of city the **number 8**which is the city **number 7**, now our route will be **3**-**6**-**4**-**1**-**5**-**8**-**7**.



Before ending this iteration, we calculated the objective function which is:
Travel distance = 17+33+38+33+21+16.

<u>Seventh Iteration</u>
For the seventh iteration we repeated the process of the first iteration but in this case, we choose the closest city of city the **number 7**which is the city **number 2**, now our route will be **3**-**6**-**4**-**1**-**5**-**8**-**7**-**2**.



Before ending this iteration, we calculated the objective function which is:
Travel distance = 17+33+38+33+21+16+45.

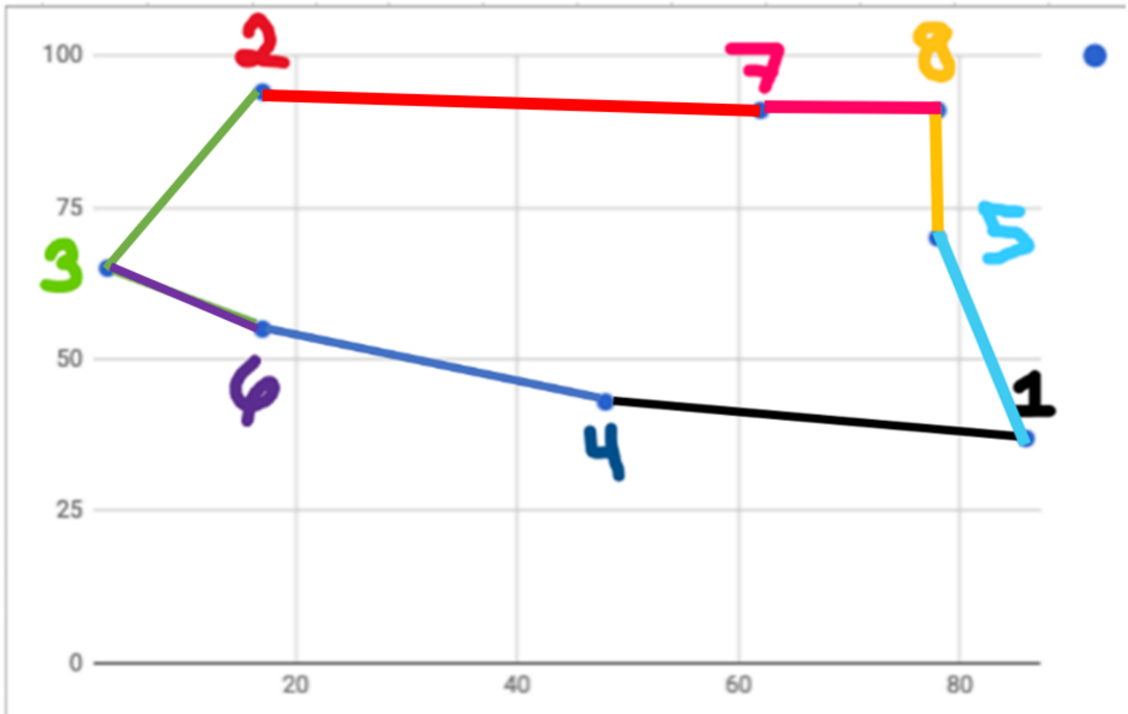
<u>Eight Iteration</u>
For the last iteration we repeated the process of the first iteration but in this case, we choose the closest city of city the **number 2** which is the city **number 3**, now our tour will be **3**-**6**-**4**-**1**-**5**-**8**-**7**-**2**-**3**.

Before ending this iteration, we calculated the objective function which is:
Travel distance = 17+33+38+33+21+16+45+32.

Now we calculated the total travel distance because our tour its completed.

Total Travel distance = 235

Our tour will look like this:



For this case the total travel distance is 235, it's a feasible solution because it met all the constraints.

In this specific case all the possible tours are going to give us the same total travel distance because it's a symmetrical problem but in asymmetrical problems we will need to calculate all the possible instances to get the optimal solution.

