**GEZGİN ZEPLİN**

*Erhan ÖZDOĞAN*

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

ozerhan9@gmail.com

*Özet*

Gezgin zeplin projesi bir ana fonksiyon ve bu fonksiyonun yanında on adet yan fonksiyondan oluşmaktadır

Proje bir zeplin aracının bir şehirden diğer bir şehre şehirlerin komşularının üstünden geçecek şekilde gidilip gidilmediğine bakarak bir graf yapısında bu bilgileri tutup bu graf üzerinde belirli algoritmalar yürüterek maliyet ve güzergah hesaplamaları yapmaktadır.

Program kendine verilen txt dosyasından şehirlerin latitude longtitude ve rakım bilgilerinin yanından komşu şehirlerinin bilgilerini de alır ve bunları bir struct yapısı içinde tutar. Daha sonra komşuluklara bakarak bir graf yapısı oluşturur bu graf yapısın şehirler düğüm şehirlerin arasındaki maliyetler kenarlar olarak alınmıştır maliyetler hesaplanırken eğim göz önünde bulundurularak aracın o şehre gidip gidemeyeceği bulunmuştur. Ve graf yapısı dosyaya yazdırılmıştır. Kullanıcıdan başlangıç ve bitiş şehri istenmiştir, kullanıcının girdiği değerlere göre graf üzerinde dijkstra algoritması çalıştırılır ve başlangıç şehrinden diğer şehirlere olan en kısa maliyetler bir diziye yazdırılır aynı zamanda güzergahta bir dizide tutulur ve txt dosyasına yazdırılır. Kullanıcının istediği problemin çözümü bulunur.

*1.Giriş*

Gezgin zeplin projesi genel olarak txt dosyasından alınan bilgiler ile maliyet hesaplayıp bu maliyetler ile graf üzerinden işletilen algortimalar ile maliyet ve kar hesaplama programıdır. Projede gerçeklenen kısımlar şunlardır:

* Txt dosyasından verileri okuma ve saklama
* Veriler ile bir graf yapısı oluşturma
* Şehirler arası maliyetleri bulma ve grafı güncelleme
* Graf üzerinden en kısa yol algoritması yürütme
* En kısa yolları ve güzergahları bulma
* %50 Kar problemi
* Sabit ücretle kişi sayısı problemi

Bu projede amaç bir turizm şirketi için en az maliyet ve en çok karlı olan yolları bulmaktır. Aynı zamanda veri yapıları dersinde teorik olarak öğrenilen graf ve en kısa

yol konularının uygulamada kullanılması ve bilgilerin kalıcılığını sağlamaktır.

*2.Temel Bilgiler*

Gezgin zeplin projesinde programlama dili olarak C dili ve geliştirme ortamı (IDE) olarak Code blocks ve DevC++ kullanılmıştır.

**C Programlama Dili:**1970'lerin başında Ken Thompson ve Dennis Ritchie tarafından UNIX işletim sistemi için geliştirilmiş yüksek seviye bir programlama dilidir.Daha çok sistem programlama için kullanılan bir dildir.C# ve C++ dilleri C dilinden türetilmiştir.

**Geliştirme Ortamı (IDE):** IDE bilgisayar programcılarının hızlı ve rahat bir şekilde program geliştirebilmesini amaçlayan, geliştirme sürecini organize edebilen birçok araç ile birlikte geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür.

*3.Diğer Bölümler*

*3.1.Proje Bölümleri*

*3.1.1Ana Fonksiyon*

Ana fonksiyon başlangıçta verilen txt dosyasından bilgileri okumak için kullanılan fonksiyonu çağırır ve böylece ana fonksiyon içinde şehir bilgileri okunmuş olur. Daha sonra kullanıcıdan başlangıç ve bitiş şehirlerini girmesini ister. Ve hesaplanması istenen problem seçilmesi istenir ve seçilen fonksiyon çağırılır.

*3.1.2.Lat Long Oku Fonksiyonu*

Bu fonksiyon programın bulunduğu dizindeki lat long dosyasını açarak dosyadaki sırasıyla lat long plaka ve rakım bilgilerini okur ve struct yapısındaki gerekli yere atar daha sonra aynı fonksiyon her bir şehrin komşu şehirlerini okur anca komşu şehirleri bir string şeklinde okur daha sonra ayrı bir fonksiyon ile bu string ayrılır. bu okuma işlemlerinde dosya fopen fonksiyonu ile açılır ve bilgiler fscanf fonksiyonu ile okunur.

