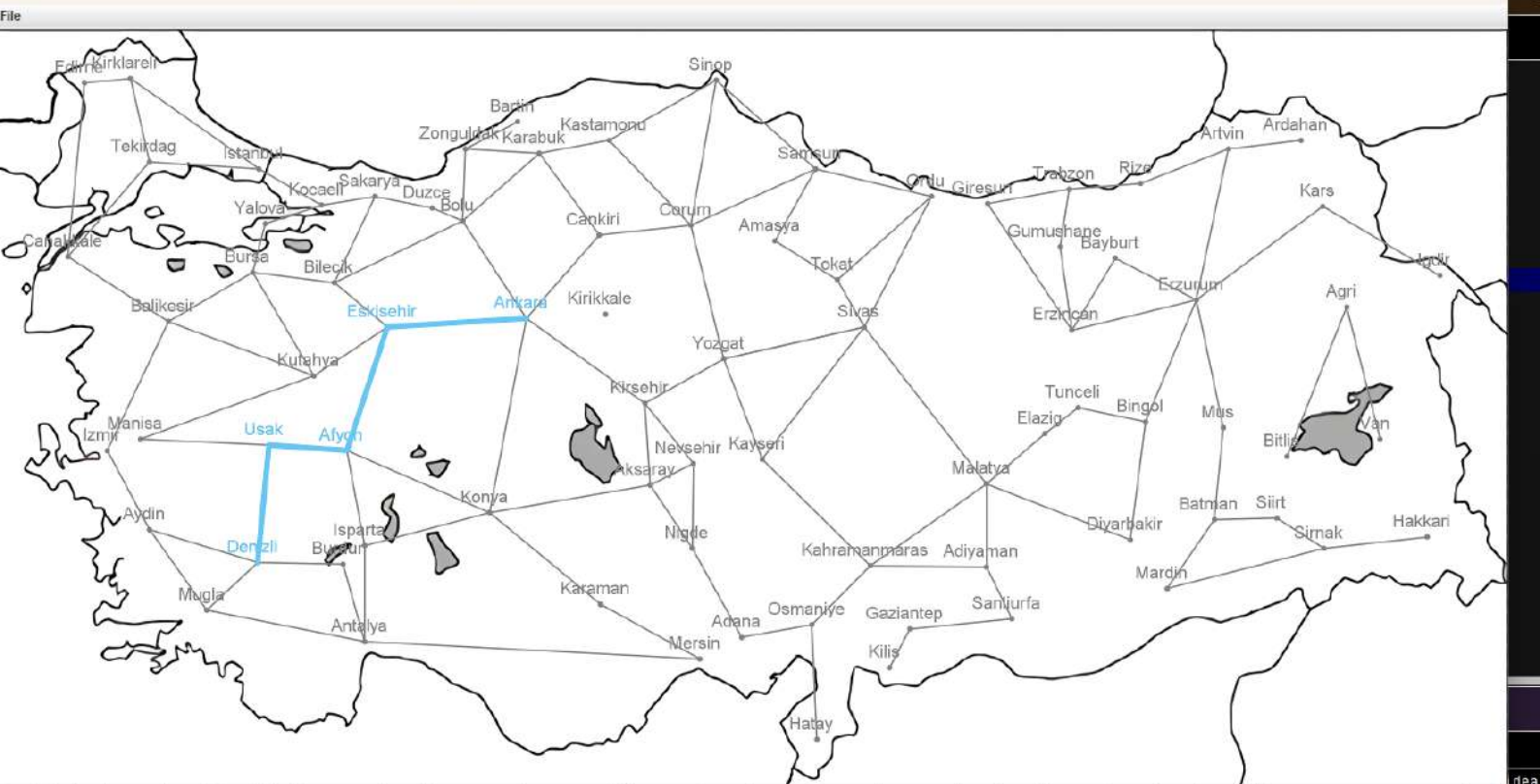
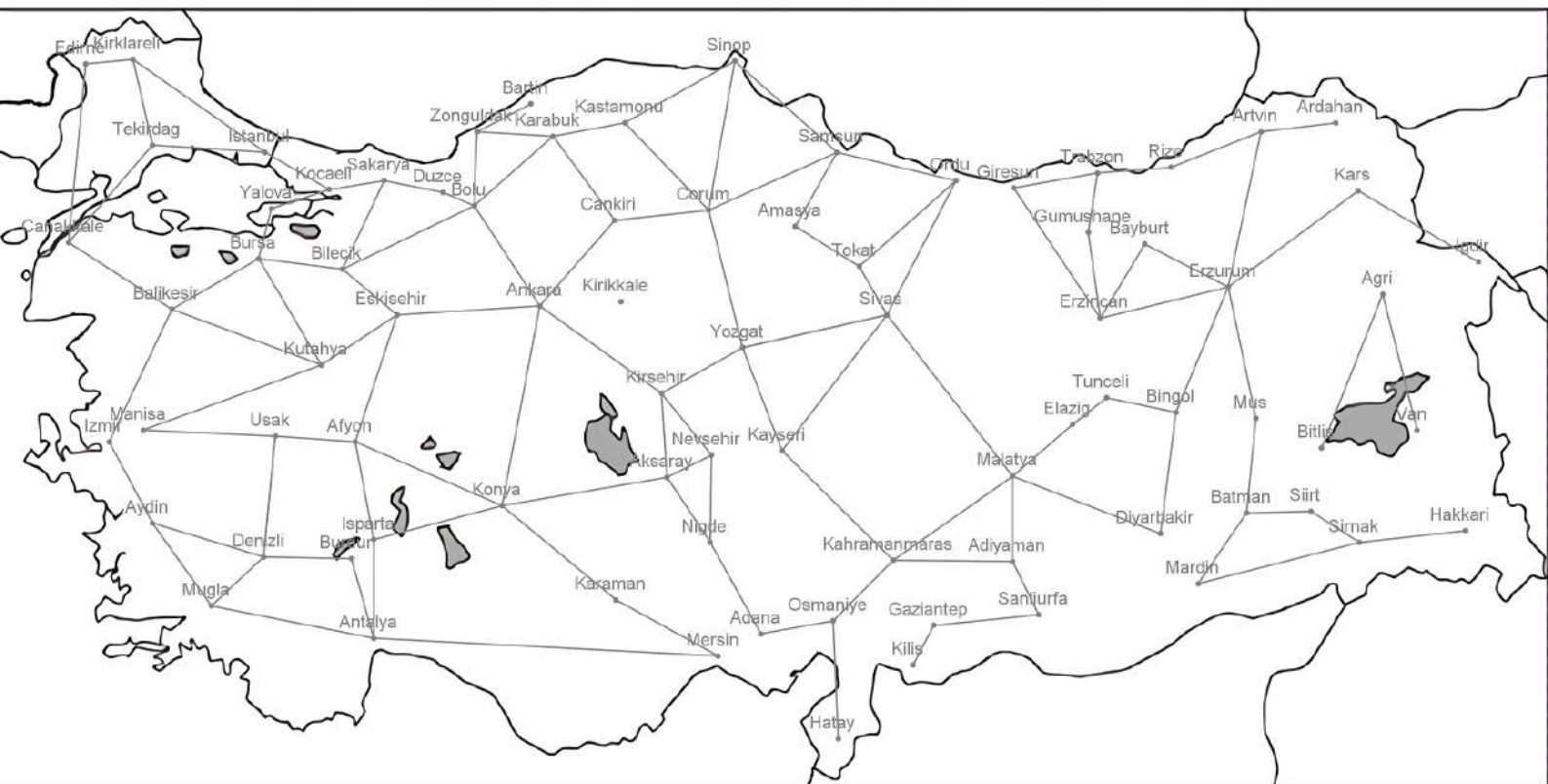


Enter starting city: *Istanbul*  
Enter destination city: *Istanbul*  
Total Distance: 0,00 Path: Istanbul



Enter starting city:Anka  
City named Anka not found Please enter a valid city name.  
Enter starting city:Ankara  
Enter destination city: Deni  
City named Deni not found Please enter a valid city name.  
Enter destination city: Denizli  
Total Distance: 689,18 Path: Ankara->Eskişehir->Afyon->Usak->Denizli



Enter starting city: *Izmir*  
Enter destination city: *Van*  
No path could be found.

In my path finding algorithm I am creating lists named distances coming before and lastplace. I initialized all distances to max value for every city in my distances list and initialized my all city's boolean value to false in my coming before list and finally initialized all cities laststayed city names to null in my lastplace list. Then I changed just my starting city's value in distances list to zero. I determined first index to -1 and mindistance to max value at the beginning. I looked all cities with a for loop and if current city was not visited and also and it's distance value less than mindistance I set minimum index to it's index and minimum distance to it's distance and also I turned it's boolean value in comingbefore list to true to not take into account this city again. Then I looked current city's neighbors with a for loop and I summed up current city's distance value at distance list and it's neighbor's distance value. If this value is smaller than neighbor's distance value in distance list I equalized neighbor's distance value to this sum and I determined neighbors last place as current city. I opened a while loop and initialized current city as destination city. In every loop I added current city to sensible path and changed current city to current city's lastplace city until current city is null. Because of list is starting from destination city I reversed it. Finally I used this sensiblepath list to draw blue lines between cities which represents shortest path.

My Pseudocode:

1.

Determine the distance of all cities in distances list as max value.

Determine all cities boolean value as false in comingbefore list.

Determine the laststayed city for all cities to null.

Determine the distance value of starting city in distances list as zero.

2. If there are cities which weren't visited yet:

Access to the not comingbefore city with the shortest known distance from the starting city.

Determine minimumindex as -1

Determine minDistance as max value.

For each unvisited city:

If the distance to the city is less than minDistance:

Determine minimumindex as the index of this city

Determine minDistance as its distance value at distances list.

If minimumindex did not change and it is still -1, break the loop.

Determine the city at minimumindex in comingbefore list as true.

For each neighbor of the current city:

Calculate the distance from the starting city to the neighbor through the current city.

If this distance is less than the lastly calculated distances value of neighbor in distances list:

Change the distance value of neighbor to current distance.

Change the laststayed city for the neighbor to the current city.

3. Create the shortest path from the starting city to the destination city:

Create an empty list called SensiblePath.

Start from the destination city.

While the current city is not null:

Add the current city to SensiblePath.

Change the current city to it's lastplace city.

Reverse SensiblePath to acquire the true order of cities.

4. Shortest path from the starting city to the destination city is now included in sensiblepath.

MY REFERENCES:

[https://tr.wikipedia.org/wiki/Edsger\\_Dijkstra](https://tr.wikipedia.org/wiki/Edsger_Dijkstra)

<https://www.youtube.com/watch?v=jT3c45XkPTg>