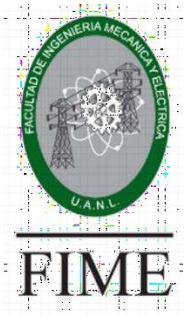
UNIVERSIDAD AUTONOMA DE NUEVO LEON

Facultad de Ingeniería Mecánica y Eléctrica



Temas Selectos de Optimización

ROGER ZIRAHUEN RIOS MERCADO

Homework 3

Saúl Emmanuel Carreón Alanís

Student number: 1901555 Career: ITS

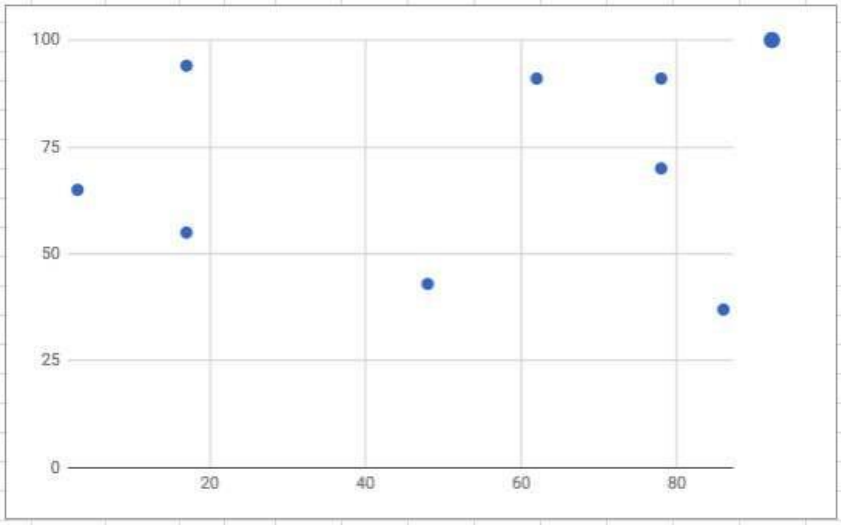
Semester January – August2024

Data

coordenadas

	x	y
1	86	37
2	17	94
3	3	65
4	48	43
5	78	70
6	17	55
7	62	91
8	78	91

	1	2	3	4	5	6	7	8
1	0							
2	89	0						
3	87	32	0					
4	38	59	50	0				
5	33	65	75	40	0			
6	71	39	17	33	62	0		
7	59	45	64	50	26	57	0	
8	54	61	79	56	21	70	16	0



Goal:

Apply both the nearest neighbor heuristic and the nearest insertion heuristic.

Nearest Neighbor Heuristic

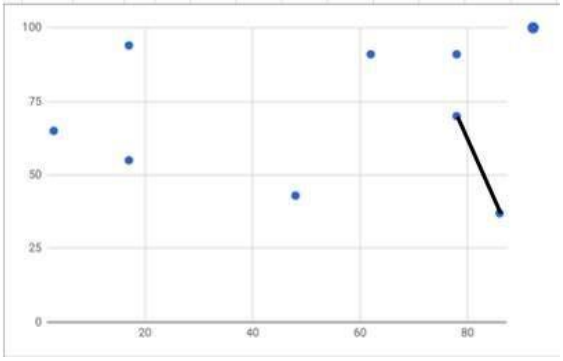
To solve the problem using the nearest neighbor heuristic, you'll have to follow the next steps:

1. Pick a random city.
2. Go to the unvisited city closest to the last city picked.
3. Do this until you finish the tour, then add the total distance.
4. Repeat this for every city and take the best tour found (the one with the lowest number).

Procedure

To do this the algorithm would have to pick a random city, for instance, city number 1. The distance matrix already gives us the distance between each city. Analyzing this information, the algorithm would pick the closest city to the first one which is number 5:

	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

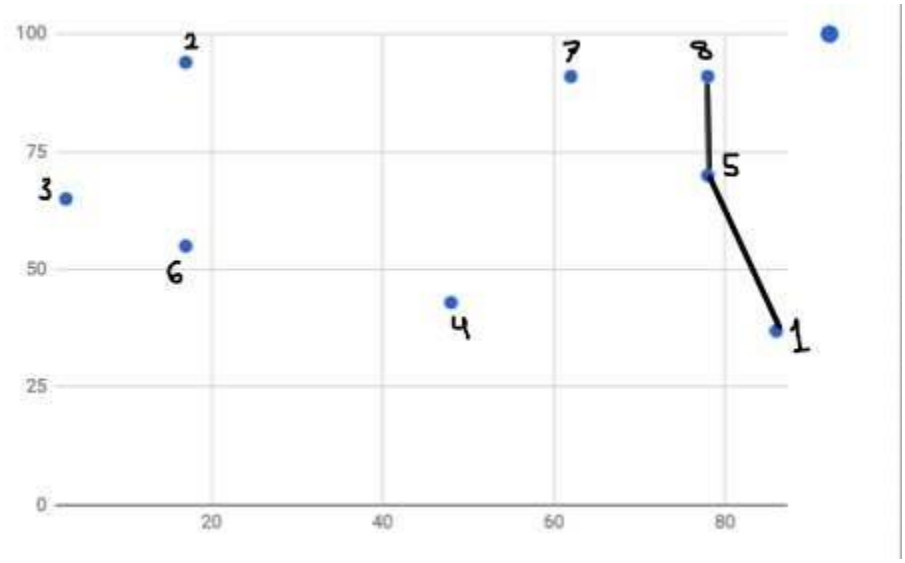


Right now, the tour is 1−5, we'll do this for every unvisited city and calculate the total distance as well.