*3.1.3.Graf Oluştur Fonksiyonu*

Graf 82x82 lik bir matriste tutulur matrisin sıfırıncı satur ve sütunu boştur buda illere plakaları ile ulaşılabilmesini sağlar. İç içe for döngüsü ile ilk önce grafın bütün değerlerine -1 atanır daha bir for döngüsü ile daha düğümün diğer düğümler ile komşu olup olmadığına bakılır komşu ise 1 atanır değilse bir şey yapmadan geçilir ve böylece komşu olanlar 1 ile işaretlenerek maliyetsiz graf oluşturulmuş olur.

*3.1.4.Lat Long İle Mesafe Hesaplama Fonksiyonu*

Bu hesaplama formülü Haversine formülü kullanılarak yapılmıştır. İki şehrin latitude farkı dlat ve longtitude farkı dlon hesaplanarak bu verilerin cos ve sinüs leri ile hesaplanan başka bir değerin dünyanın yarıçapı ile çarpılmasıyla bulunur bu mesafe şehirler arasındaki yatay mesafedir.

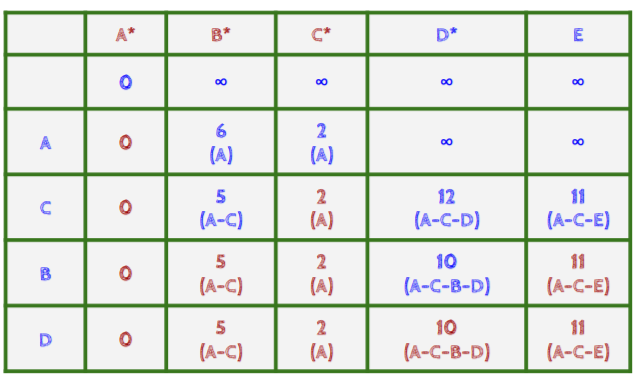
*3.1.5.Grafın Maliyetini Bulma Fonksiyonu*

Bu fonksiyon iç içe for ile ilk önce şehirler arasındaki yatay mesafeyi bulmak için fonksiyon çağırır daha sonra şehirler arasındaki rakım farkını bulur eğer başlangıç şehri ise 50 eklenir değilse eklenmez dikey mesafede bulunduktan sonra net mesafe hesaplanır net mesafe dikey mesafenin km cinsinden karesi çarpı yatay mesafenin karesinin toplamının kareköküdür. Daha sonra şehirler arasındaki açı hesaplanır ve açı kontrolü yapılır eğer açı 80-kişi sayısından büyükse o şehre gidilemez yine grafta güncellenmez ve -1 olarak kalır. Eğer açı uyumluysa net mesafe grafta güncellenir.

*3.1.6.En Kısa Yol Algoritması*

En kısa yol algoritmasında dijkstra algoritması kullanılmıştır.Bu fonksiyona graf başlangıç ve bitiş şehirleri gönderilir daha sonra işleme alındı adında bir dizi ve anlık düğüm adında değişkenler oluşturulur biri gidilen bir şehre tekrar gidilmemesini sağlar diğeride anlık olarak ele alınan düğümü ifade eder.Ayrı olarak maliyetler adında bir dizi daha oluşturulur bu dizide başlangıç şehirleri ile diğer şehirlerin en kısa mesafesini tutar.Algortimaya gelecek olursak Dijkstrada ilk şehir olarak başlangıç şehri alınır ve anlık düğüm değişkenine atanır daha sonra bütün maliyetler sonsuz yapılır ve iç içe for içinde anlık düğümün diğer düğüm ile komşuluğu olup olmadığına ve işleme alınıp alınmadığına bakılır eğer bunlar uyuyorsa ve yeni maliyet eski maliyetten küçükse maliyetler dizisi güncellenir. İçteki for bittiğinde yeni düğüm ele alınmalıdır bunun içinde greedy yaklaşımı kullanılır.Bu yaklaşımda sonraki ele alınan düğümün en az maliyetli düğüm olması gerekmektedir. Bunun içinde döngü içinde maliyetler dizisinden en az maliyetli düğüm bulunur ve o anlık düğüm değişkenine atanır ve gidildi olarak işaretlenir. Örnek bir djikstra algoritması örneği:

