

**İSTANBUL OKAN ÜNİVERSİTESİ**

**MESLEK YÜKSEKOKULU**

**BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ**

**YAPAY ZEKA**

**MİNİMUM YOLUN BULUNMASI**

**PROJE İLERLEME DURUM RAPORU**

**BÖLÜM**

**BiLGİSAYAR PROGRAMCILIĞI**

**NUMARA**

**18MY03015**

**PROJEYİ HAZIRLAYAN ADI-SOYADI**

**EMİRHAN KARAPINAR**

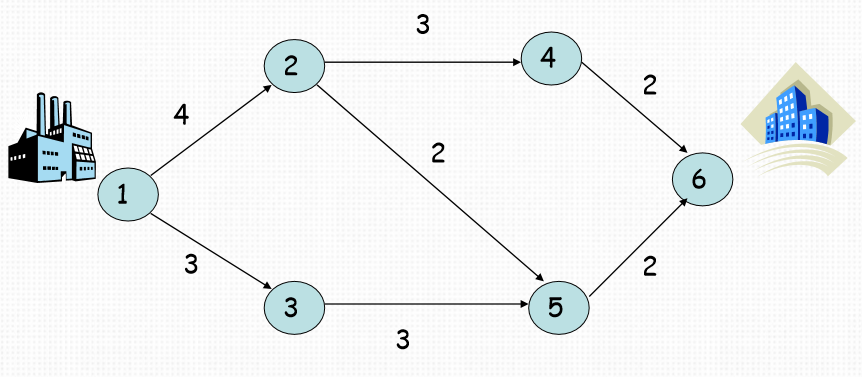
**DANIŞMAN/ÖĞRETMEN**

**NİLGÜN İNCEREİS**

**UYGULAMA ORTAMI VE PROBLEMİN YAZILACAĞI DİL :**  Netbeans(Java)

**MİNİMUM YOLUN BULUNMASI ARAŞTIRMASI**

Minimum yolun bulunması probleminin ilk olarak ne olduğunu,ne işe yaradığın ve bize hangi kolaylıklar sağladığını araştırdım.Bunlar ile ilgili bilgiler şunlar: Ayrıtları, iki düğüm arasındaki uzaklığı gösteren bir serimde, başlangıç düğümden (kaynak düğüm), verilen bir bitiş düğümüne (varış düğümü) toplam ayrıt uzunluğu en küçük olan güzergahı bulma problemidir.Örnek olarak Powerco bu örneğin ne demek olduğunu açıklayım: Birinci santralden birinci şehre olan gönderimin aktarma istasyonları vasıtasıyla gerçekleştiğini ve santral ile birinci şehir arasında 4 adet aktarma istasyonu olduğunu farz edelim. Santral ve şehir arasındaki en kısa yolu bulmak için serim modellerinden örnek olarak faydalanabiliriz.Resimle örneklemek bu örneklemeyi düğüm olarak gösterelim:



Kaynak santralin 1 den 6 noktasına en kısa gidişini 1-3-5 olarak gözüküyor.Fakat Aynı zamanda 1-2-5 yolundan giderek de aynı zamanda minimum olarak diğer binaya varmaktadır.

Peki bizim bu bahsettiğimiz minimum yolun bulunması problemleri ne gibi kolaylıklar sağlıyor işte yukarıdaki örneğimizden gösterirsek:1-2-4 olarak gitseydi toplam 9 dakikada alacaktı fakat bu minimum yolun bulunması problemi sayesinde 2 tane yolu kullanarak aynı zamanda ve farklı yollardan karşı adrese ulaşabiliyor.

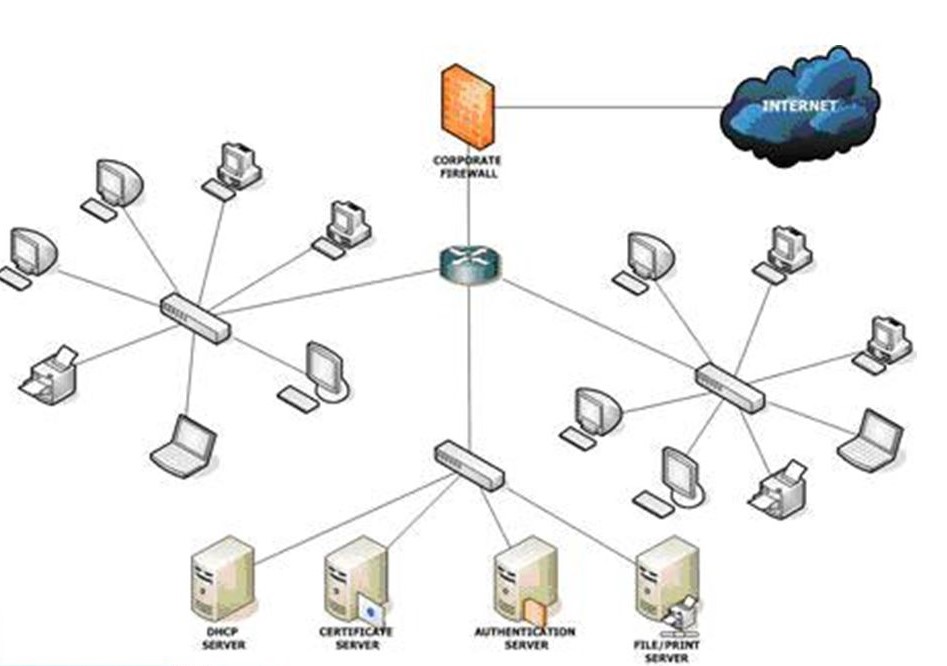
Bunu sadece düğüm olarak aklınıza getirmeyin örneğin evden okula gideceksiniz ev ile okul arası çok uzak bir mesafe olduğunu varsayalım.Evden gitmek için 2 tane yolunuz var bu yolların hangisini kullanarak en hızlı şekilde varacağımızı bu minimum yol problemini kullanarak bulabiliriz hadi başlayalım:

1.Yol evin arka sokağına 10 dakika da varabiliyoruz oradan saat 12.30 da kalkan arabaya binip aktarma yapma ihtimalimiz var fakat tam denk getirebilmemiz için saat 12.20 de evden çıkmış olmamız gerekir denk getirdiğimizi düşünürsek bu araba 15 dakika’da aktarma yerine varabiliyor.

2.Yol ise evin önünden kalkan arabaya binip aktarma yerine gideceğiniz yere kadar sizi bırakan araca binmek bu araç evin önünden katlığı için yürüme dakikanızı ortadan kaldıracak ve aktarma yerine hızlıca 20 dakikada varacaksınız.Aynı saatte çıktığımızı var sayar isek 12.20 direk araca binerek 20 dakika giderse 12.50 de aktarma yapan yerde inmiş olur.Bu iki yolun hangisi ile daha hızlı varma yolunu bulmamız aynı zamanda minimum yolun bulunmasına bir örnektir.2.Yolu kullanarak 5 dakika önceden varabiliyoruz fakat 1.Yolu seçseydik 5 dakika geç gelip aktarma yapacağımız aracımızı kaçırma ihtimalimiz olacaktı.Sonuç olarak bu örneğimizde en yakın hedefe ulaşabilmemiz yani minimum yolun bulunması problemini çözmüş olduk 2.Yol.

Bu araştırdığımız ve resim ile açıkladığımız örneğin genel ismi “GRAF”’dır.Grafın teorisi ve veri modelinden söz edelim:

1736 Yılında Ünlü Matematikçi “Leonhard Euler” tarafından önerilmiştir.Bu öneriyi “Könsiberg Köprüsün”’de köprünün probleminin çözümünde kullanılmıştır.Bu Graf teorisi bir problemin veya bir olayın düğüm ve çizgiler kullanılarak anlatılması,biçimlendirilmesidir.Graf modellemesi tarihi 18.yüzyıla dayanmaktadır.Graf teorisi günümüzde bilgisayar uygulamalarında ve genelde modellemede kullanılmaktadır.Aynı zamanda bazı mühendislik alanlarında Graf teoremi kullanılarak dersler sürdürülmektedir.Bilgisayar bağlantısının Graf ile gösterimi ağların arasındaki bağlantıları ve nereye bağlandığını göstererek olur.Bunu resim ile ifade etmek gerekirse:



Graf’ın kavram olarak ve nerelerde tanımlanıyor bilgisinide açıklayayım:

Garflar ayrık matematik dediğimiz matematik dallarından yapısında önemli bir role sahiptir.Graflar matematiksel konular arasında bir sürü geliştirilen teoremler ve yazılım diyarında algoritimsik yapılarda yoğun olarak kullanılmaktadır.Graf matematiksel tanım olarak;çok sayıda düğümler ve bu düğümlerin arasındaki bağlantıyı gösteren bir küme olarak açıklanmıştır.Matematiksel olarak d tane düğüm ve k tane kenar varsa bunlar arasındaki ilişkilendirme düğüm ve kenarların birbiri arasında bağlantı olarak gösterilir.Kümesel olarak açıklamak gerekirse:

D={n.0,n.1,n.2 …….n.d-1,n.d} Bu küme Değerler kümesidir

K={m.0,m.1,m.2…..m.k-1,m.k} Bu küme Kenarlar kümesidir

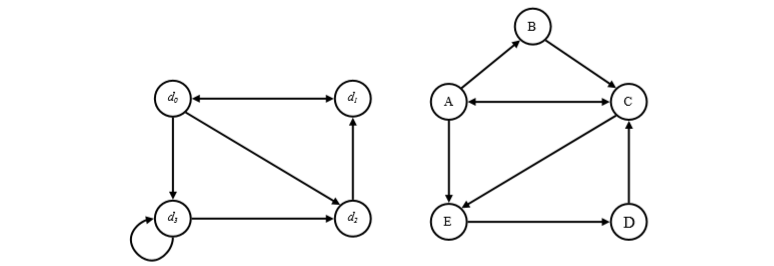
G={D,K} Graf’ın Küme ile gösterilmiş halidir.

Graf’ın Türlerinden Birini Açıklayayım Yönlü Graf Aynı zamanda Yönlendirilmiş Graf diye duyabiliriz.Bunların Tanımı Şu şekilde:

Bir grafın üzerindeki kenarlar bağlantılarının nereden başlayıp nerede sonlandırıldığını belirten graflara denir.Yönlü graf iki düğüm arasındaki bağlantının olduğunu,arasındaki ilişkinin de oka bağlı olaraktan tek yönlü olarak gösterilir.Eğer düğümler arasındaki ilişkiler karşılıklı ise zıt yön işareti kullanılarak belirtilir.Bu grafın matematiksel gösterim ifadesinde “< >”bu karakterlerle gösterilir.Yönlü grafların komuşuluk matrisinde her bir bağlanti bire bir bir düğüme bağlı değil ise simetrik bir matris olamazlar.Bu düğümlerin gösterim şekli karşılıklı şunlardır:

G.dd ={<d.0,d.1>,<d.0,d.2>,<d.0,d.3>,<d.1,d.0>,<d.2,d.1>,<d.3,d.2>,<d.3,d.3>}

İki düğüm arasındaki karşılıklı bağlantı bulunuyorsa farklı yönde iki kenar kullanılmaktadır.Örnekteki “<d.0,d.1>” ve “<d.1,d.0>” aralarındaki bağlantıyı gösteriyor.Yönlü graf örneği olarak göstermek gerekirse:



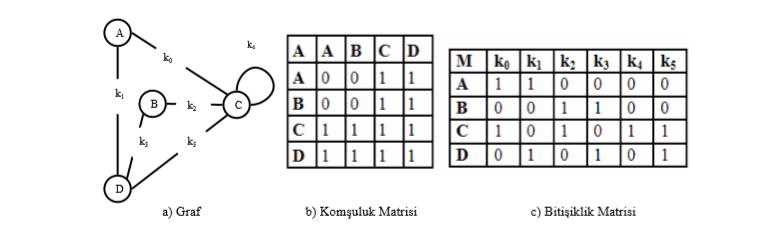
Komşuluk Matris ile ilgili açıklama:

Düğümlerden düğümlere arasındaki bağlantıları göstermek amaçlı kullanılan kare içindeki matrise komşuluk matrisi denir.Komşuluk matrisinin elemanları k.1 değerinden oluşur.Komşuluk matrisi G.dd ‘ nin matris biçiminde gösterilmesinden meydana gelir.

Bitişiklik Matris ile ilgili açıklama:

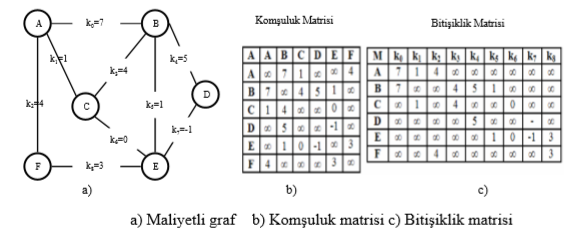
Düğümlerle kenarlar arasındaki ilişkiyi göstermek için kullanılan matrise bitişiklik matrisi denir.Bu matrisin satır sayısı düğüm, sutun sayısı kenar sayısı ile belli olur.Bitişik matrisi G.dk’nın matris şeklinde gösterilmesiyle oluşur.

Bu matrisleri resim ile örneklemek gerekirse:



Maliyetli Graf ile ilgili Açıklama:

Maliyetli graf dediğimiz şey grafın maliyetidir.kenarlarının pozitif veya negatif olabilen maliyet konusu olarak geçmektedir.Bu durumlarda normal grafda olduğu gibi bağlantı olmayan yok anlamında sıfır sayısı kullanılmaz.Nedeni sıfır maliyeti kenar olma ihtimali olabilir.Bunun yerine kullanılması gereken ifade olarak “∞” sonsuz ifade şekli belirlenmiştir.Maliyetli Grafa örnek göstermek gerekirse :



**PROJEYİ NASIL YAPILACAĞINI AÇIKLAMA**

Projeyi Netbeans programı üzerinden javaapplication oluşturarak java dili ile yazılacaktır.Bu projeye başlamadan önce minimum oylun bulunması probleminin ne demek olduğunu,projenin nasıl çalıştığını ve ne demek olduğunu araştırarak başlanacaktır.Projede ilk olarak minimum ve maksimum yolun belirlenecektir.Bir dizi oluşturarak bu yolun minimum maliyetini koruyacaktır.Başlangıç noktası(0,0) olacaktır.Tanımladığımız dizinin içine belirlediğimiz minimum ve maksimim yolu diziye belirtilecektir.Bir tane for döngüsü oluşturarak ilk önce maksimum yolun ve ardından bir tane daha for açarak minimum yolu yazılacaktır.Minimum yolun içine Belirlediğimiz “T” dizisinin içine maksimum ve minimum yollardan küçük olarak belirlediğimiz ifadeleri yazıyoruz.İlk satırı başlatmak amacıyla ve bitişik sol hücresinden başlaması için ise oluşturuyoruz.Eğer maksimum yoldan küçük ifade “0” a eşit aynı zamanda minimum yoldan küçük ifade “0”’dan büyük ise “T” dizisindeki elemanlardan minimum yoldan küçük ifadeden “1” çıkartılarak eklensin.ilk sutunu başlatmak amacıyla ve bitişik üst sutundan başlaması için değil ise ifadesi oluşturup; minimum yoldan küçük olan ifade “0”’a eşit ise ve aynı zamanda maksimum yoldan küçük olan ifade “0”’dan büyük ise “T” isimli oluşturduğumuz diziye maksimum yoldan küçük olan ifadeden “1” çıkartım dizimize ekliyoruz.Matrise ulaşmak,bitişik olan hücreleleri,sol hücre veya bitişik üst hücreden göstermek için değil ise ile başlıyoruz.Maksimum yoldan küçük ifade “0”’dan büyük ise ve aynı zamanda minimum yoldan küçük olan ifade “0”’dan büyük ise t dizimizden minimum ve maksimum yoldan küçük ifadelerden bir çıkarıp dizimize ekliyoruz.”T” hücresinin hedefe ulaşması için minimum maliyetini koruması için minimum ve maksimum yoldan “1”çıkarılarak geri dönmesini sağlıyoruz.Asıl main sınıfmızda bir maliyet sınıfı oluşturarak kendimiz bir matris belirliyoruz.Bu belirlediğimiz matris sınıfa göre program minimum maliyetli ve en kısa yolu bize söylemiş oluyor.