Nearest Insertion Heuristic

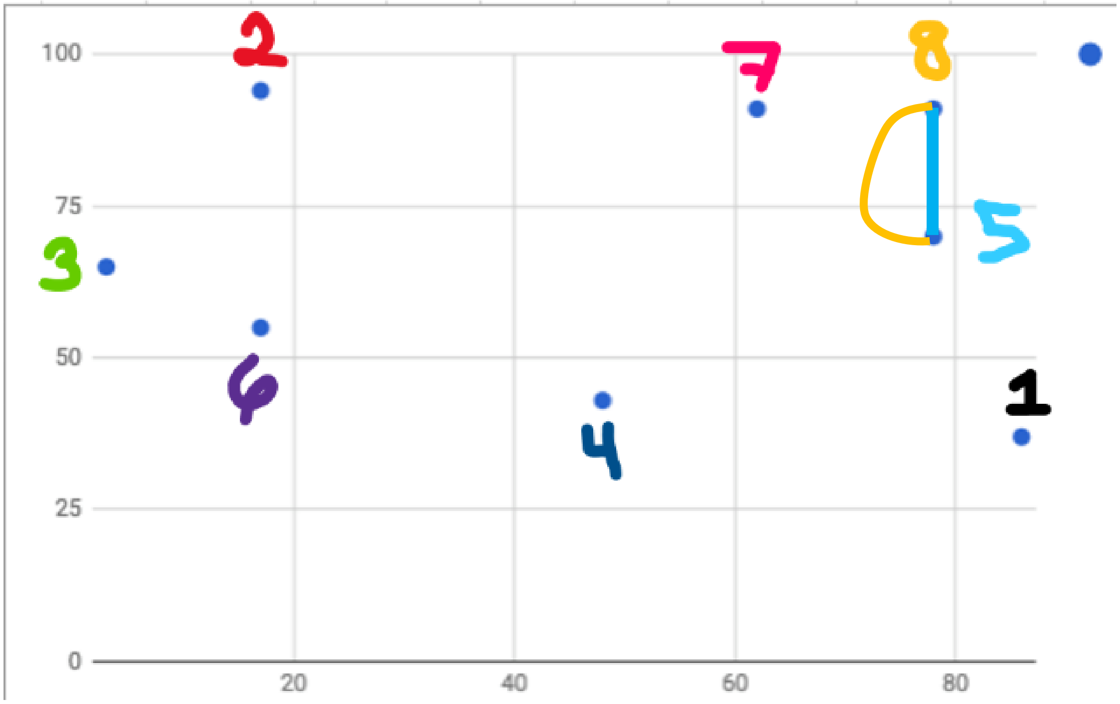
To solve the problem using the nearest insertion heuristic, we'll have to follow the next steps:

- Pick a random starting city and add it to the subtour.
- Search the city closest to the starting city and add it to the subtour.
- For each subsequent city not yet added to the subtour, find the pair of cities in the current subtour that are closest to it.
- Select the pair of cities that minimize the increase of total distance when the new city is inserted between them. (the formula to calculate this is $C_{ik} + C_{jk} - C_{ij}$).
- Insert the new city between the selected pair of cities.
- It's necessary to repeat steps 3-5 until all cities are connected.
- For last we calculated the total travel distance of the tour.
- Calculate all the possible instances and pick the one with the lowest total travel distance value.

First Iteration

In this example we are going to pick the city **number 5** so in base of the data, the city **number 8** is the closest one.