Total distance: 33

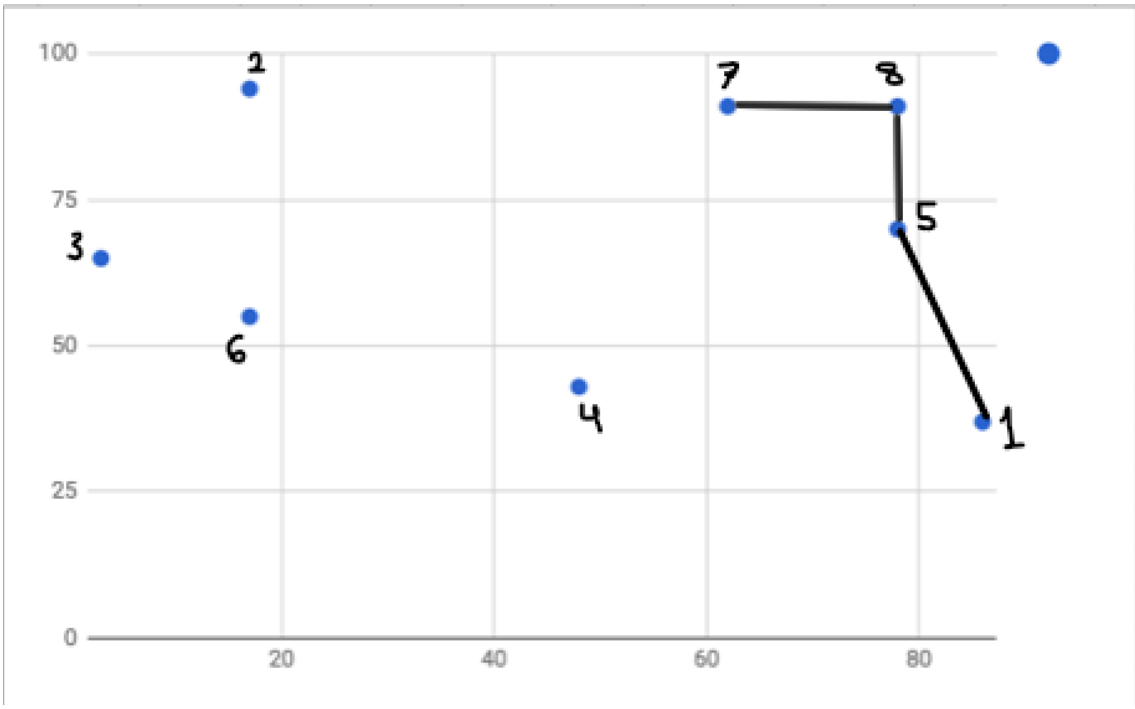
1−5−8

Total distance: 33+21



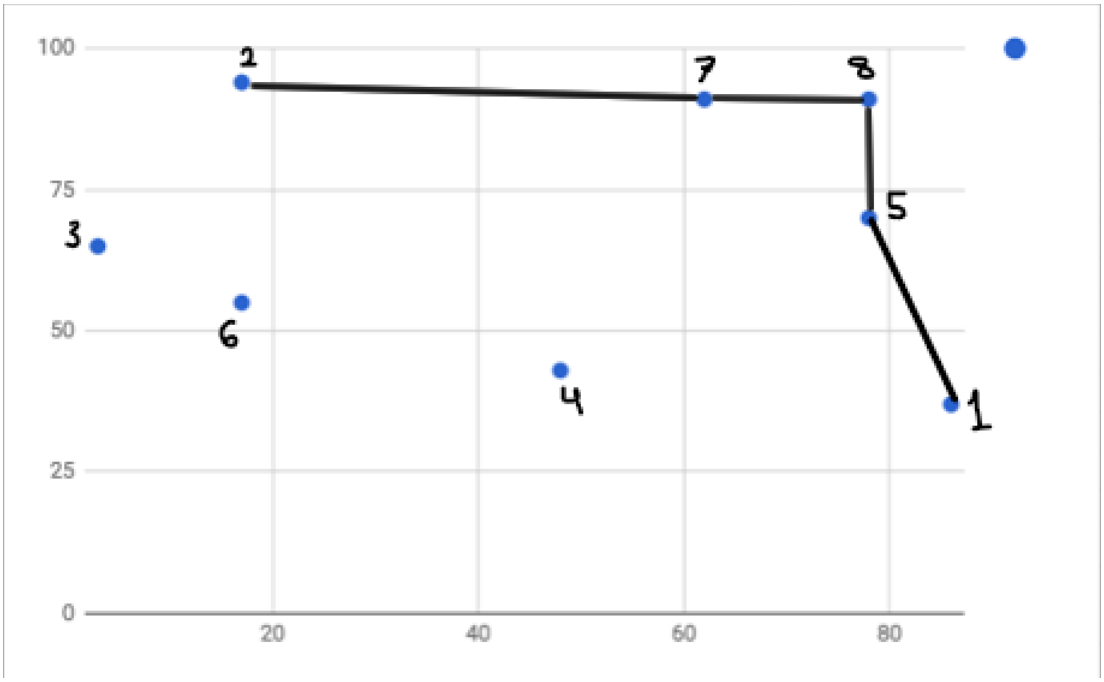
1−5−8−7

Total distance: 33+21+16



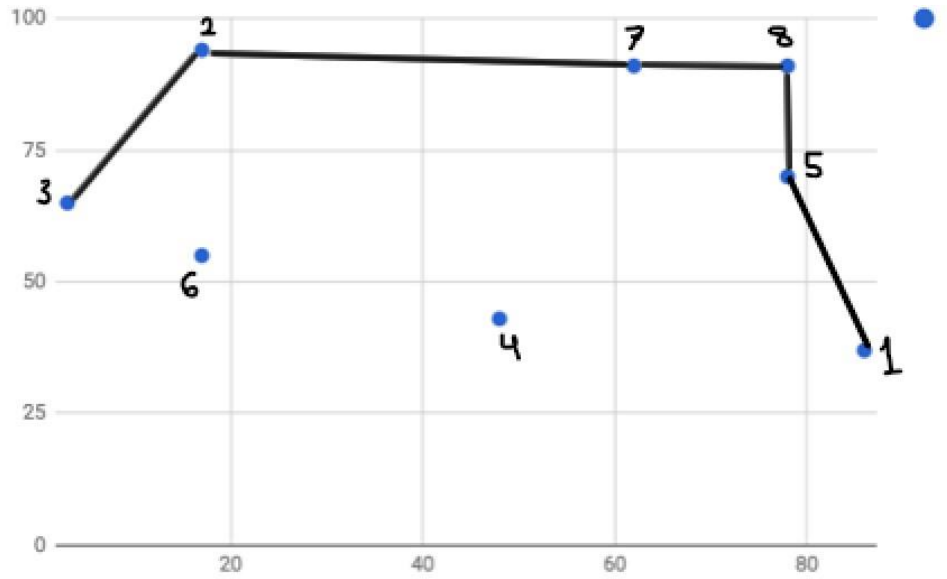
1−5−8−7−2

Total distance: 33+21+16+45



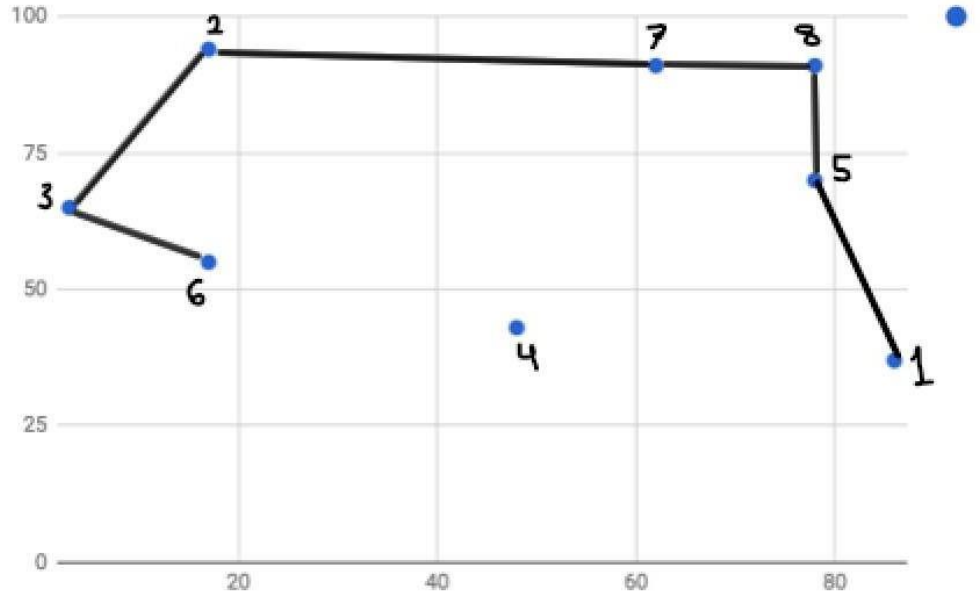
1–5–8–7–2–3

Total distance: 33+21+16+45+32



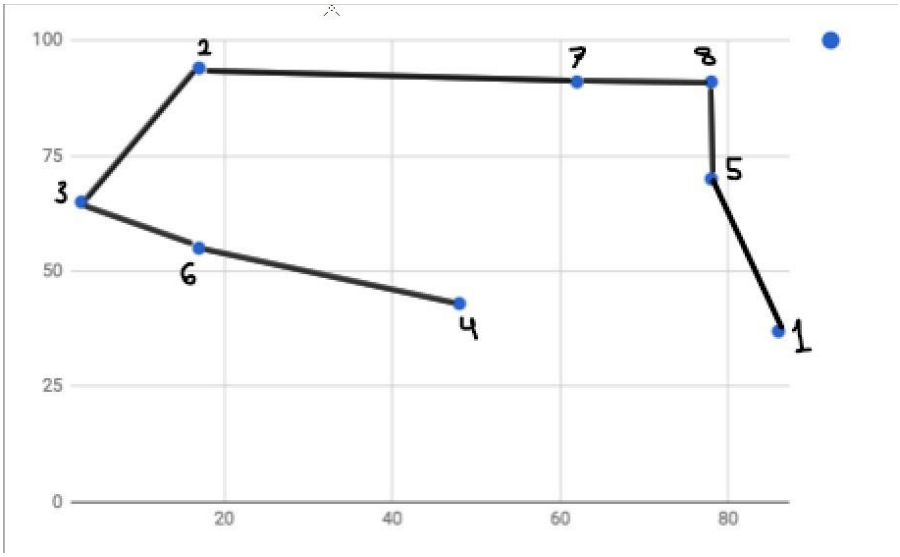
1–5–8–7–2–3–6

Total distance: 33+21+16+45+32+17



1–5–8–7–2–3–6–4

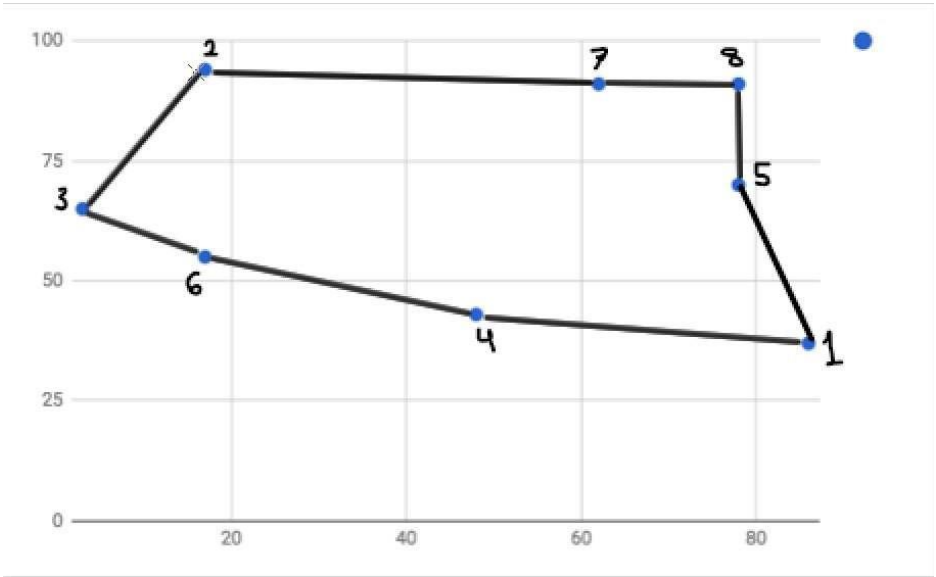
Total distance: 33+21+16+45+32+17+33



If there are no more unvisited cities, the algorithm must pick the first city to finish the tour.

1–5–8–7–2–3–6–4–1

Total distance: 33+21+16+45+32+17+33+38= 235

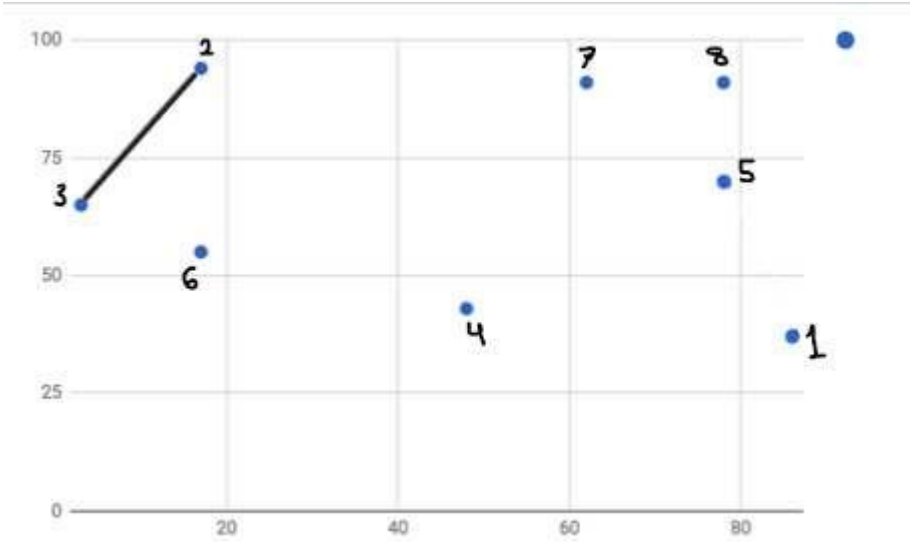


If you wanted to know the most optimal solution, the algorithm would have to do this starting from every single city:

City 2

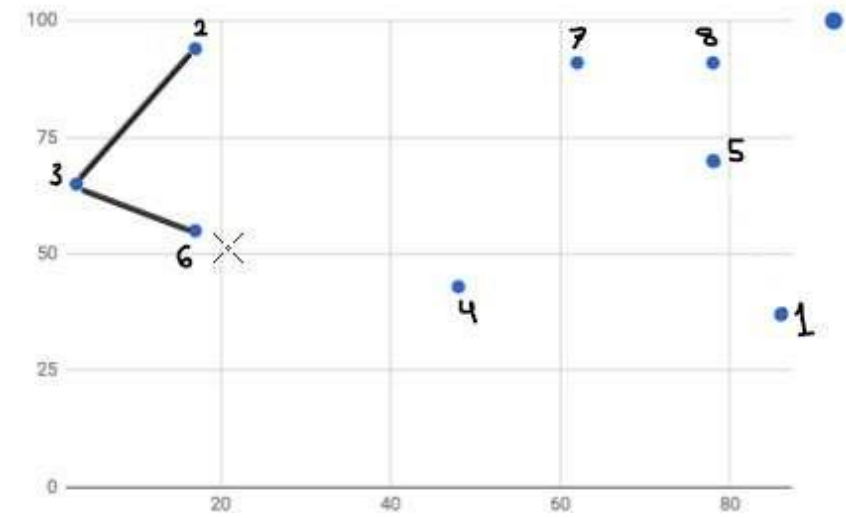
2−3

Total distance: 32



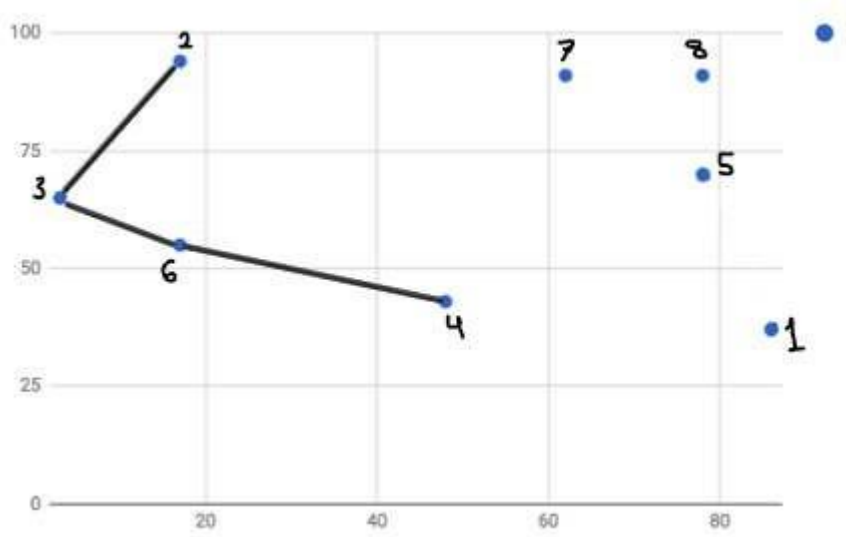
2−3−6

Total distance: 32+17



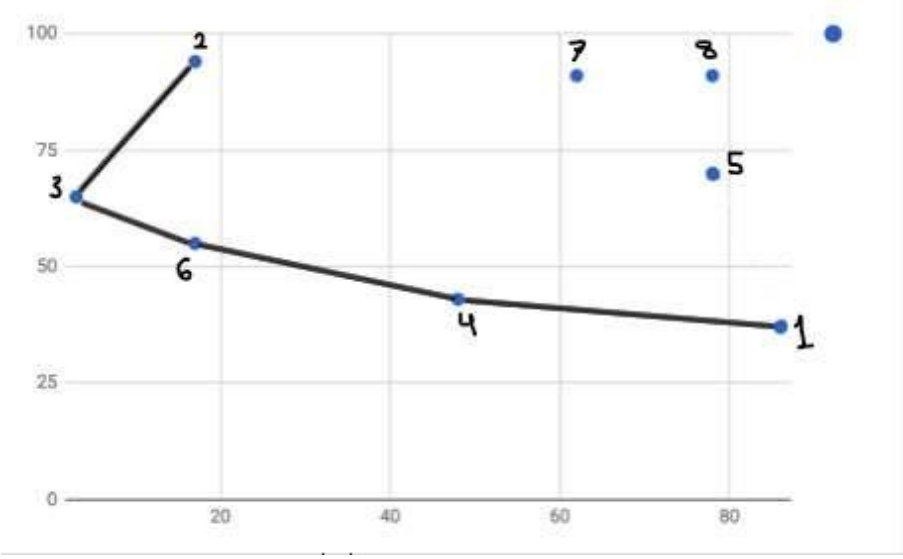
2−3−6−4

Total distance: 32+17+33



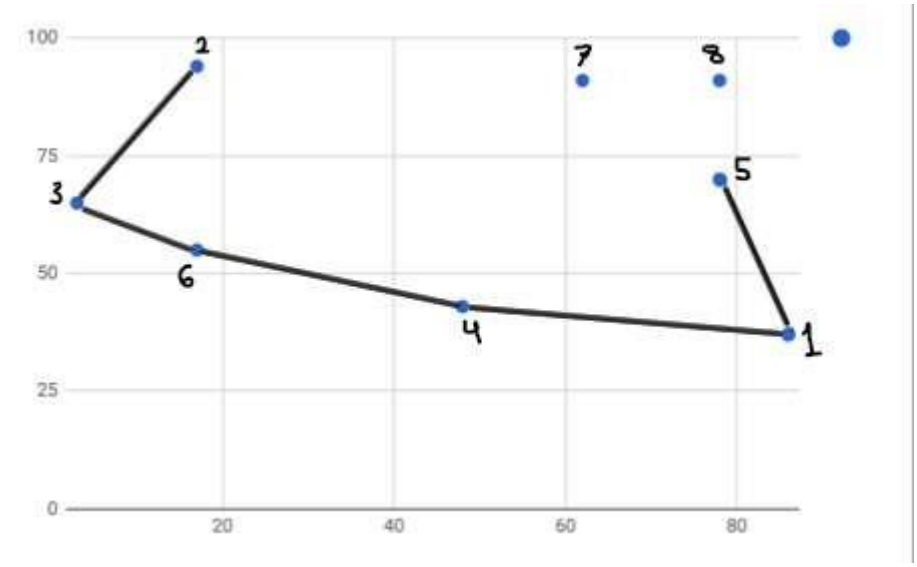
2−3−6−4−1

Total distance: 32+17+33+38



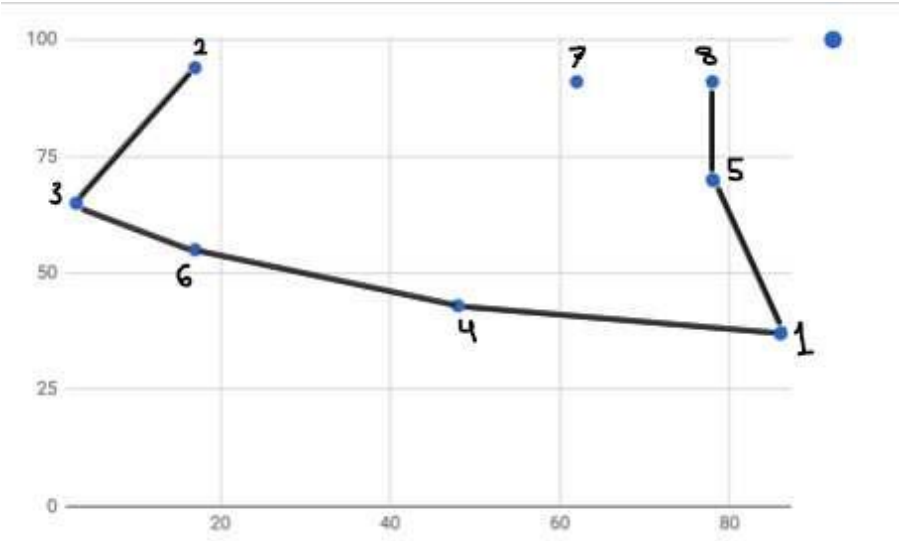
2−3−6−4−1−5

Total distance: 32+17+33+38+33



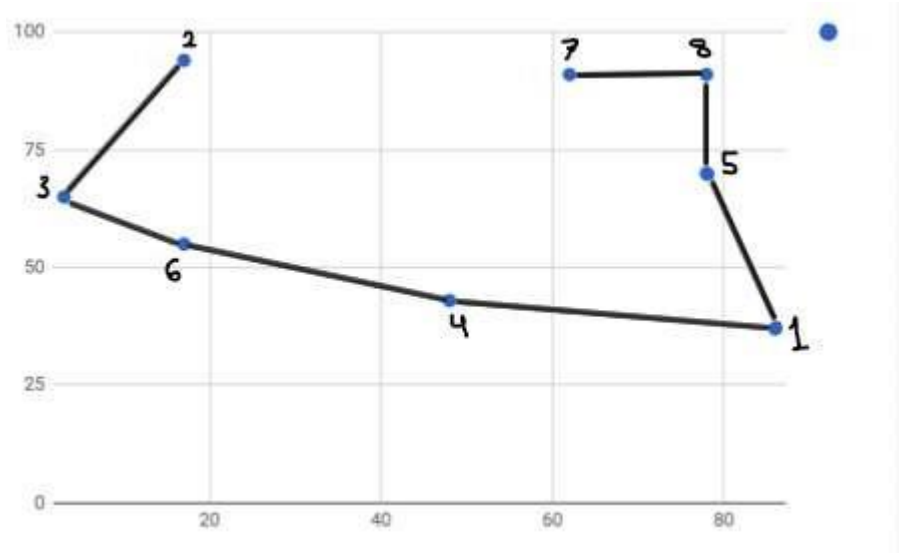
2−3−6−4−1−5−8

Total distance: 32+17+33+38+33+21



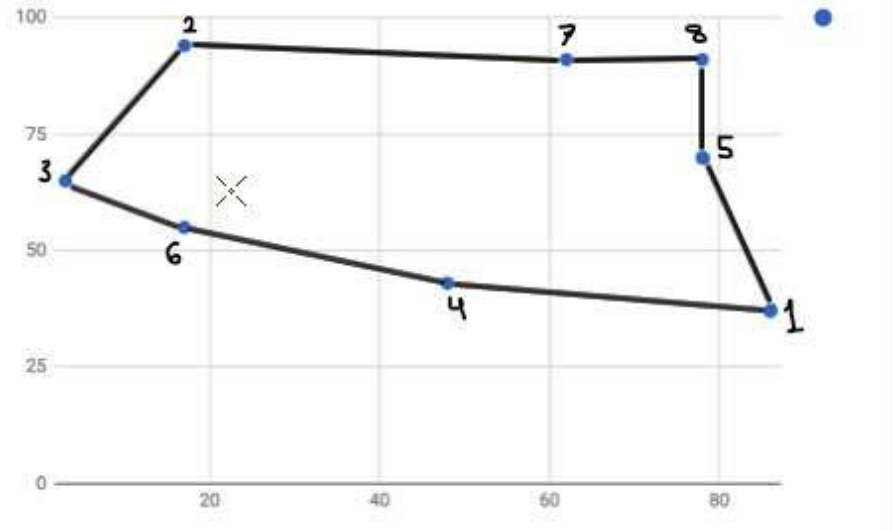
2–3–6–4–1–5–8–7

Total distance: 32+17+33+38+33+21+17



2–3–6–4–1–5–8–7–2

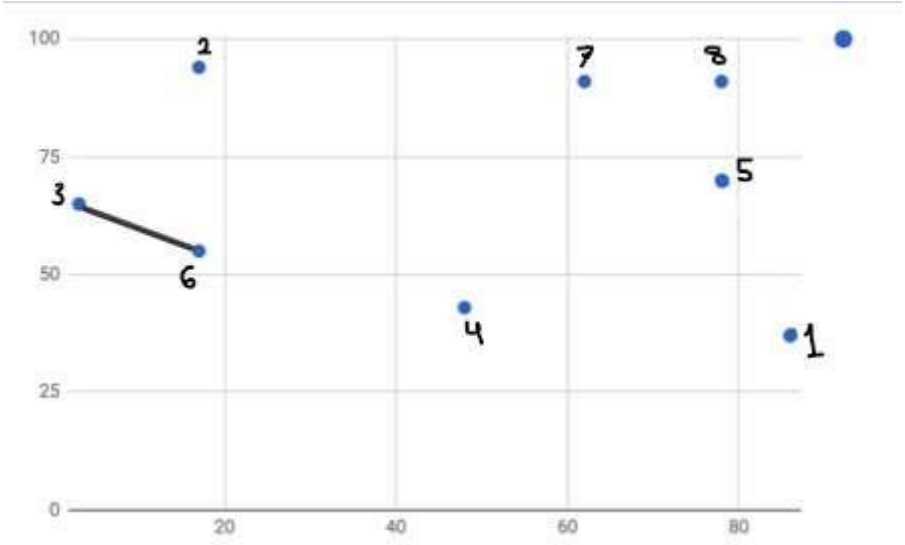
Total distance: 32+17+33+38+33+21+16+45 = 235