A->B=6,A->C=2,B->C=3,B->D=5,C->E=9,D->E=2

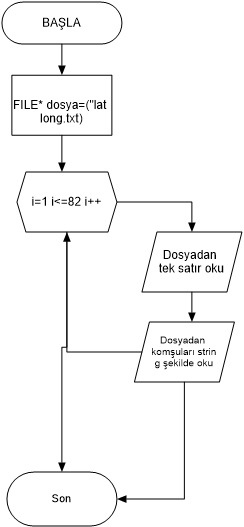


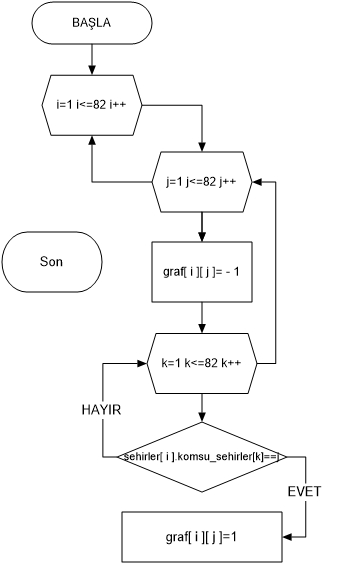
*3.1.7. Kar Problemleri*

İki problem çözen iki fonksiyon bulunmaktadır. Birincisi %50 kar problemidir bu problemde diğer problemle ortak olarak başlangıç ve bitiş şehirlerine göre graf oluşturulur ve maliyetler hesaplanarak graf güncellenir. Ve dosyaya maliyetler güzergahlar bir txt dosyasına yazılır. Grafta başka bir txt dosyasına yazılır. Daha sonra %50 kar probleminde yolcu sayısı beşer kişi arttrılarak hedef şehrin maliyeti ve %50 kar için ne kadar ücret alınması gerektiği yazılır. Diğer problemde kişi sayısı birer arttırılarak hedef şehre giden maliyet ve güzergahlar yazdırılır ve kişi başı 100 tlden maksimum kar sıralaması yapılır.