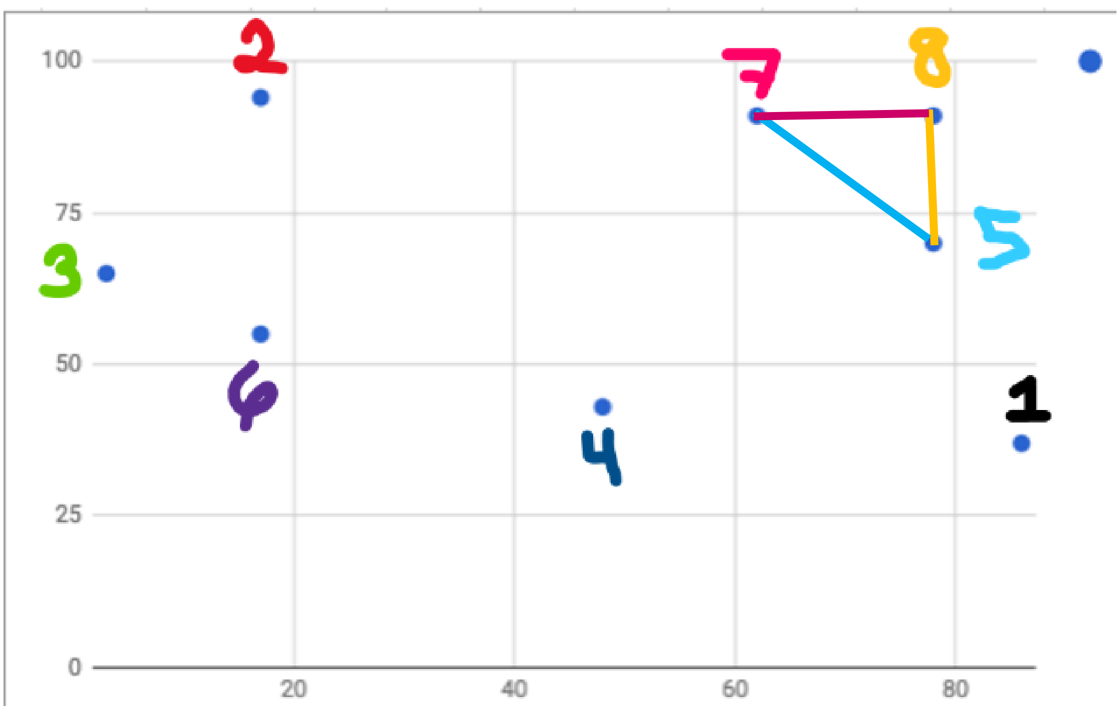
So now our subtour will be **5**-**8**-**5**.



Second Iteration

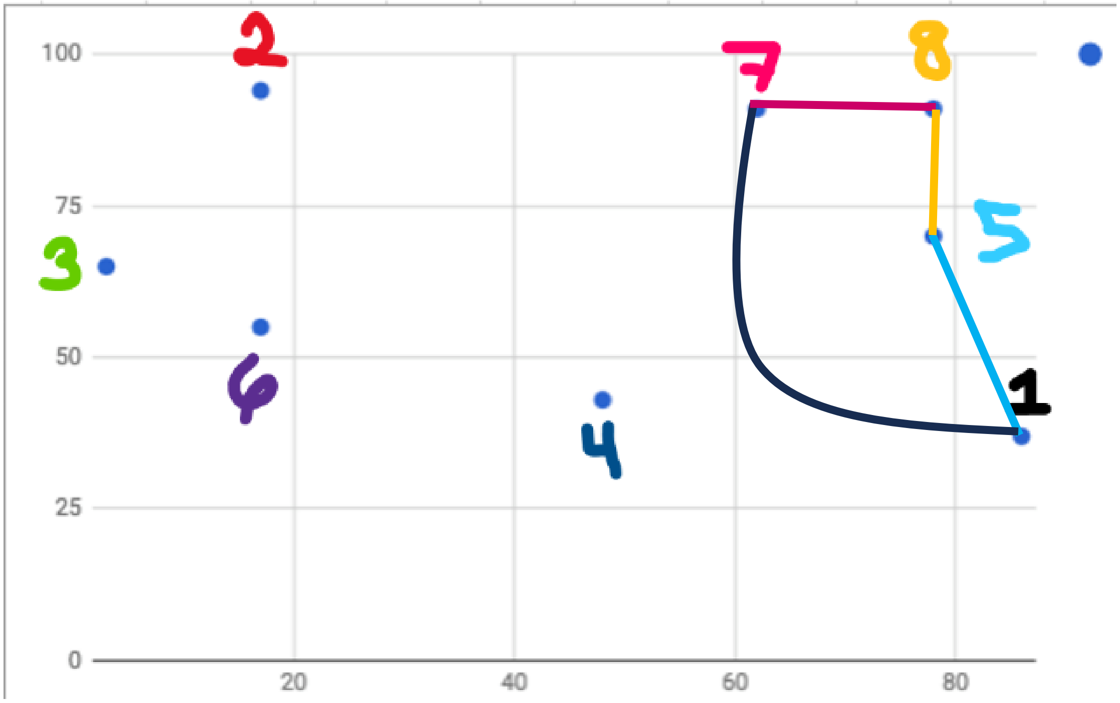
For this second iteration we are going to use the formula of step 4 which is $C_{ik} + C_{jk} - C_{ij}$ to calculate the optimal city to be inserted between the subtour (in the formula **j** and **i** represent a city in the subtour, k is the closest city of the subtour). $\mathbf{C_{57}+C_{87}-C_{58}} =$ 26+16-21=17

in this case the city **number 7** will be the one inserted. So now our subtour will be **5**-**7**-**8**-**5**.