City 3

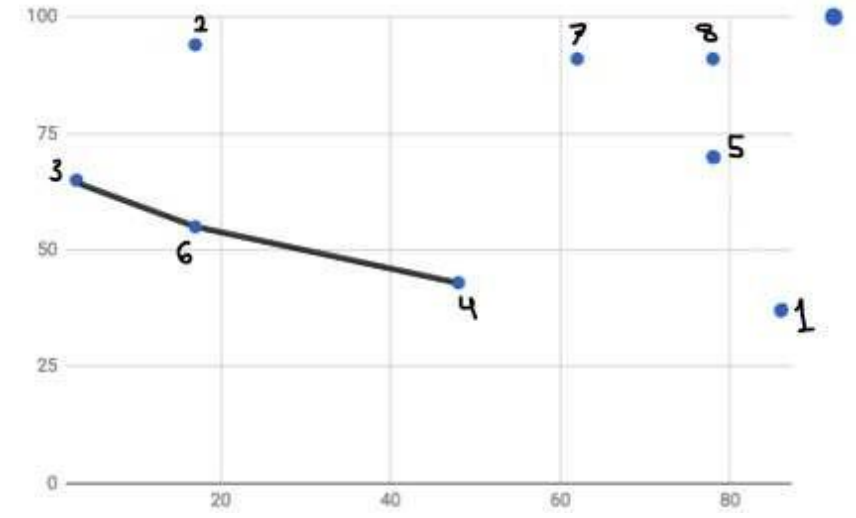
3−6

Total distance: 17



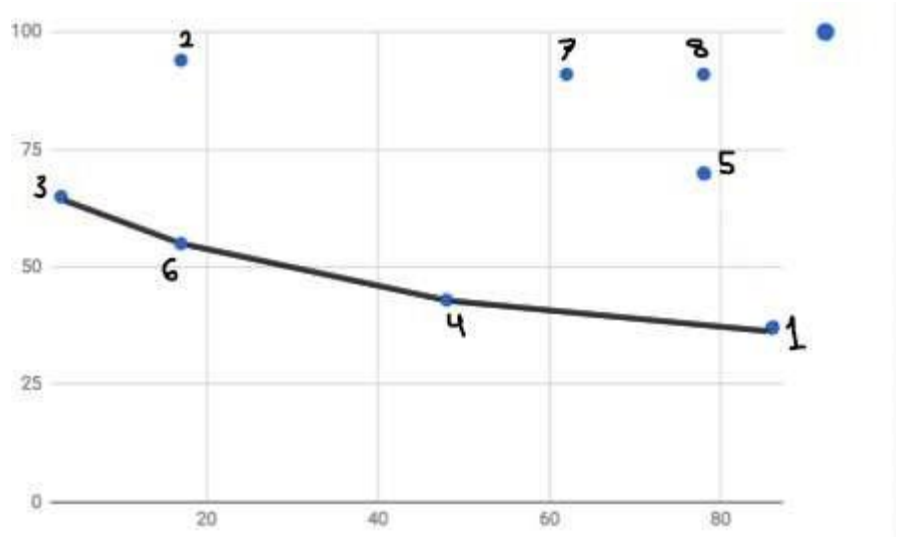
3−6−4

Total distance: 17+33



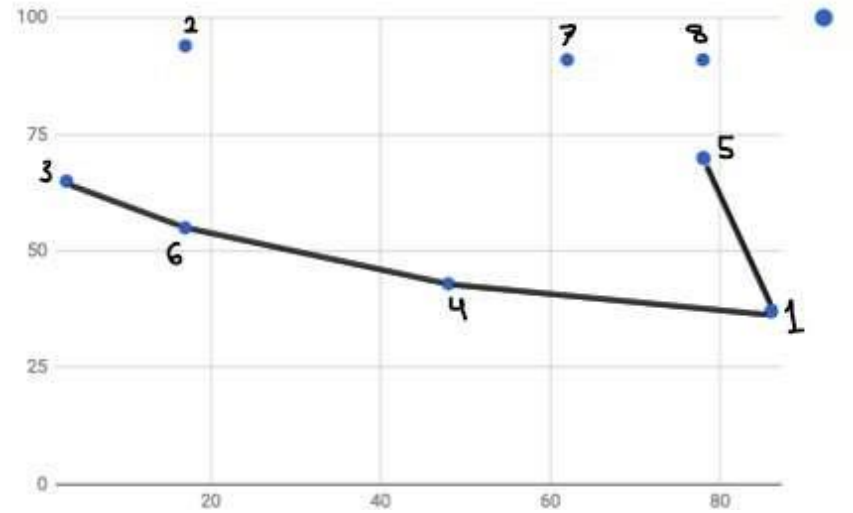
3−6−4−1

Total distance: 17+33+38



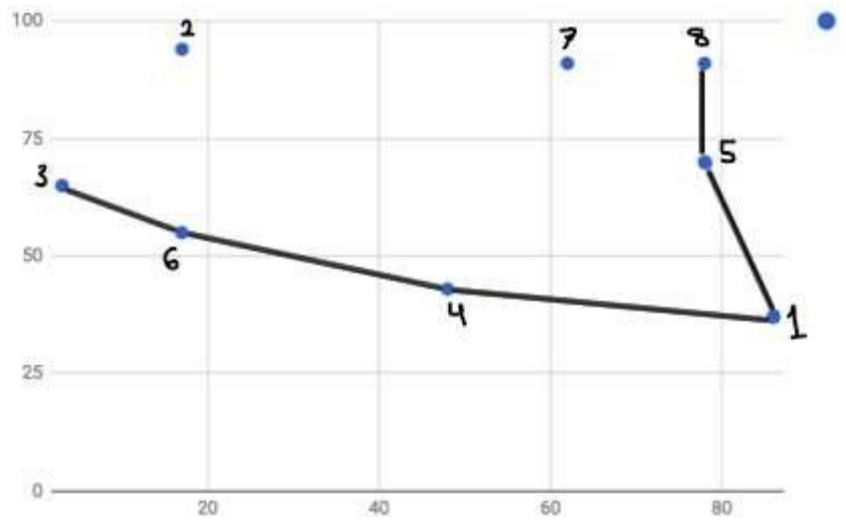
3−6−4−1−5

Total distance: 17+33+38+33



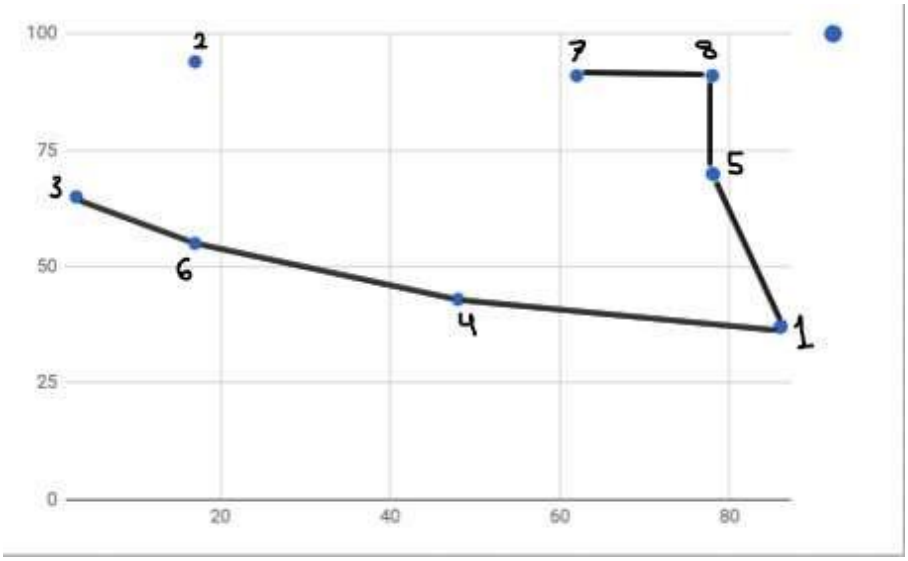
3–6–4–1–5–8

Total distance: 17+33+38+33+21



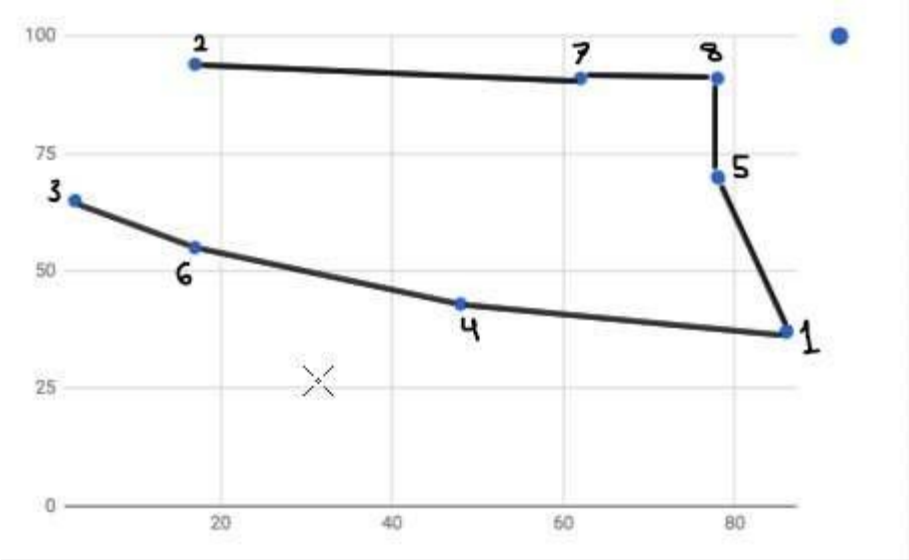
3–6–4–1–5–8–7

Total distance: 17+33+38+33+21+16



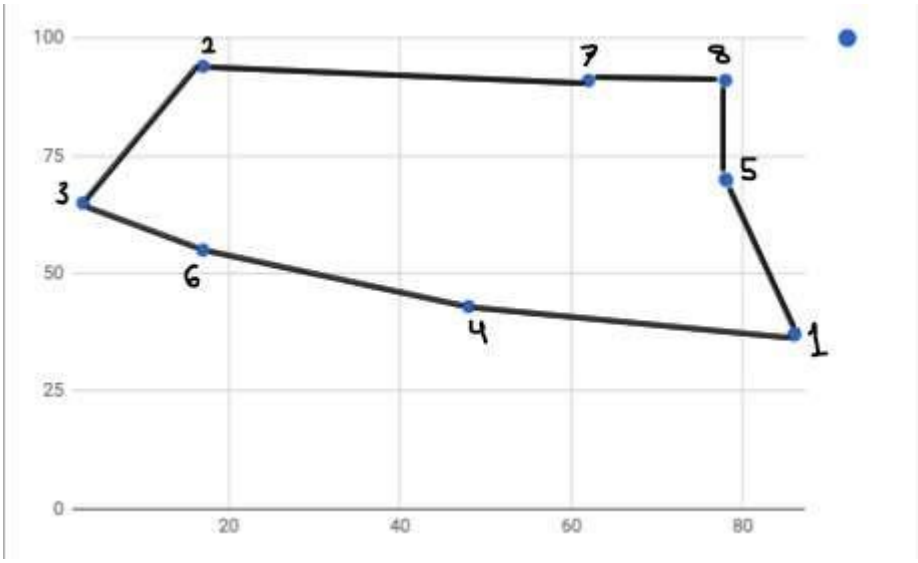
3–6–4–1–5–8–7–2

Total distance: 17+33+38+33+21+16+45



3–6–4–1–5–8–7–2–3

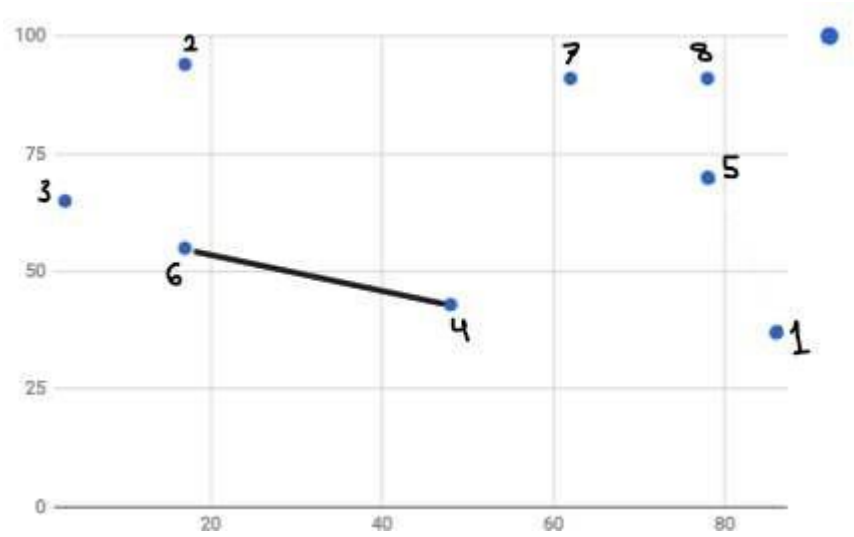
Total distance: 17+33+38+33+21+16+45+32= 235



City 4

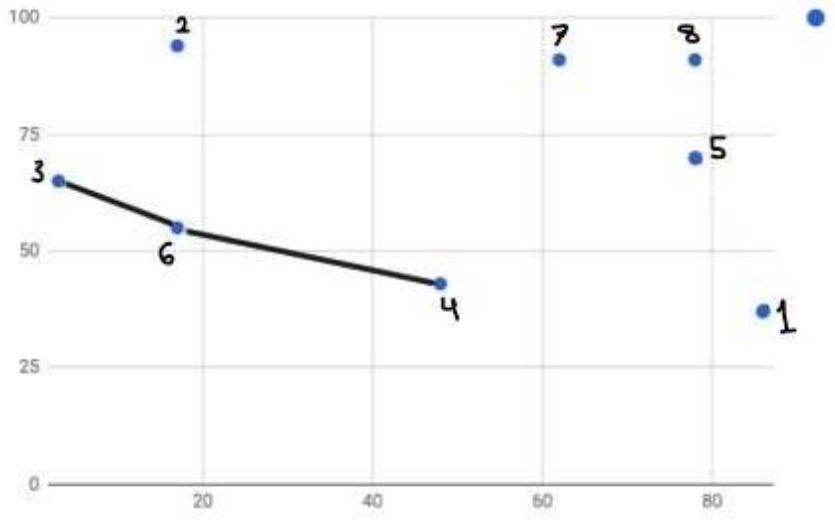
4–6

Total distance: 33



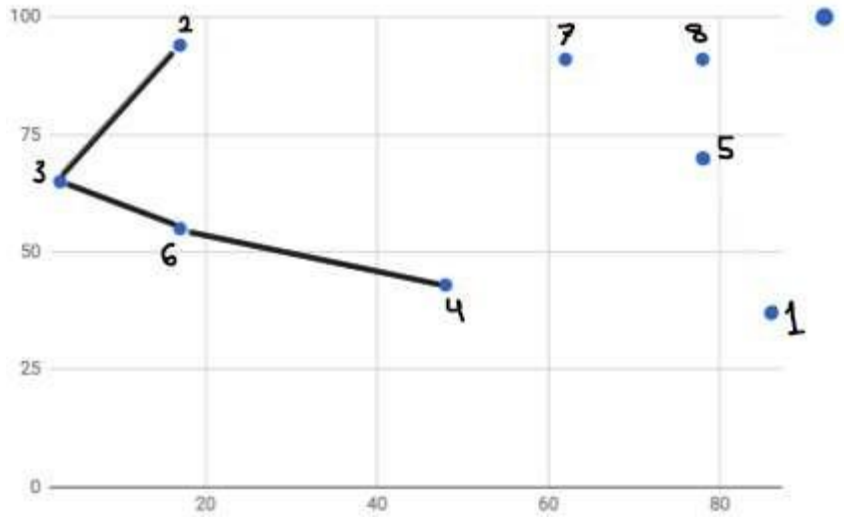
4–6–3

Total distance: 33+17



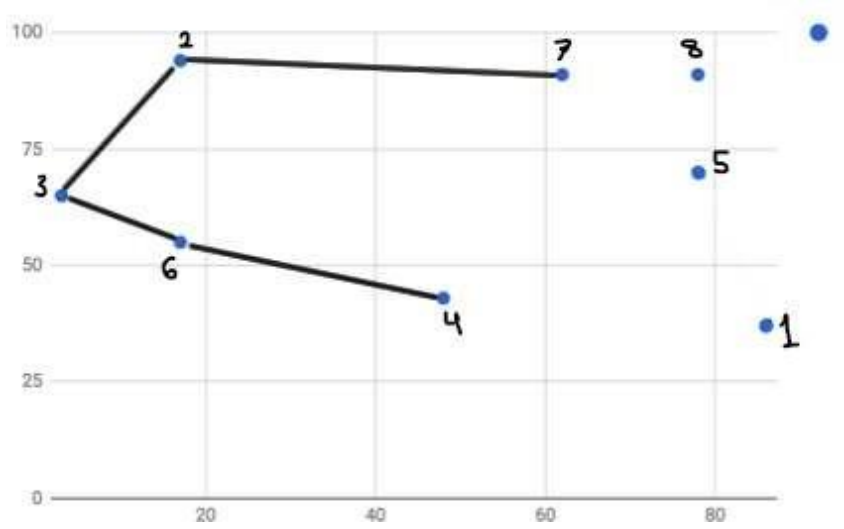
4−6−3−2

Total distance: 33+17+32



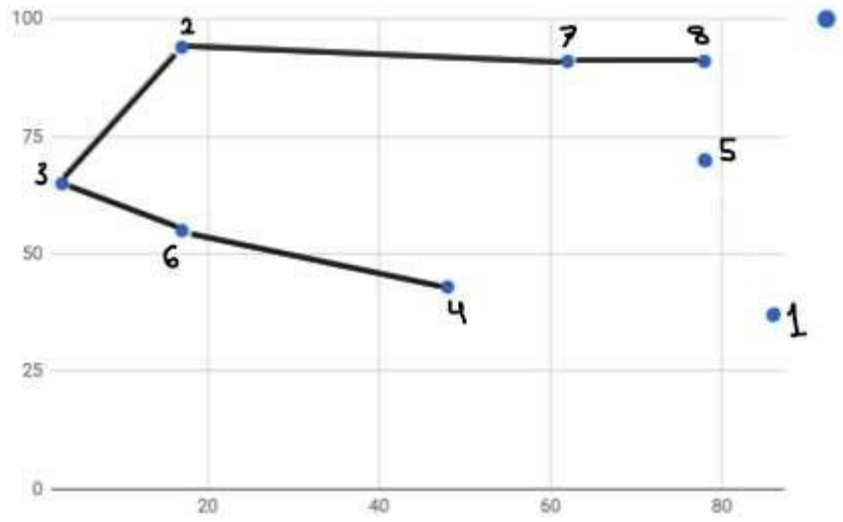
4−6−3−2−7

Total distance: 33+17+32+45



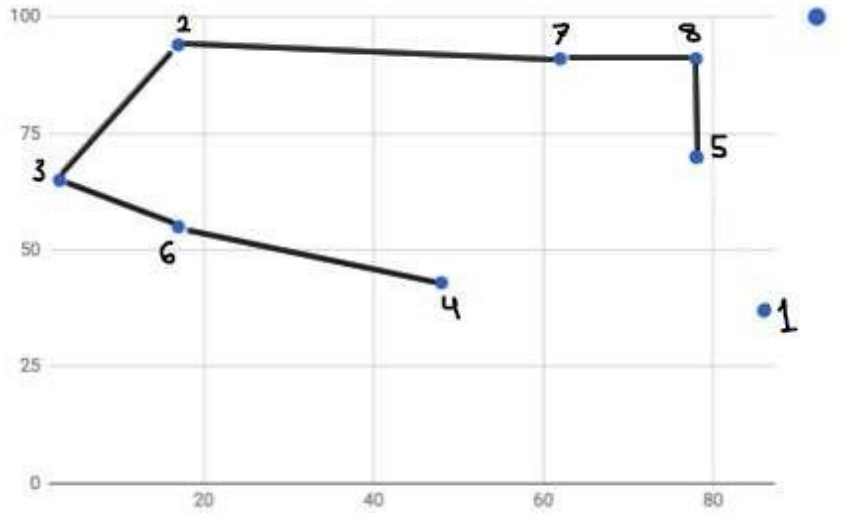
4–6–3–2–7–8

Total distance: 33+17+32+45+16



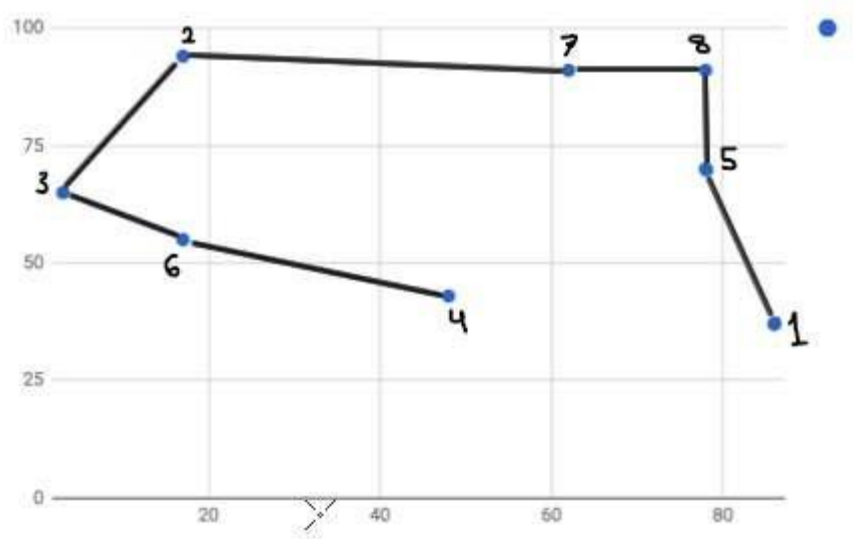
4–6–3–2–7–8–5

Total distance: 33+17+32+45+16+21



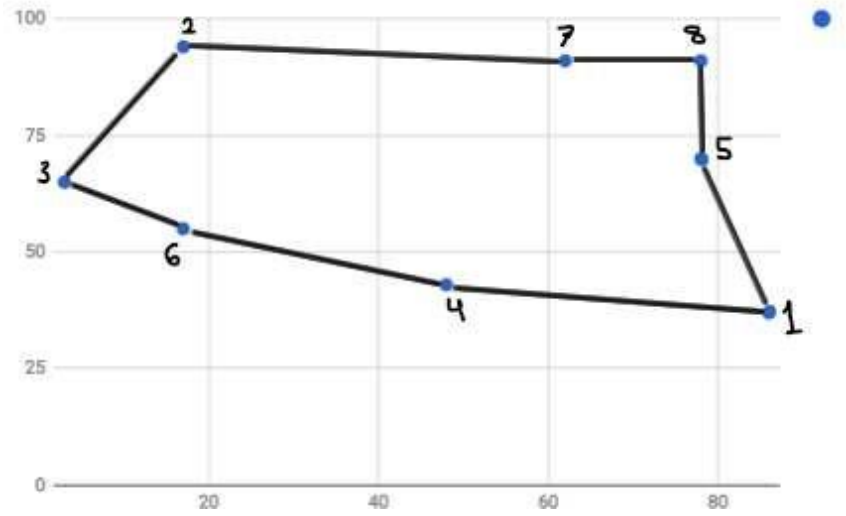
4–6–3–2–7–8–5–1

Total distance: 33+17+32+45+16+21+33



4–6–3–2–7–8–5–1–4

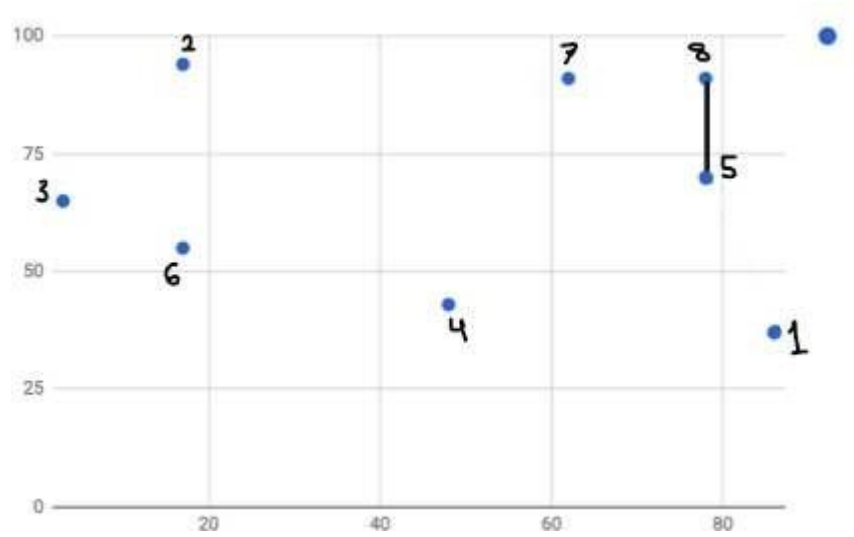
Total distance: 33+17+32+45+16+21+33+38= 235