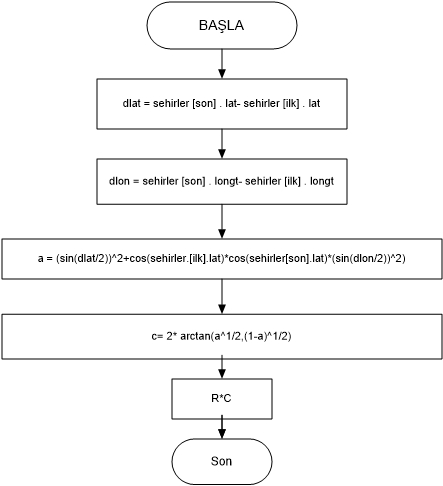
*3.2. Akış Şemaları*

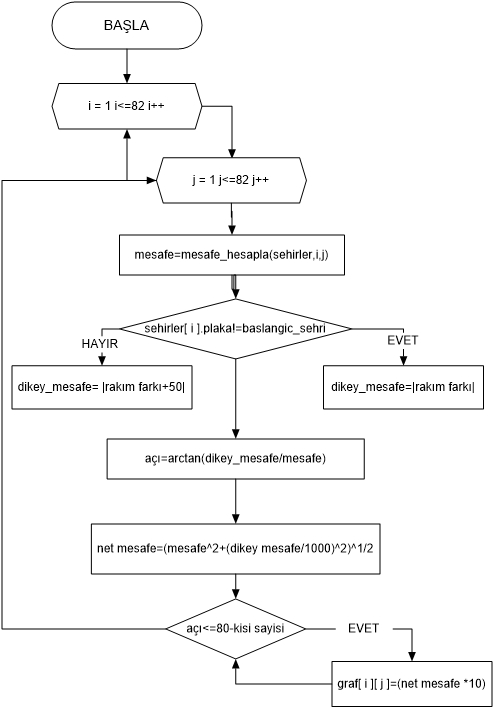
*3.2.1.Dosyadan Okuma*

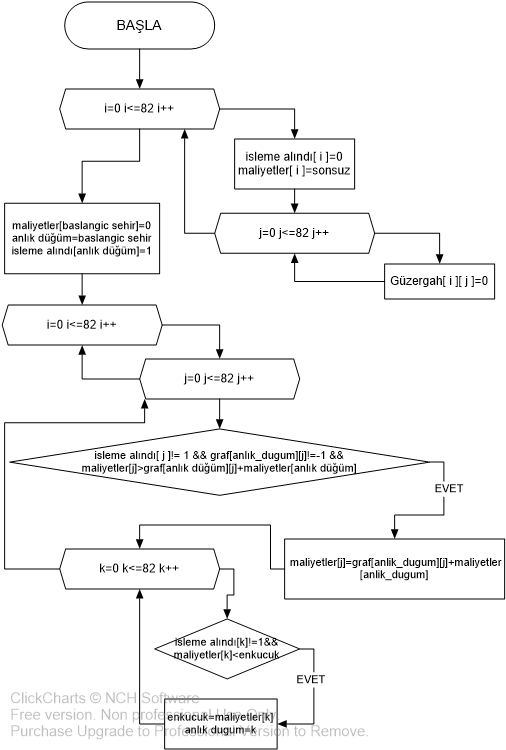


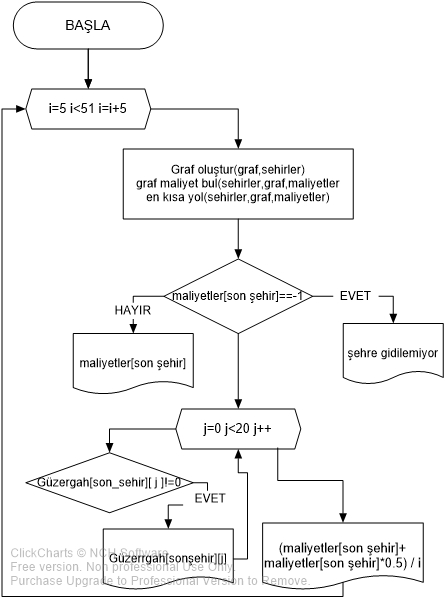
*3.2.2.Graf Oluşturma*

*3.2.3.Mesafe Hesaplama*

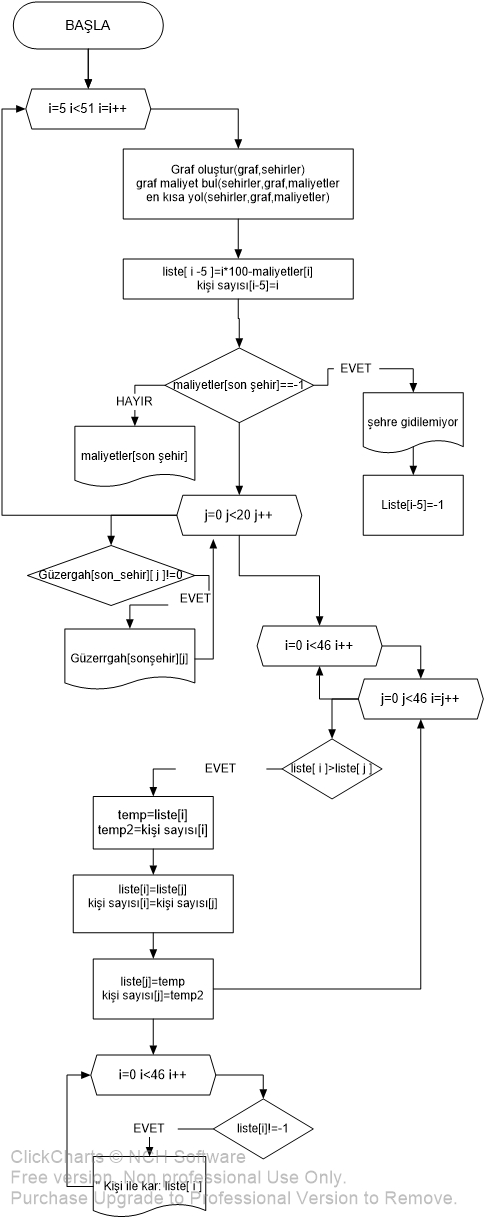


*3.2.4.Graf Güncelleme*

*3.2.5.En Kısa Yol Algoritması*

*3.2.6.%50 Kar Problemi*

*3.2.7.Sabit Ücret ile Kar Problemi*



*4.Sonuçlar*

Projede

* Şehir bilgileri txt den okunmuş ve struct yapısında tutulmuştur
* Graf yapısı kullanılmış ve graf matriste tutulmuştur
* Değişen kişi sayısına göre graf güncellenebilmektedir
* Yatay ve dikey mesafe doğru olarak hesaplanmış ve eğim kontrolü yapılmaktadır
* Maliyetler, güzergahlar ve graf txt dosyasına yazılmıştır
* Graf üzerinde en kısa yol algoritması çalıştırılmış ve maliyetler ve güzergâh bulunmuştur
* Çıkan sonuçta güzergahlar harita üzerinde görsel olarak gösterilememektedir.

*5.Kaynakça*

[1] Yorulmaz,M. ve Yorulmaz,S. “Programlamayı C ile Öğreniyorum”,2016

[2] Gök,O. ve Şahin,S.”En Kısa Yol Algoritmaları:Dijkstra”, <http://embedded.kocaeli.edu.tr/veri_1718_12/>

[3] Çölkesen R.,”Veri Yapıları ve Algoritmalar”,2017