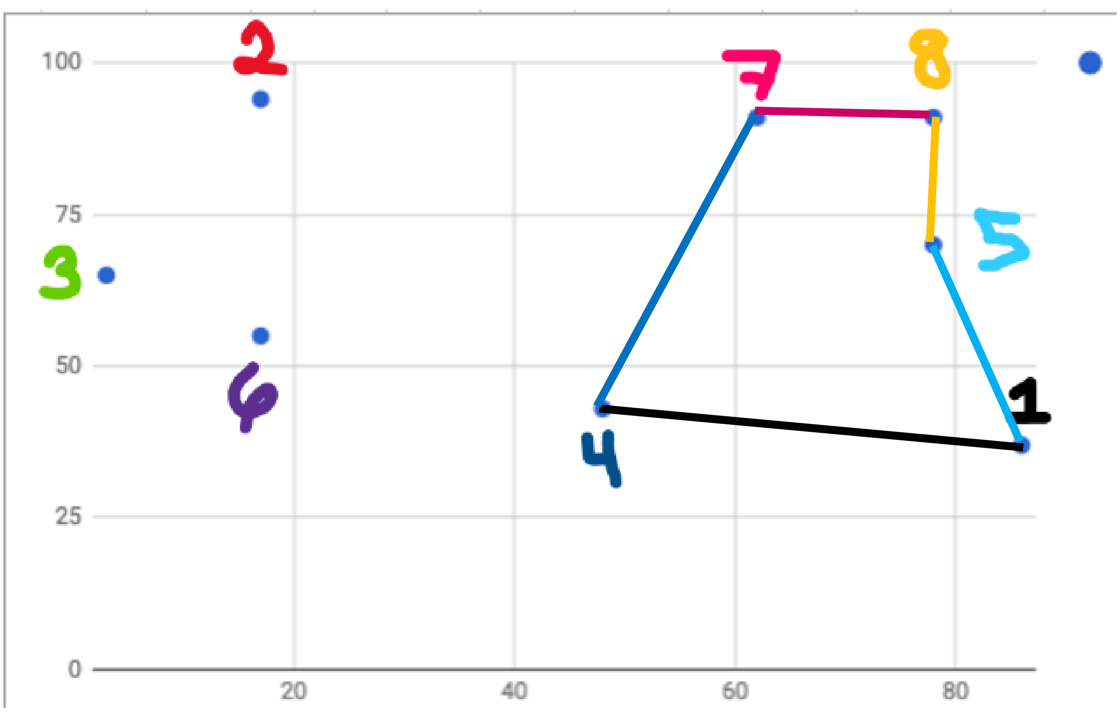
Third Iteration

We repeat the process of the second iteration, so in this case the city **number 1** (**C_{51}+C_{71}-C_{57} =** 33+59-26=66) is the one inserted. Now our subtour will be **5**-**1**-**7**-**8**-**5**.