City 5

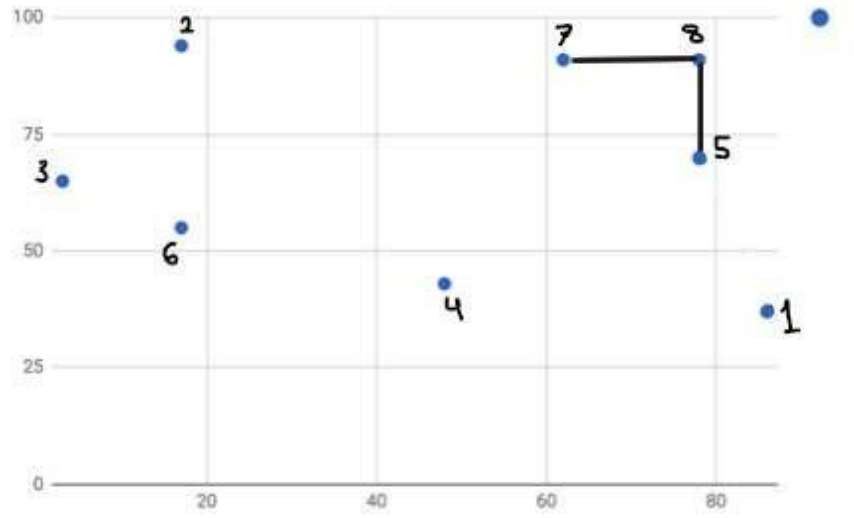
5−8

Total distance: 21



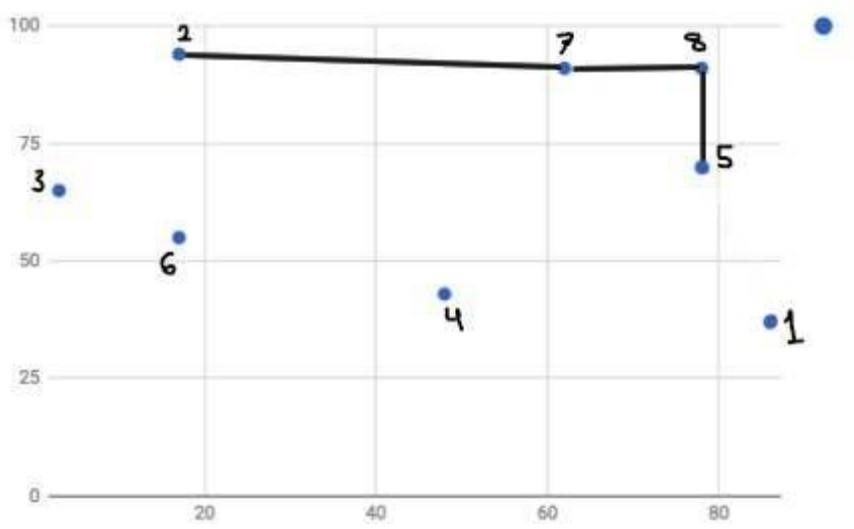
5−8−7

Total distance: 21+16



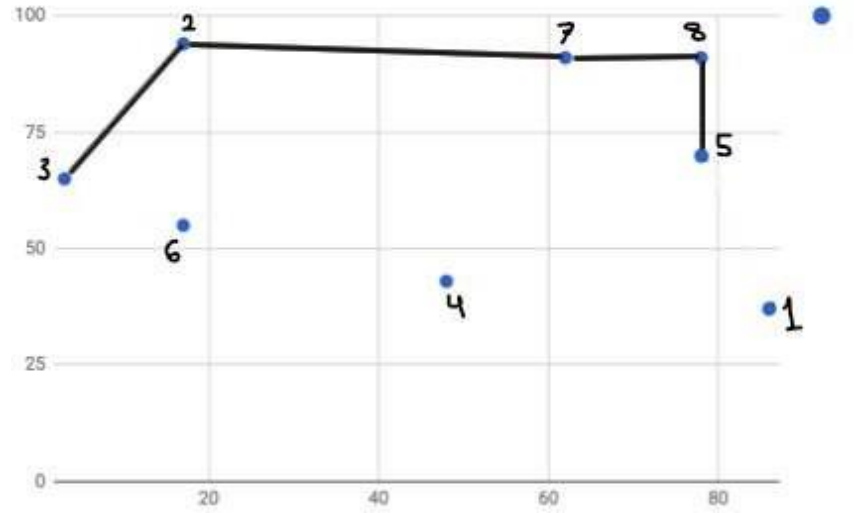
5−8−7−2

Total distance: 21+16+45



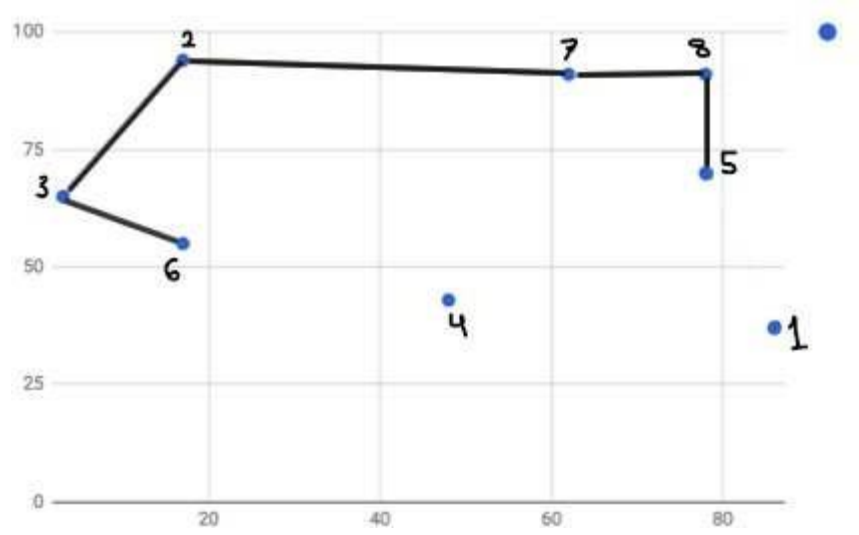
5−8−7−2−3

Total distance: 21+16+45+32



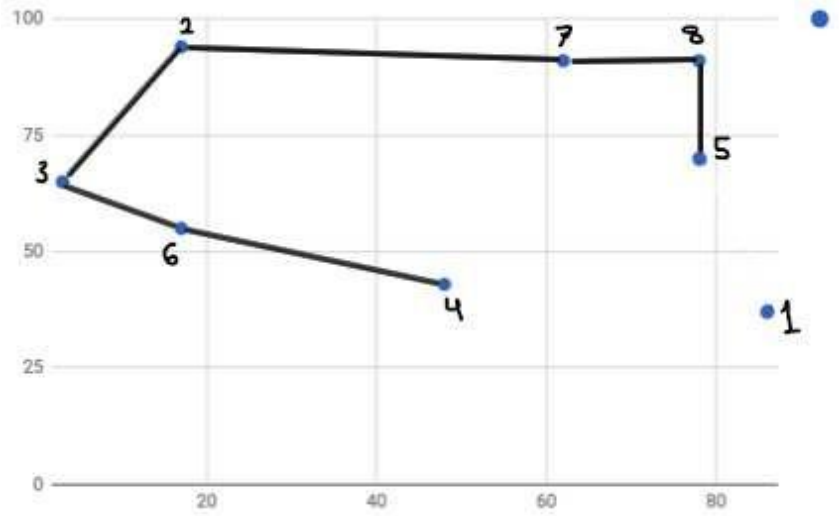
5–8–7–2–3–6

Total distance: 21+16+45+32+17



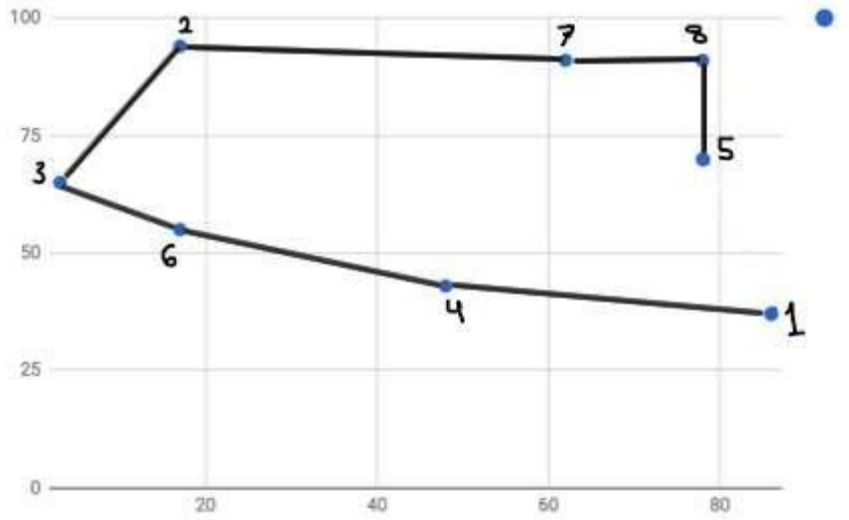
5–8–7–2–3–6–4

Total distance: 21+16+45+32+17+33



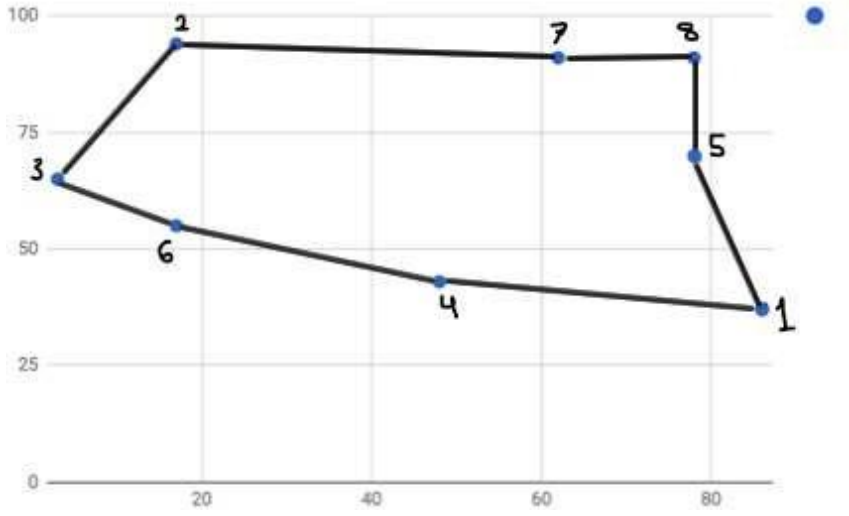
5–8–7–2–3–6–4–1

Total distance: 21+16+45+32+17+33+38



5–8–7–2–3–6–4–1–5

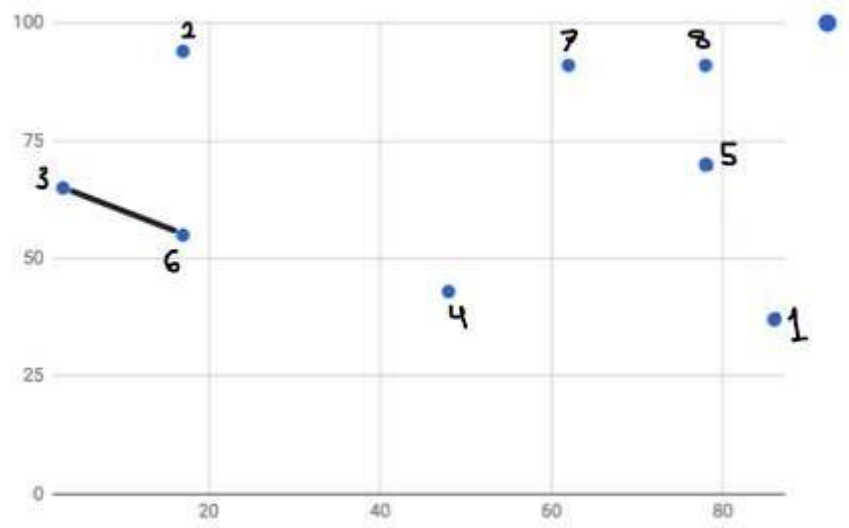
Total distance: 21+16+45+32+17+33+38+33= 235



