



## Sezgisel Metotlar

-4-

### Şebeke Modelleri

- Üretim, Dağıtım, Proje planlama, Kaynak yönetimi, Finansal planlama ve benzeri problemlerde yaygın olarak kullanılmaktadır.
- Şebeke modelleri genel olarak 5 gruba ayrılırlar
  - Minimum dağılım (minimum kapsayan ağaç)
  - En kısa yol problemi
  - Maksimum akış problemi
  - Minimum maliyet akış problemi
  - Proje planlama ve kontrol

#### Minimum Dağılım Problemi

- Bir şebekede, tüm düğümleri birbirine bağlayan ve toplam en kısa mesafeyi veren bağlantıları veya yolları bulma problemidir
- Bu problemin optimum çözümü her zaman bir minimum kapsayan ağaçtır. Çünkü bağlantı sayısı her zaman düğün sayısından 1 eksiktir
- Uygulama alanları: iletişim ağları, telefon hatları, su tesisleri vb.

## Prim Algoritması

- Döğüm tabanlı bir algoritmadır. Bir kök döğüm ile başlar ve bütün döğümleri içine alıncaya kadar ağaç büyütölür. İşlem adımları;

$N = \{1, 2, 3, \dots, n\}$  şebekenin döğümler kümesi olsun.

$C_k$ : algoritmanın  $k$ . iterasyonda kalıcı olarak bağlanmış döğümler kümesi

$\bar{C}_k$ : Henüz kalıcı olarak bağlanmamış döğümler kümesi

1. Adım:  $C_0 = \emptyset$  ve  $\bar{C}_0 = N$  olarak belirle

2. Adım: Bağlanmamış döğümler kümesi  $\bar{C}_0$  deki herhangi bir «i» döğümü ile başlayıp  $C_1 = \{i\}$  ve  $\bar{C}_1 = \{N - i\}$  olarak güncelle

k. Genel Adım: bağlanmış döğümler kümesi  $C_{k-1}$  deki bir döğüme en kısa bağlantıyı sağlayacak şekilde,  $\bar{C}_{k-1}$  bağlanmamış döğümler kümesinden bir «j» döğümü  $C_{k-1}$  kümesine kalıcı olarak bağla yani :

$$C_k \leftarrow C_{k-1} + \{j\}, \quad \bar{C}_k \leftarrow \bar{C}_{k-1} - \{j\}$$

Bağlanmamış döğümler kümesi boş ise  $\bar{C}_k = \emptyset$  dur , aksi halde  $K=K+1$  olarak belirle ve genel adımı tekrar et.

## Prim Algoritması

Örnek 1:

## Prim Algoritması

Örnek 2:

## Kruskal Algoritması

- Bir şebekede, tüm kenarlar maliyet verilerine göre küçükten büyüğe doğru sıralanır.
- En düşük maliyetli kenardan başlayarak ağaç çizilmeye başlanır.
- Herhangi bir döngü oluşturan kenar ağaca dahil edilmez.
- Aynı maliyete sahip kenarlardan biri keyfi olarak seçilir. Bu durumda farklı şekillerde minimum yayılan ağaç ortaya çıkacak ancak toplam maliyet bütün ağaçlarda aynı olacaktır.

Örnek 1:

## Kruskal Algoritması

Örnek 2:

## En kısa yol problemi

En kısa yol problemi ağırlıklı bir şebekede herhangi bir düğümden diğer tüm düğümlere (veya herhangi bir düğüme) en kısa yolun bulunması problemidir. Dijkstra algoritması bu problemin çözümü için kullanılan en popüler algoritmalarından biridir.

### Dijkstra algoritması

- Çözüm zamanı  $O(n^2)$  ile sınırlıdır
- Açgözlülük (greedy) yaklaşımı ile çalışır.
- **Girdi:** tüm kenarları negatif olmayan  $G\{E,V\}$  şebek ve  $v \in V$  kaynak düğüm.
- **Çıktı:** verilen  $v \in V$  düğümünden diğer düğümlere olan en kısa yol uzunluğu hesaplanır.

## Dijkstra algoritması

### Pseudocode (sözde kod)

Dist [S] $\leftarrow$ 0	(kaynak düğüme uzaklık sıfır)
For all $v \in V - \{S\}$	
Do	
Dist [v] $\leftarrow \infty$	(diğer düğümlere olan uzaklıkları sonsuz olarak al)
R $\leftarrow \emptyset$	(R, ziyaret edilen düğümler kümesini başlangıçta boş olarak al)
Q $\leftarrow V$	(Q, başlangıçta tüm düğümleri içeren küme)
While Q $\neq \emptyset$	
Do	
U $\leftarrow \text{min distance}(Q, \text{dist})$	(Q'dan minimum uzaklığa sahip olan düğümü seç)
R $\leftarrow R \cup \{U\}$	{U'yu ziyaret edilen düğümler kümesine ekle }
For all $V \in \text{komşu}[U]$	
Do	
If Dist [U]+ W(U,V) < Dist [v]	(eğer yeni yol bulunur ise)
Dist [v] $\leftarrow$ Dist [U]+ W(U,V)	(en kısa yolun yeni değerini ata)
End while	
Return Dist	

## Dijkstra algoritması

Örnek 1:

## Dijkstra algoritması

Örnek 2: