ŞEHİRLER ARASI KISA MESAFE CETVELİ

Yağızhan Fethi Kanbur - 150202055

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[yagizhankanbur@hotmail.com ,](mailto:yagizhankanbur@hotmail.com,) [kanburyagizhan@gmail.com](mailto:kanburyagizhan@gmail.com)

**Problem Tanımı**

Verilen text dosyalarında bulunan şehir ve mesafe bilgilerini girdi olarak kullanacak. İstenildiğinde iki şehir arasında ki en kısa mesafeyi ve geçilen şehirler veya bir başlangıç şehrinden diğer tüm şehirlere olan en kısa mesafeyi dijkstra algoritması kullanarak bulan programı yazmak.

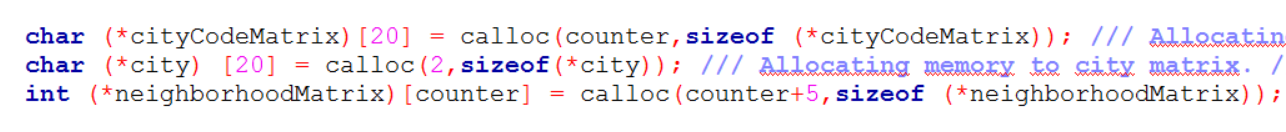
**Yapılan Araştırmalar**

Projeye başlmadan önce dosyalardan verileri çekmek için hangi fonksiyonu kullanmanın daha yararlı olacağına dair bir araştırma yaptım. Kaynak olarak bulduğum internette bulduğum bir siteden yararlandım.[[1]](#endnote-2) Sitede bulunan fonksiyonları inceleyip kullanacağım fonksiyonları belirledim. Daha sonra kullanmaya karar verdiğim fonksiyonlar ve kullanılış şekillerini inceledim.[[2]](#endnote-3) [[3]](#endnote-4)Verilen text dosyalarını inceledikten sonra verildiği şekilde kullanmanın zor olacağını kanaat getirip dosyadan sadece istediğim bilgileri çekmenin yollarını aradım.[[4]](#endnote-5) Dosyadan verileri alırken kullanmak istediğim veriler ile kullanmak istemediğim verileri ayırabilmek için çeşitli kontroller kullandım. Şehir kod dosyasından şehir isimlerini atadığım karakter matrisinde beklenmedik karakterlerin çıkması sonucu yanlış sonuçlar aldım. Bu problemi çözmek için önce matrisi tanımladıktan sonra tüm elemanlarına ‘NULL’ değeri atamayı denedim. Ancak bu durumda hafızada kullandığım başka bir matris ile üst üste binmesi gibi bir sorunla karşılaştım. Stackoverflow’u sorunumu çözmek için kullanmaya çalıştım.[[5]](#endnote-6) Sorunu çözmek için matrisi tanımlarken ‘memory allocation’ fonksyionlarından birini kullanmaya karar verdim ve fonksiyonları inceledim.[[6]](#endnote-7) [[7]](#endnote-8) [[8]](#endnote-9) ‘calloc’ fonksiyonunu kullandım. Projede dijkstra algoritması için gerekli olan komşuluk matrisini oluşturabilmek için iki ayrı dosyadaki verileri karşılaştırabilmem gerekiyordu. String kütüphanesi fonksiyonları arasında ihtiyacıma uygun bir fonksiyon bulabileceğimi düşünerek kütüphaneyi inceledim. [[9]](#endnote-10) [[10]](#endnote-11) Araştırmalarım sonucu strcmp() fonksiyonunu kullanabileceğime karar verdim. Projenin dosya işleme ile ilgili olan kısmını bitirdikten sonra her şeyin düzgün çalışıp çalışmadığını görmek için kontrol çıktıları yazdırdım. Herhangi bir hatanın olmadığını gördükten sonra dijkstra algoritmasını araştırmaya başladım. [[11]](#endnote-12) [[12]](#endnote-13) Dijkstra algoritmasını koda dökebilmek için sözde kod üzerinden çalışmanın daha kolay olacağını düşündüm sözde kod bulmaya çalıştım.[[13]](#endnote-14) [[14]](#endnote-15) Komşu olmayan şehirler arası mesafeye sonsuz değer atamam gerektiğini gördüm bu yüzden limits.h kütüphanesini ve ‘INT\_MAX’ tanımını kullandım. Sözde kodları kendi kodumla uyumlu hale getirip asıl kodlamaya başladım.[[15]](#endnote-16) Dijkstrayı uygularken en büyük zorluk programın iki çıktı vermesiyle ilgiliydi. Bunu iki farklı dijkstra fonksiyonu yazarak çözdüm. INT\_MAX tanımı integer bir değişkenin alabileceği en büyük değeri tutuğu için dijkstrada kullanırken herhangi bir sayı eklenince negatif sonsuz değerine geçiyor ve algoritmanın yanlış sonuç vermesine yol açıyordu. Bu sorunuda kod da geriye dönerek komşuluk matrisini oluşturduğum yerde değişiklik yaparak çözdüm.

Projeyi geliştirirken dil olarak c programlama dilini kullandık.C Programlama Dili genel amaçlı orta seviyeli ve yapısal bir programlama dilidir. 1972 yılında Dennis Ritchie tarafından Bell Telefon Labaraturvarında Unix işletim sistemi ile kullanılmak için tasarlanmıştır. C, özellikle sistem programlamada sembolik makine dili (Asembler) ile tercih edilmektedir. İşletim sistemleri, derleyiciler ve debug gibi aşağı seviyeli sistem programlarının yazılımında yoğun olarak C programlama dili kullanılır.

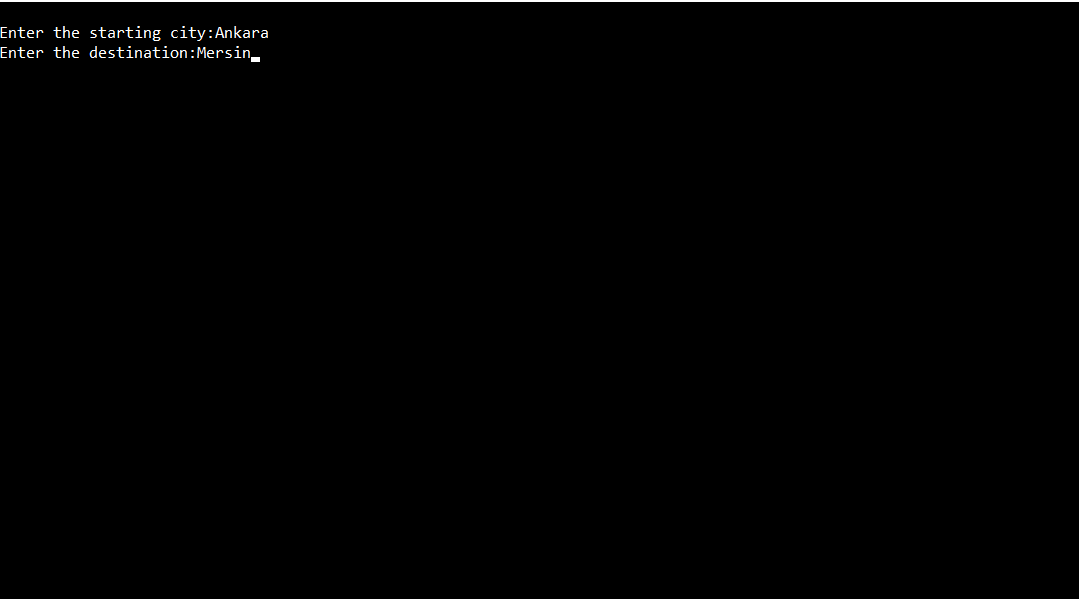
Proje içerisinde kütüphane olarak stdio, string, limits, stdbool kütüphaneleri kullandım. Bu kütüphanelere ek olarak kendi yazdığım fonksiyonları tutmak için 'fdijkstra' ve 'dijkstra' kütüphanelerini oluşturdum. Stdio kütüphanesi, C/C++ programlama dillerinde kullanılan dosya giriş/çıkış işlemlerini gerçekleştirmeye yarayan en temel C kütüphanelerden biridir. İsmi, C Standart Input and Output Library (cstdio.h, bilinen adıyla stdio.h)'den gelmektedir. Stdbool kütüphanesi bool değişken tipini stabil kullanabilmek için kullanılır. “limits.h” header’ı “char”, “short”, “int”, “long” gibi değişken tiplerinin maximum ve minumum değer aralıklarını belirleyen standart C kütüphanesidir. [[16]](#endnote-17) String.h başlık dosyasında bulunan fonksiyonların hemen hemen hepsi bir yazının başlangıç adresini alarak yazı ile ilgili işlem yapar yada yazı hakkında bir takım bilgileri geri döndürür.

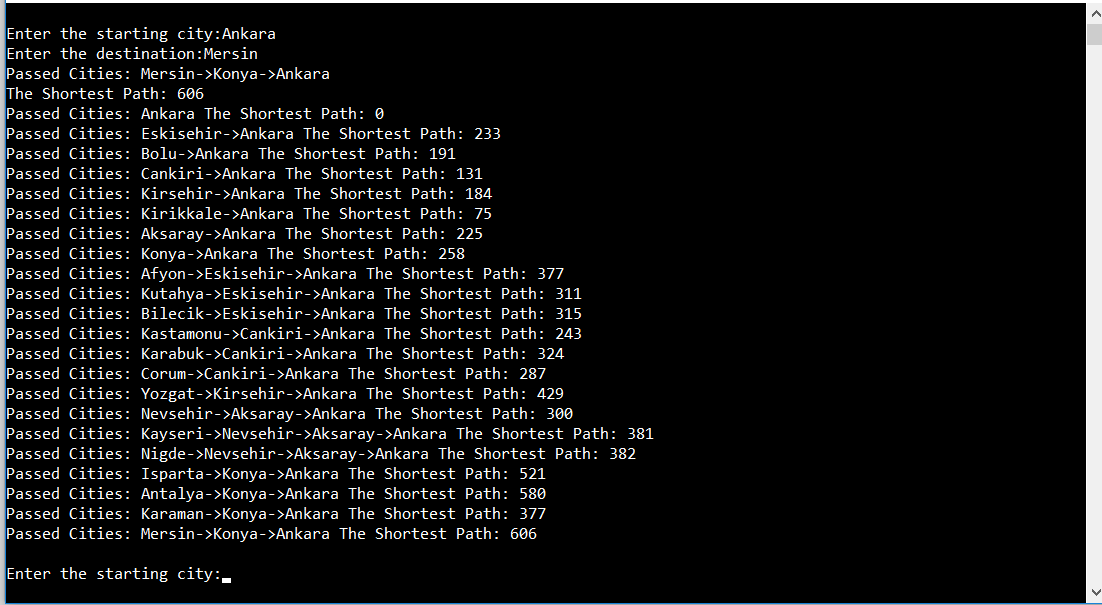
**Veri Yapıları**

*Şekil 1: Üç adet matristen oluşan veri yapısı.*

Veri yapısı olarak üç matris kullanıldı. CityCodeMatrix şehir kod dosyasında bulunan şehir isimlerini tutmaktadır. City matrisi şehir mesafe dosyasında bulunan her satırdaki iki şehri tutmaktadır dosyada satır atlanıldığında matris sıfırlanıp yeniden kullanıma hazır hale gelir. NeighborhoodMatrix şehirler arasındaki mesafeyi tutar eğer şehirler arasında komşuluk yoksa mesafe sonsuzdur.

**Kullanıcı Kataloğu**





*Şekil 2: Girdi ekranı.*

*Şekil 3: Sonuç ekranı.*

**Yazılım Mimarisi**



**Tasarım**

**Akış Şeması**



**Veri Tabanı Diyagramı**



**Genel Yapı**

Program kullanıcıdan girdiler alarak çalışmak zorundadır. Girdilerin ilki kullanıcının programı hangi işlev için kullanacağı girdisidir. İşlevlerden kasıt programdan istenen iki şehir arasındaki en kısa mesafe çıktısı ve bir şehirden diğer tüm şehirlere olan en kısa mesafeler çıktısıdır. İkinci girdi için kullanıcıdan seçtikleri işleve göre bir yada iki şehir ismi istenmektedir. Program kullanıcının yanlış girdiler girme ihtimaline karşı kontroller yapmakta ve eğer yanlış girdi girildiyse hata vermekte yeniden girdi istemektedir. Kullanıcı girdileri dışından iki adet text dosyasıda veri girdisi olarak kullanılmaktadır. Dosyalardan birinden kullanılacak şehirlerin ismi alınmakta ve bir matrise atanmaktadır. Diğer dosyadan ise şehirler arasındaki mesafe alınıp komşuluk matrisine atanmaktadır. Kullanıcıdan alınan girdiler ile dosyadaki girdiler karşılıklı kontrol edilip hata olup olmadığına bakılır. Alınan girdilerden gerekli görülenleri dijkstra algoritmasının fonksyionuna girdi olarak gönderilmektedir. Dijkstra algoritması en kısa yolu bulma işlemini gerçekleştirir. Sonra gerekli fonksiyonlar çağırılıp çıktı ekrana yazdırılır. Bundan sonra program kapatılana kadar kullanıcıdan yeniden girdi istenip yeni sonuçlar bulunabilir. Kullanılan fonksiyonlar ve açıklamaları aşağıdaki şekildedir.

int compareFirstCity(int , char ,char) : Verilen şehir ile şehir isimlerini tutan matrisi karşılaştırır verilen şehirin matristeki indisini döndürür.

int compareSecondCity(int , char ,char) : Verilen şehir ile şehir isimlerini tutan matrisi karşılaştırır verilen şehirin matristeki indisini döndürür.

void addDistance (int ,int , int , int nm, int ): Verilen indislere göre komşuluk matrisine verilen yolu atar.

int cityCounter (int ): Verilen text dosyasına göre dosyadaki şehir sayısını bulur ve döndürür.

void addToCCM (char ): Verilen text dosyasına göre şehir matrisine şehir isimlerini atar.

void addToNM (int , char , char , int ): Verilen text dosyasında bulunan şehirleri geçici bir matrise alır girdilerini kullanarak addDistance fonksiyonunu çağırır.

int shortestPathBetweenTwoCities(int , int ,int ,int , char ): Dijkstra algoritmasını kullanarak verilen iki şehir arasındaki en kısa mesafeyi bulur mesafeyi döndürür.

void shortestPathToAllCites(int , int ,int ,char ) : Dijkstra algoritmasını kullanarak verilen şehirden geri kalan tüm şehirlere olan en kısa mesafeyi bulur ve yazdırır.

void passedCities(int , int , char , int , int , int ): Geçilen şehirleri yazdırır.

1. <http://stackoverflow.com/questions/16715633/multidimensional-arrays-allocated-through-calloc> (Access date: 10.03.2017) [↑](#endnote-ref-2)
2. <https://www.tutorialspoint.com/c_standard_library/c_function_rewind.htm> (Access date: 10.03.2017) [↑](#endnote-ref-3)
3. <https://www.tutorialspoint.com/c_standard_library/c_function_fgetc.htm> (Access date: 10.03.2017) [↑](#endnote-ref-4)
4. <http://www.sanfoundry.com/c-programming-examples-file-handling/> (Access date: 10.03.2017) [↑](#endnote-ref-5)
5. [http://stackoverflow.com/questions/42721183/unintendedunexpected-change-in-matrix?noredirect=1#comment72564005\_42721183](http://stackoverflow.com/questions/42721183/unintendedunexpected-change-in-matrix?noredirect=1" \l "comment72564005_42721183) (Access date: 10.03.2017) [↑](#endnote-ref-6)
6. <https://en.wikipedia.org/wiki/C_dynamic_memory_allocation> (Access date: 10.03.2017) [↑](#endnote-ref-7)
7. <https://www.programiz.com/c-programming/c-dynamic-memory-allocation> (Access date: 10.03.2017) [↑](#endnote-ref-8)
8. <http://stackoverflow.com/questions/16715633/multidimensional-arrays-allocated-through-calloc> (Access date: 10.03.2017) [↑](#endnote-ref-9)
9. <http://pubs.opengroup.org/onlinepubs/7908799/xsh/string.h.html> (Access date: 10.03.2017) [↑](#endnote-ref-10)
10. <http://pubs.opengroup.org/onlinepubs/7908799/xsh/strcmp.html> (Access date: 10.03.2017) [↑](#endnote-ref-11)
11. [https://en.wikipedia.org/wiki/Dijkstra%27s\_algorithm](https://en.wikipedia.org/wiki/Dijkstra's_algorithm) (Access date: 10.03.2017) [↑](#endnote-ref-12)
12. <http://bilgisayarkavramlari.sadievrenseker.com/2010/05/13/dijkstra-algoritmasi-2/> (Access date: 10.03.2017) [↑](#endnote-ref-13)
13. <https://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html> (Access date: 10.03.2017) [↑](#endnote-ref-14)
14. <http://www.gitta.info/Accessibiliti/en/html/Dijkstra_learningObject1.html> (Access date: 10.03.2017) [↑](#endnote-ref-15)
15. <http://math.mit.edu/~rothvoss/18.304.3PM/Presentations/1-Melissa.pdf> (Access date: 14.03.2017) [↑](#endnote-ref-16)
16. <https://muhammedcobanoglu.wordpress.com/2014/05/02/c-limits-h-kutuphanesi/> (Access date: 15.03.2017) [↑](#endnote-ref-17)