City 6

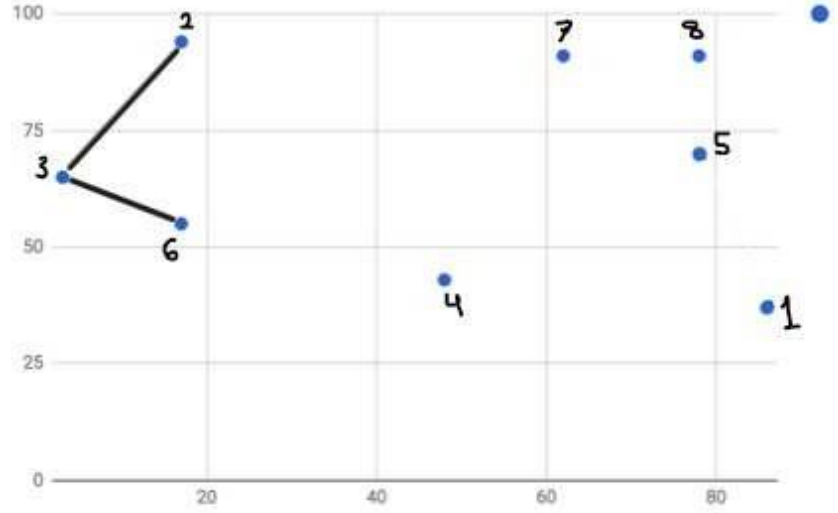
6−3

Total distance: 17



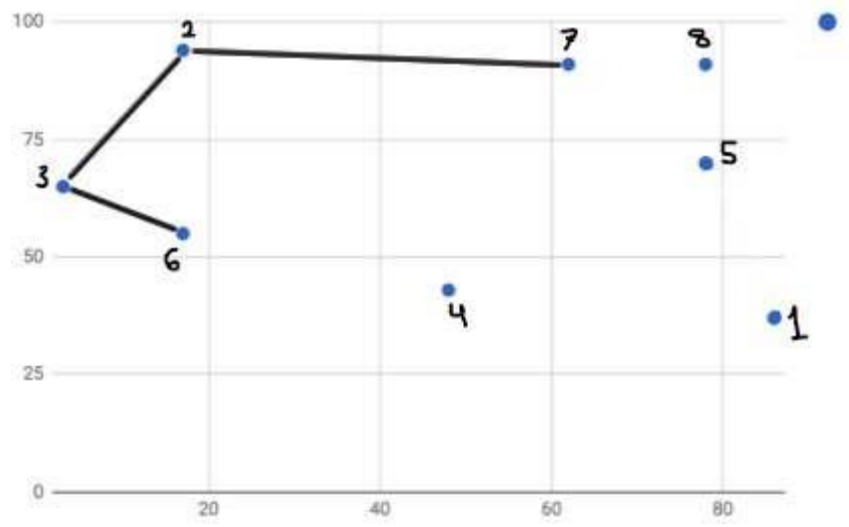
6−3−2

Total distance: 17+32



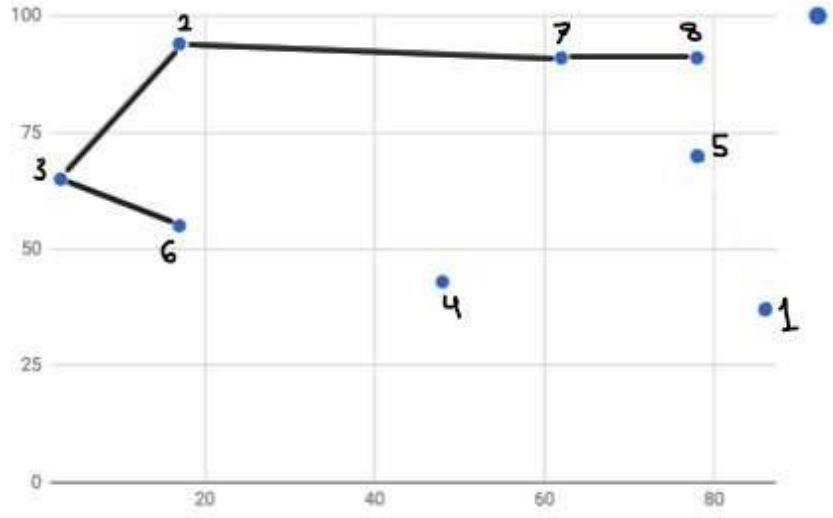
6−3−2−7

Total distance: 17+32+45



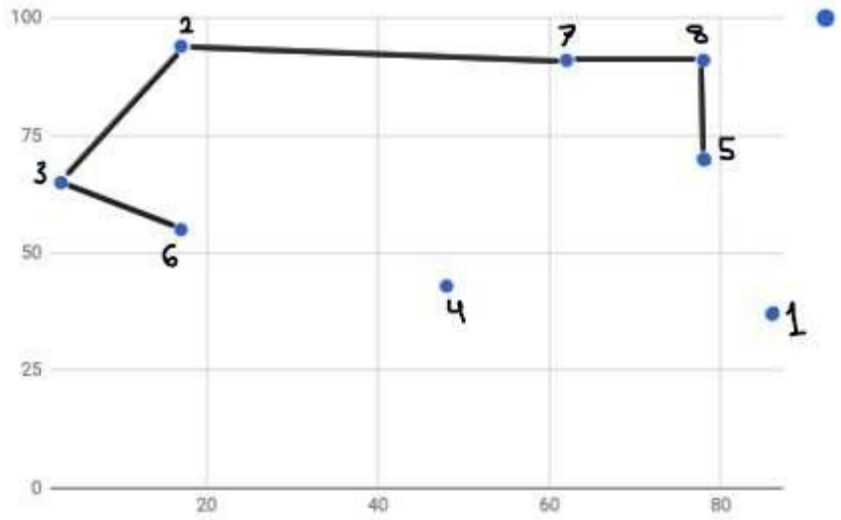
6−3−2−7−8

Total distance: 17+32+45+16



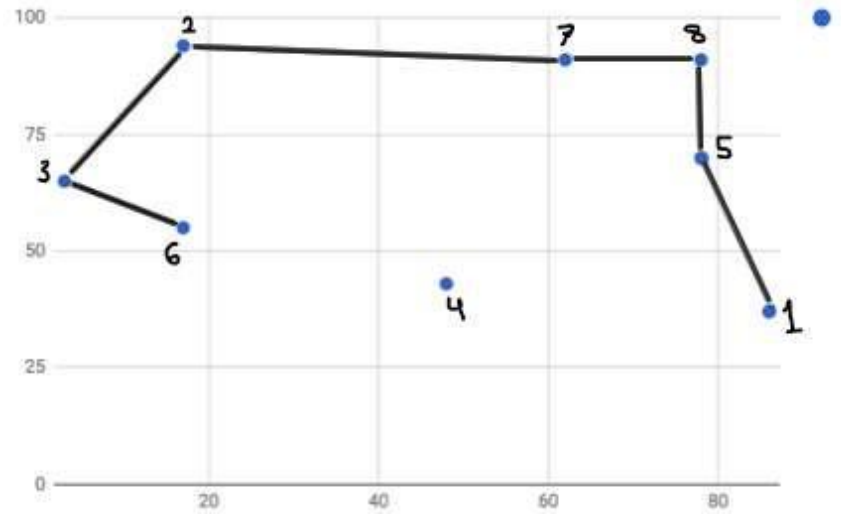
6–3–2–7–8–5

Total distance: 17+32+45+16+21



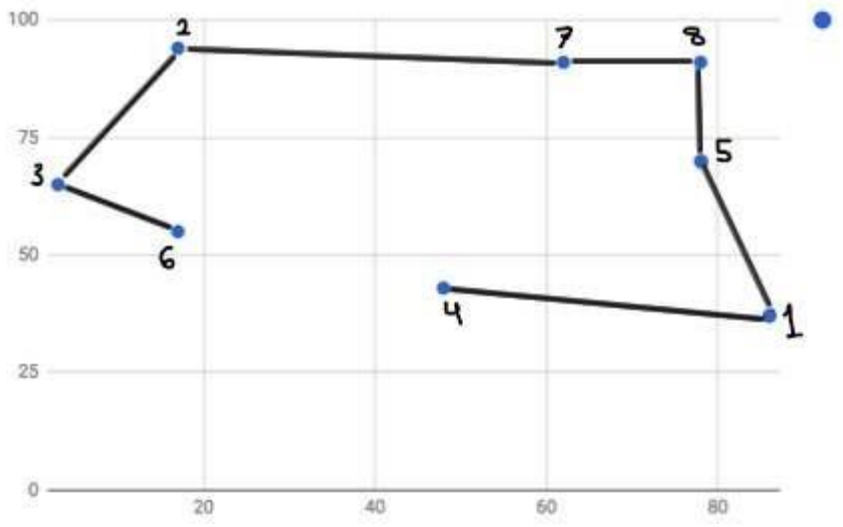
6–3–2–7–8–5–1

Total distance: 17+32+45+16+21+33



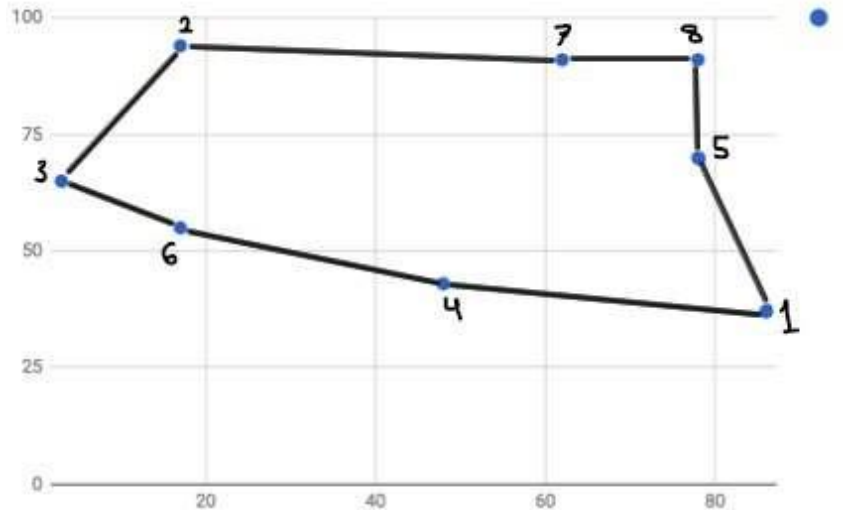
6–3–2–7–8–5–1–4

Total distance: 17+32+45+16+21+33+38



6–3–2–7–8–5–1–4–6

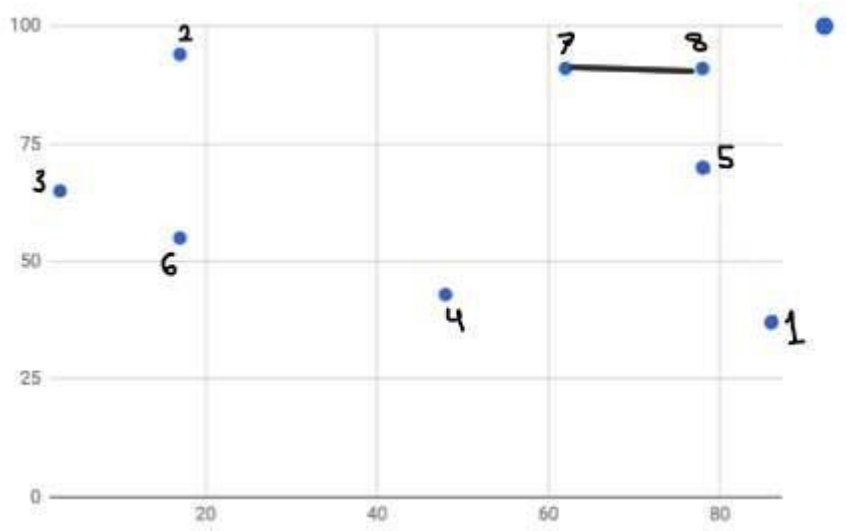
Total distance: 17+32+45+16+21+33+38+33= 235



City 7

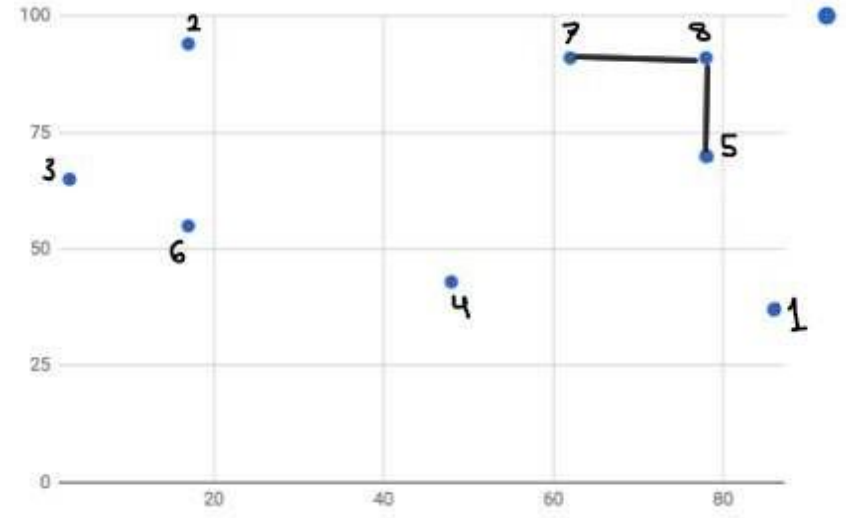
7−8

Total distance: 16



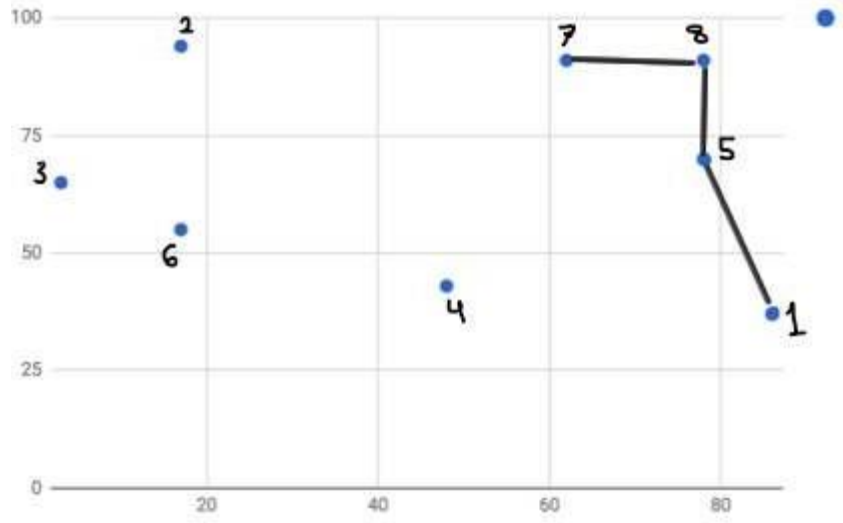
7−8−5

Total distance: 16+21



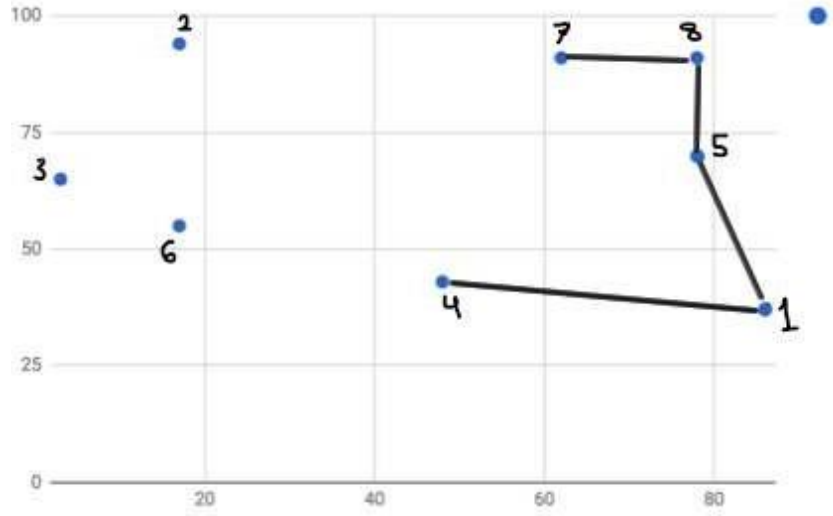
7−8−5−1

Total distance: 16+21+33



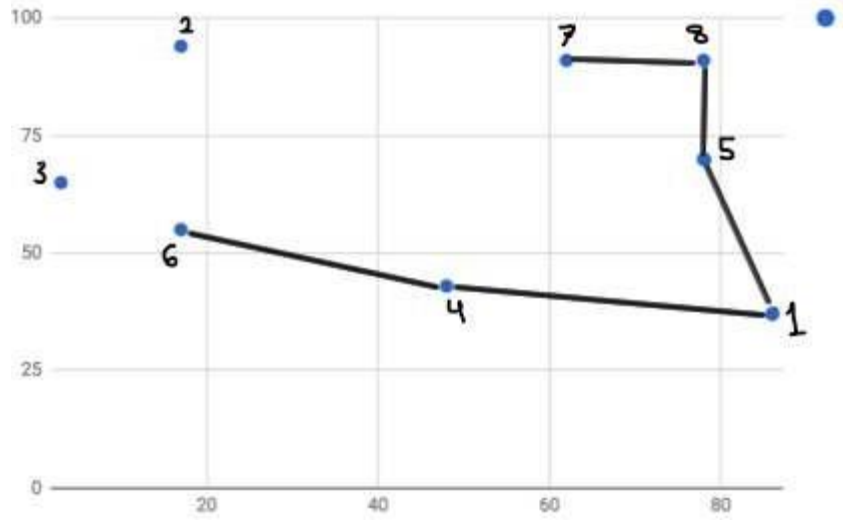
7−8−5−1−4

Total distance: 16+21+33+38



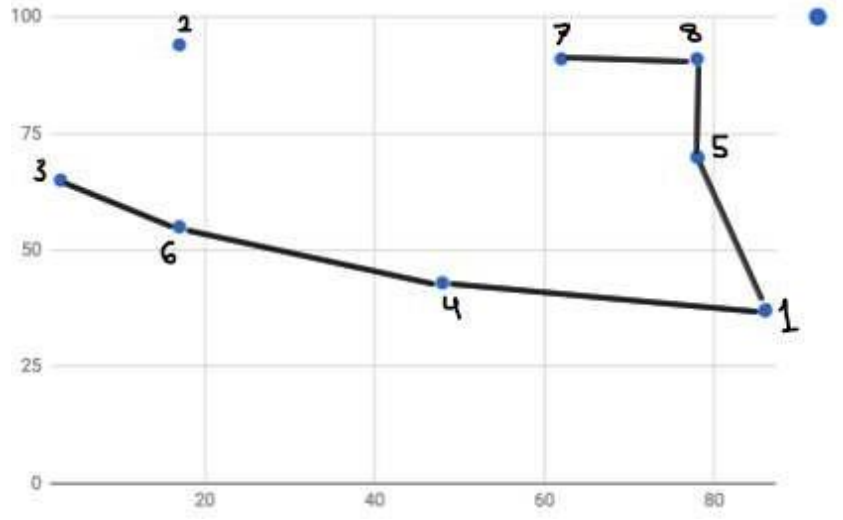
7−8−5−1−4−6

Total distance: 16+21+33+38+33



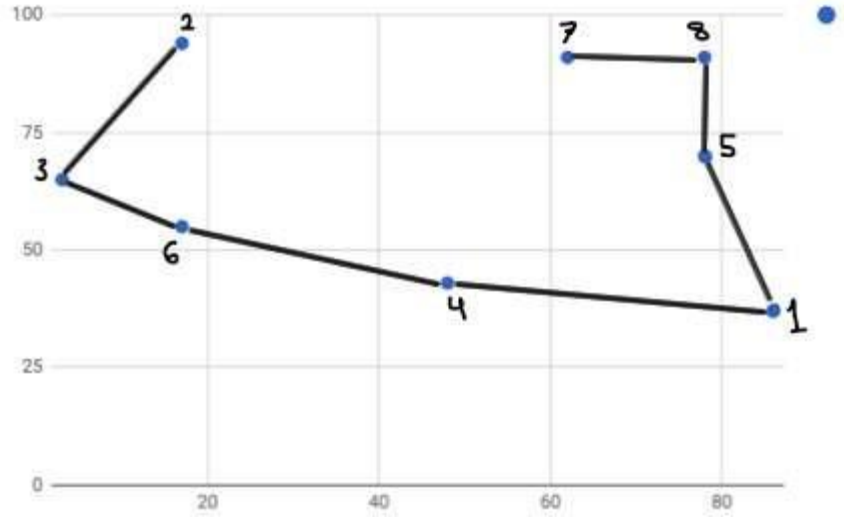
7−8−5−1−4−6−3

Total distance: 16+21+33+38+33+17



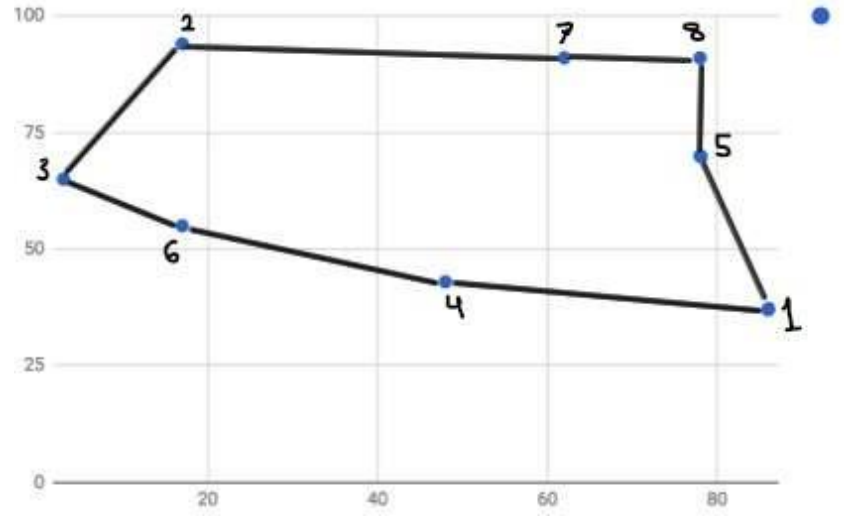
7−8−5−1−4−6−3−2

Total distance: 16+21+33+38+33+17+32



7−8−5−1−4−6−3−2−7

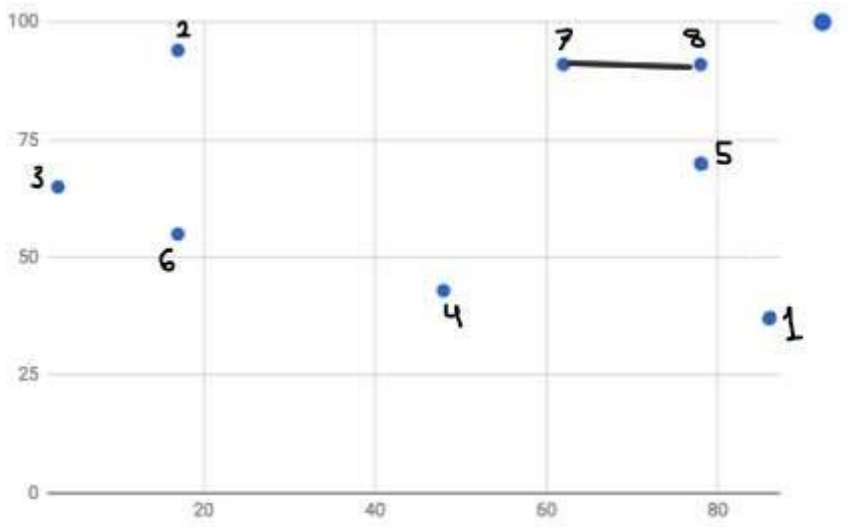
Total distance: 16+21+33+38+33+17+32+45= 235



