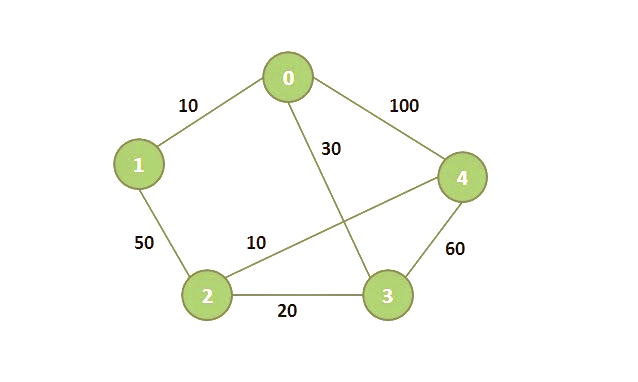
### [Gezgin Satıcı Problemi](https://www.matematiksel.org/gezgin-satici-problemi/)

**En Kısa Yol Algoritması** iki hedef düğüm arasında en az uğraş ile gidilebilecek yolun belirlenmesidir. Ödevin yapılmasında Dijkstra algoritması kullanılmıştır.

Dijkstra algoritmasının çalışma prensibi;

1. Düğümler arasında uzaklık değerleri belirlenmiş olmalı.
2. Bir başlangıç noktası belirliyoruz. Bu başlangıç noktası 0 noktamız.
3. Algoritma başlangıçta bütün düğümlere henüz erişim olmadığını kabul ederek sonsuz değeri atar.
4. Ardından başlangıç düğümünün komşusu olan bütün düğümleri dolaşarak bu düğümlere ulaşım mesafesini günceller.
5. Son düğüme gelene kadar bu işleme devam edilir.
6. Sonunda programımız bize en kısa yolun olduğu düğümleri gösterir.



Yandaki şekilde Dijkstra algoritmasını uygulayalım. 0 düğümü için en kısa yolu bulmak için verileri tutacak diziler oluşturulur. Bu diziler x’den y düğümüne gidilmesi için gereken yolu, yol uzunluğunu ve ziyaret edilen düğüm bilgilerini tutmak içindir. Dizilerin boyutu düğüm sayısı olacaktır. Diziler oluşturulduktan sonra (Yol[n], Uzunluk[n], Ziyaret[n]) işlem adımları şu şekilde devam eder;

1. Dizilerin elemanları ilk olarak varsayılan olarak doldurulur. Yol dizisi için tüm değerlerine 0 verilir. Bunun anlamı 0 düğümünden tüm düğümlere direk erişim vardır. Uzunluk dizisi için 0. düğümünün erişim olan düğümlere olan yol uzunluğu, erişim olmayan düğümler için sonsuzu temsil etmek için 9999 değerini girilir (komşuluk matrisinde 0. düğümünün ilişkileri alınır). Ziyaret dizisinde başlangıç düğümünü 1 yapılır. İşlemler sonucunda diziler;

Yol[n] = {0=>0, 1=>0, 2=>0, 3=>0, 4=>0}

Uzunluk[n] = {0=>0, 1=>10, 2=>9999, 3=>30, 4=>100}

Ziyaret[n] = {0=>1, 1=>0, 2=>0, 3=>0, 4=>0}

1. 0. düğüme en yakın ve ziyaret edilmemiş düğüm seçilir. Seçilen (1. Düğüm) düğüm ziyaret dizisinde işaretlenir. Ziyaret dizisinde henüz uğranmamış düğümlere bağlantısı var ise bu bağlantı yolu kendisine gelinen düğümün bağlantı yolu ile toplanıp uzunluk dizisi ile karşılaştırılır. Örneğin 2. Düğüm yol uzunluğu 50dir. 0. düğümünden 1. Düğüme yol uzunluğu 10dur. 50+10 <Uzunluk[2](9999) olduğundan Uzunluk[2] değeri 60 olur ve 2. Düğüme gidiş yolu 1 olarak belirtilir İşlemler bittiğinde dizilerimiz

Yol[n] = {0=>0, 1=>0, 2=>1, 3=>0, 4=>0}

Uzunluk[n] = {0=>0, 1=>10, 2=>60, 3=>30, 4 =>100}

Ziyaret[n] = {0=>1, 1=>1, 2=>0, 3=>0, 4=>0}

1. Seçilecek düğüm 3dür. 3. Düğüme bağlantısı olan 2. Düğüm için 30+20 <Uzunluk [2](60) olduğundan 2. Düğümün uzunluk değeri güncellenir. Gerekli düğüm kontrol işlemleri bittiğinde dizilerimiz

Yol[n] = {0=>0, 1=>0, 2=>3, 3=>0, 4=>3}

Uzunluk[n] = {0=>0, 1=>10, 2=>50, 3=>30, 4 =>90}

Ziyaret[n] = {0=>1, 1=>1, 2=>0, 3=>1, 4=>0}

1. Seçilecek düğüm 2dir ve son kontrol işlemidir (kontrol işlemi <n-1 kez tekrarlanır). 2. Düğümün 4 ile bağlantısı kontrol edilir. 50+10<Uzunluk[4](90) olduğundan 4. Düğüm uzunluk değeri 60 olarak değiştirilir. Gerekli düğüm kontrol işlemleri bittiğinde dizilerimiz

Yol[n] = {0=>0, 1=>0, 2=>3, 3=>0, 4=>2}

Uzunluk[n] = {0=>0, 1=>10, 2=>50, 3=>30, 4 =>60}

Ziyaret[n] = {0=>1, 1=>1, 2=>1, 3=>1, 4=>0}

İşlemler bittiğinde yol ve uzunluk dizilerinden istenilen kısa yol bulunur. Yol dizisini kullanmak için örneğin 2. Düğüme olan uzaklık. Yol[2] = 3dür. Bu durumda 2. Düğüme erişim 3. Düğümden sağlanmaktadır. Yol[3] = 0 yani başlangıç düğümü olduğundan durulur. Bu adımlar izlendiğinde en kısa yol ve uzaklıkları aşağıdaki gibi bulunur.

* Graf uzaklığı 1 = 10 Yol = 1<0
* Graf uzaklığı 2 = 50 Yol = 2<3<0
* Graf uzaklığı 3 = 30 Yol = 3<0
* Graf uzaklığı 4 = 60 Yol = 4<2<3<0

Mehmet Katı