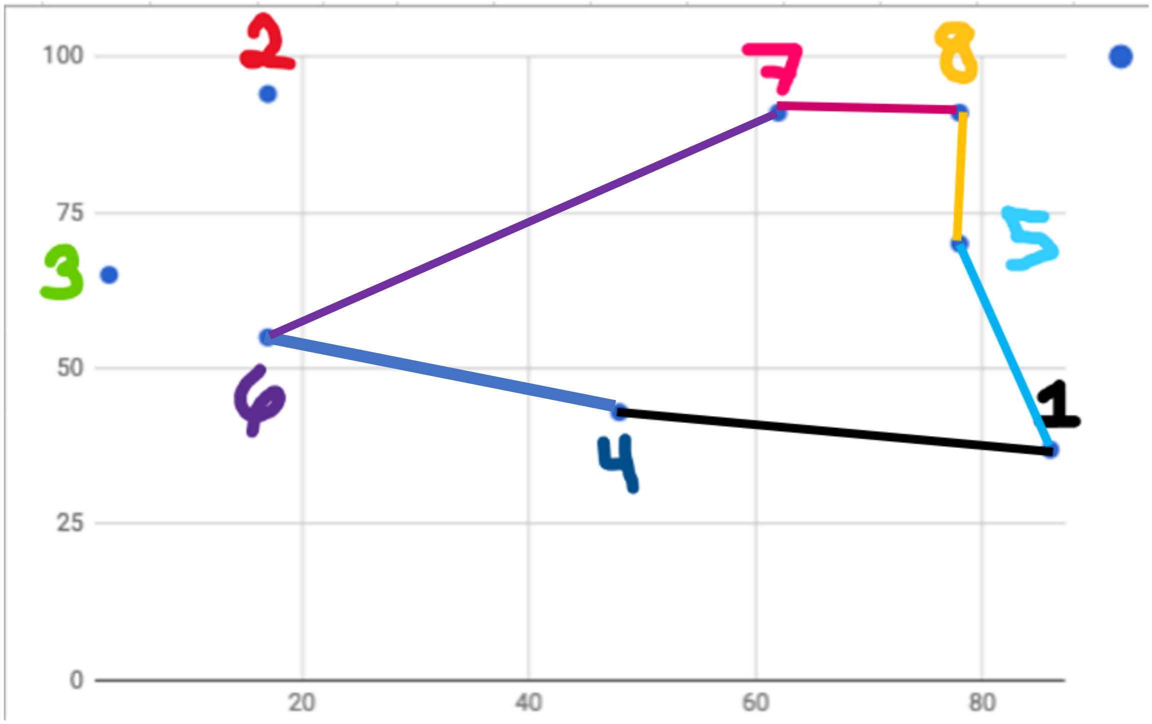
Fourth Iteration

We repeat the process of the second iteration, so in this case the city **number 4** (**C_{14}+C_{74}-C_{17} =** 38+50-59=29) is the one inserted. Now our subtour will be **5**-**1**-**4**-**7**-**8**-**5**.



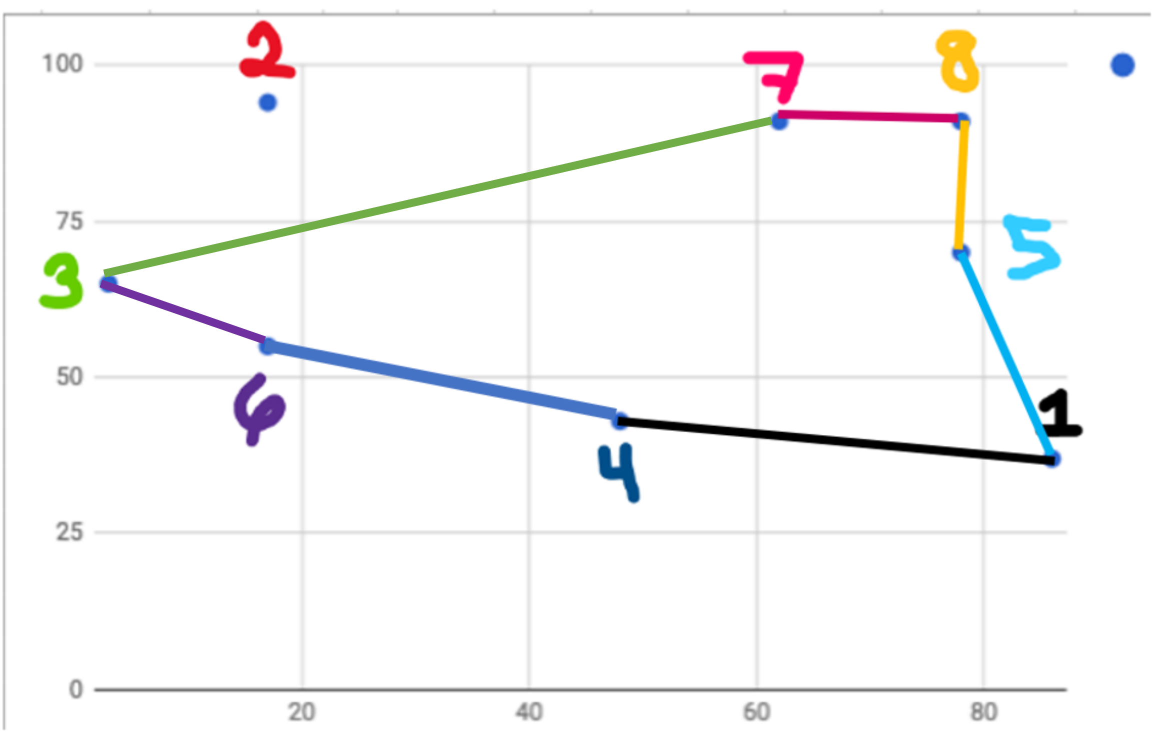
Fifth Iteration

We repeat the process of the second iteration, so in this case the city **number 6** (**C_{46}+C_{76}-C_{47} =** 33+57-50=40) is the one inserted. Now our subtour will be **5**-**1**-**4**-**6**-**7**-**8**-**5**.