City 8

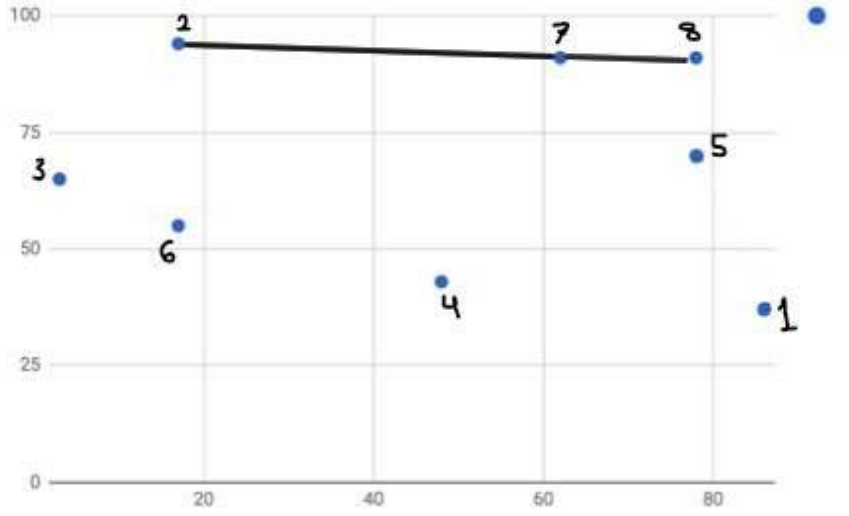
8–7

Total distance: 16



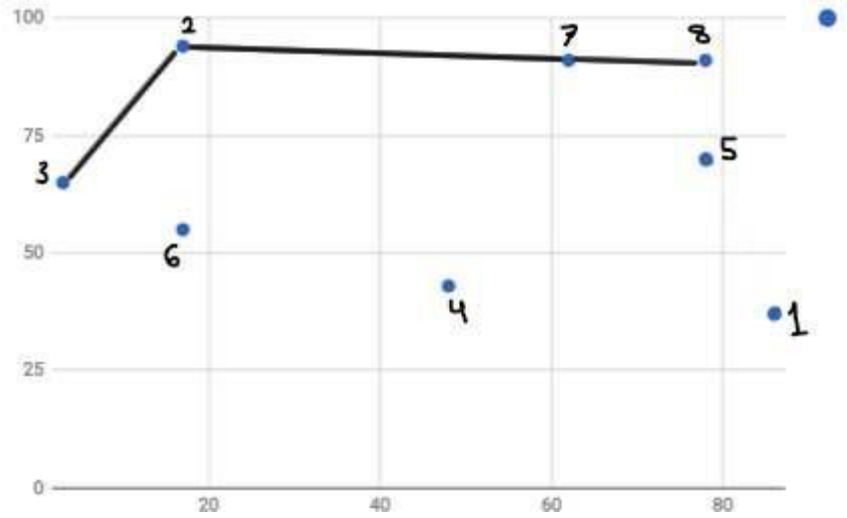
8–7–2

Total distance: 16+45



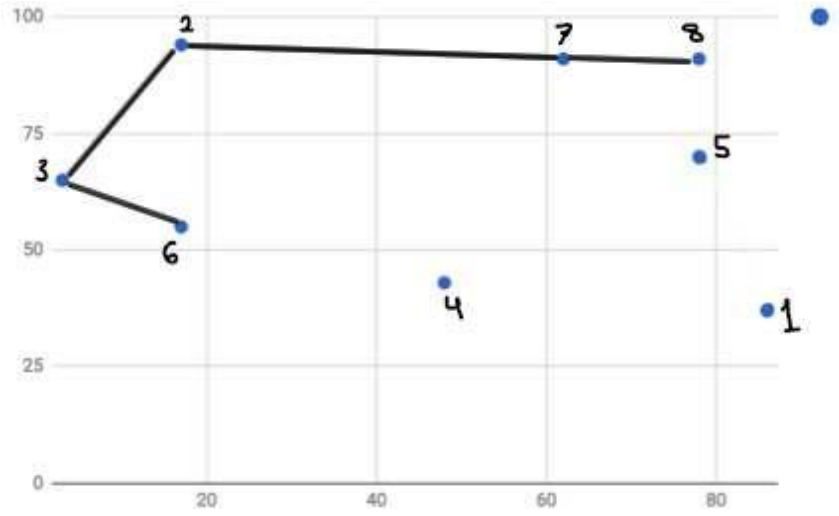
8−7−2−3

Total distance: 16+45+32



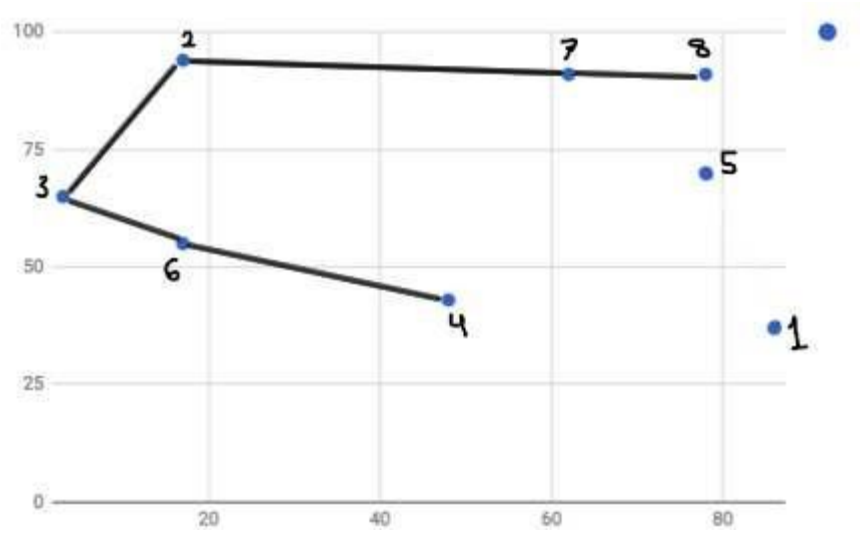
8−7−2−3−6

Total distance: 16+45+32+17



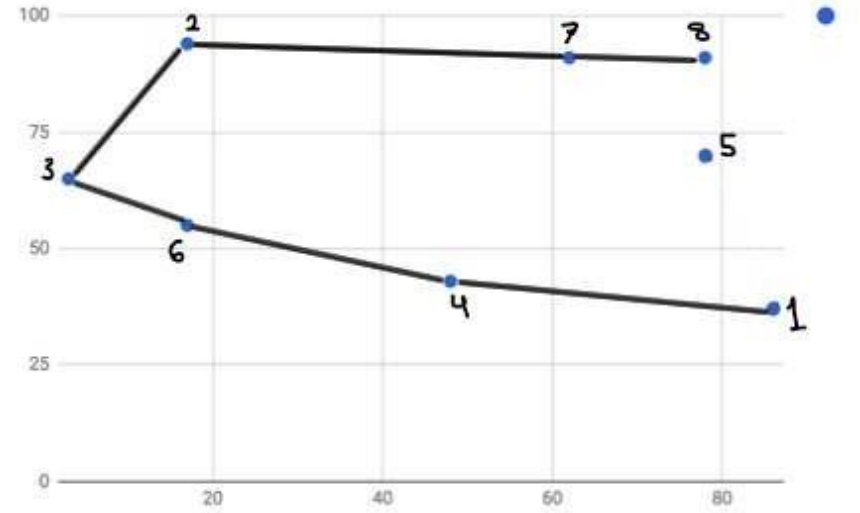
8−7−2−3−6−4

Total distance: 16+45+32+17+33



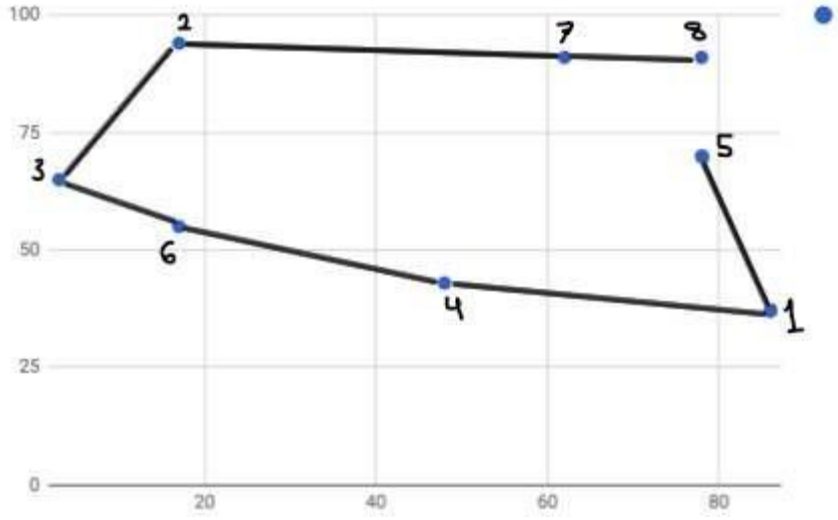
8−7−2−3−6−4−1

Total distance: 16+45+32+17+33+38



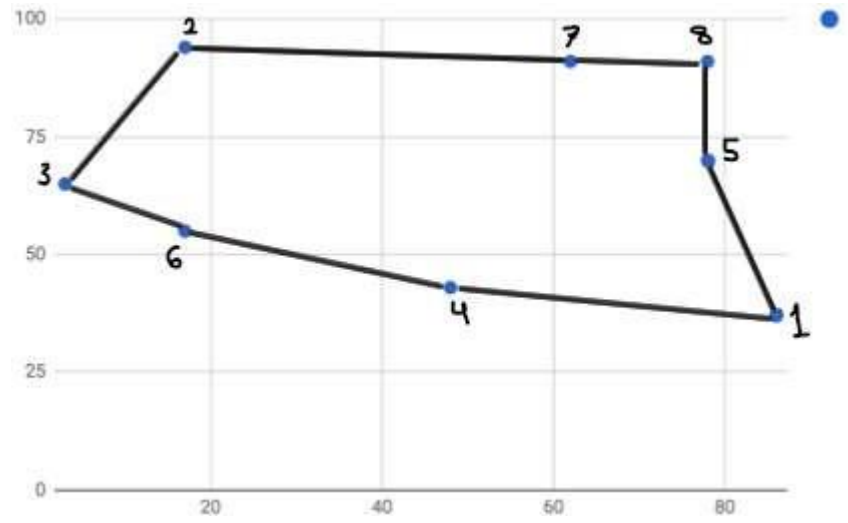
8–7–2–3–6–4–1–5

Total distance: 16+45+32+17+33+38+33



8–7–2–3–6–4–1–5–8

Total distance: 16+45+32+17+33+38+33+21= 235



As seen above, every tour is a feasible solution, you can prove this by seeing at the graphics, every single one form an irregular shape or polygon.

Nearest Insertion Heuristic

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

1. Choose a starting node and add it to the solution.
2. Find the node closest to the starting node and add it to the solution.
3. For each subsequent node not yet added to the solution, find the pair of nodes in the current solution that are closest to it.
4. Select the pair of nodes that minimize the increase in total distance when the new node is inserted between them.
5. Insert the new node between the selected pair of nodes.
6. Repeat steps 3−5 until all nodes are added to the solution.
7. Connect the last node added to the solution with the starting node to complete the cycle.

We'll start by city 1, the algorithm should find the nearest city, which is 5.

	1	2	3	4	5	6	7	8
1	0							
2	89	0						
3	87	32	0					
4	38	59	50	0				
5	33	65	75	40	0			
6	71	39	17	33	62	0		
7	59	45	64	50	26	57	0	
8	54	61	79	56	21	70	16	0

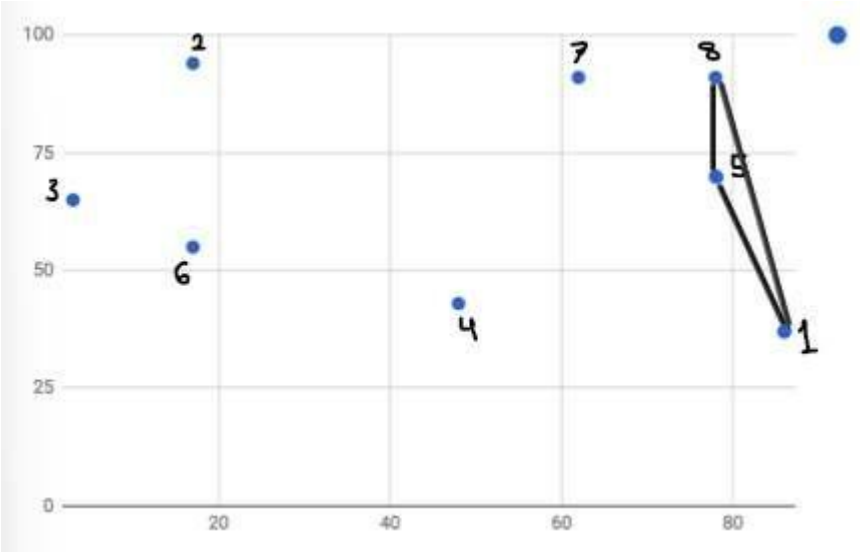
In order to have a better understanding of the heuristic, we will put the first city at the end.

1−5−1

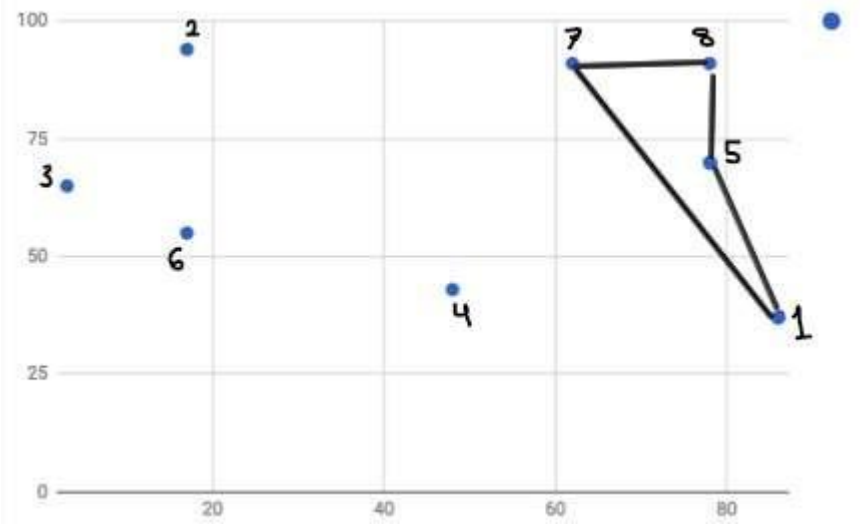
Then, in order to find k the algorithm needs to find the nearest city to j (j being part of the tour and k being the closest unvisited city to the tour), which is city 8. We will also add another variable i, i and j will be used to know the edge where the city has to be inserted.

After doing that, now the algorithm has to select where does the city will be inserted, it does it by using ***Cik + Cjk – Cij,*** by doing so, the algorithm know knows that it has to insert it between 1 and 5.

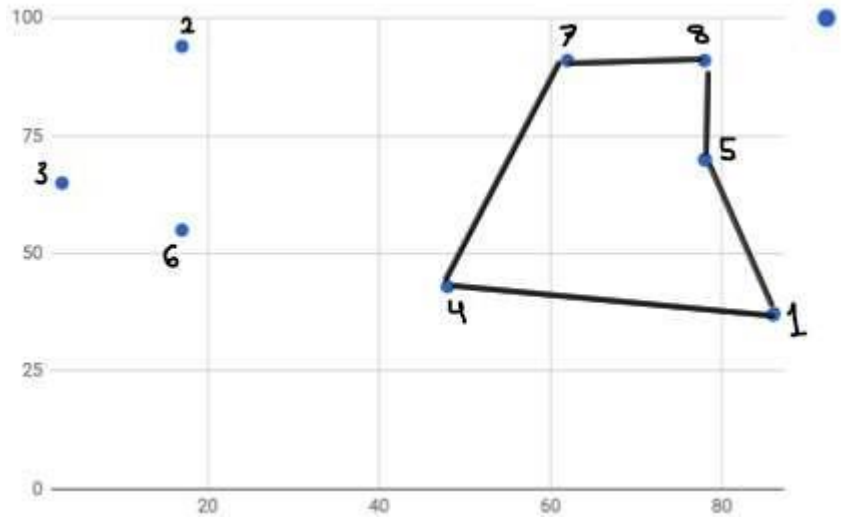
Now it just has to repeat those steps until there is no more unvisited cities.



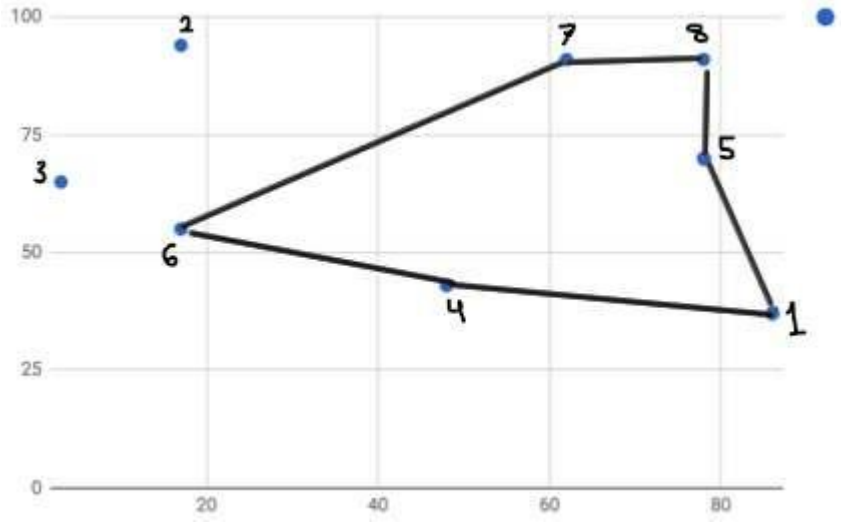
Current tour: 1–8–5–1



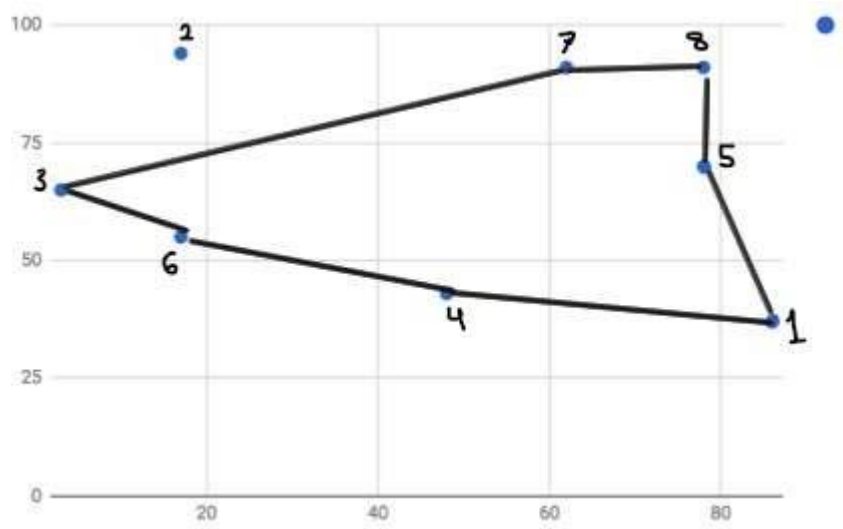
Current tour: 1–7–8–5–1



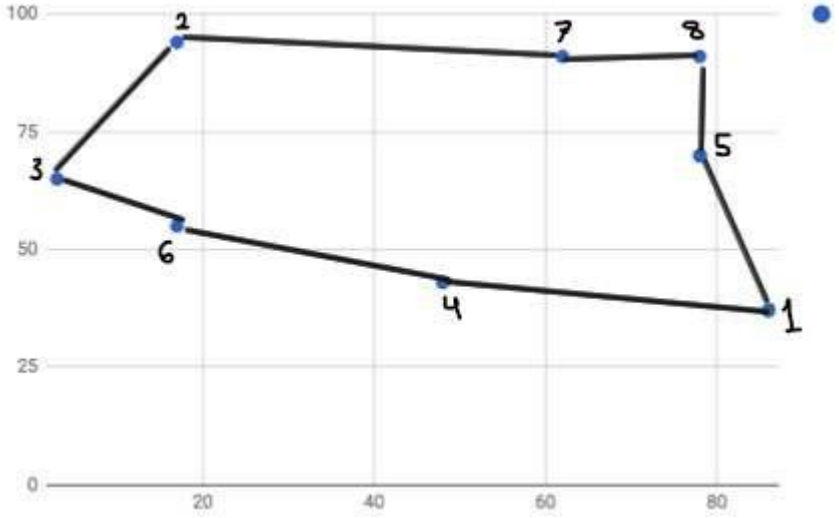
Current tour: 1–4–7–8–5–1



Current tour: 1–4–6–7–8–5–1



Current tour: 1–4–6–3–7–8–5–1



Final tour: 1–4–6–3–2–7–8–5–1

Now that we have the final tour, we can get the total distance:

Total distance: 33+21+16+45+32+17+33+38= 235

Just like the nearest neighbor algorithm, the nearest insertion heuristic would need to do loop his procedure starting from each city in order to get the most optimal solution, it is not necessary to do it right now because we already know that the tour forms an irregular polygon, this means that every tour will have the same distance.