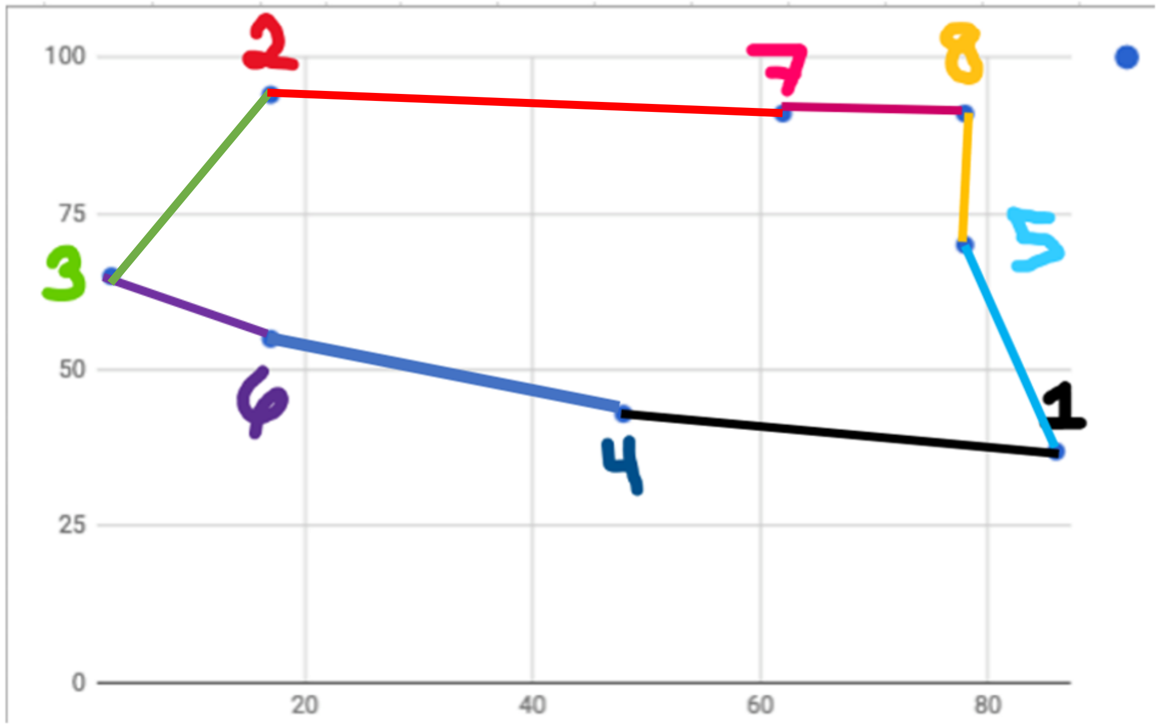
Sixth Iteration

We repeat the process of the second iteration, so in this case the city **number 3** (**C_{63}+C_{73}-C_{67} =** 33+64-57=40) is the one inserted. Now our subtour will be **5**-**1**-**4**-**6**-**3**-**7**-**8**-**5**.



Seventh Iteration

We repeat the process of the second iteration, so in this case the city **number 2** (**C_{32}+C_{72}-C_{37} =** 32+45-64=13) is the one inserted. Now our tour will be **5-1-4-6-3-2-7-8-5.**



Tour: 5-1-4-6-3-2-7-8-5.

Now that we have the tour, we calculated the total distance traveled:
Total distance traveled: 33+38+50+17+32+45+16+21 = 235

For this case the total travel distance is 235, it's a feasible solution because it met all the constraints.

We already know that all the possible tours are going to give us the same total traveled distance because it's a symmetrical problem but in asymmetrical problems we will need to calculate all the possible instances to get the optimal solution.