



FLOYD WARSHALL ALGORİTMASI

SHORTEST PATH PROBLEM

FEVZİ AYVAZ

SUNUM PLANI

- FLOYD WARSHALL ALGORİTMASI
- ALGORİTMA NASIL ÇALIŞIR?
- SHORTEST PATH PROBLEM
- PİSEUDO CODE

Floyd Warshall Algoritması

ALGORİTMANIN AMACI:

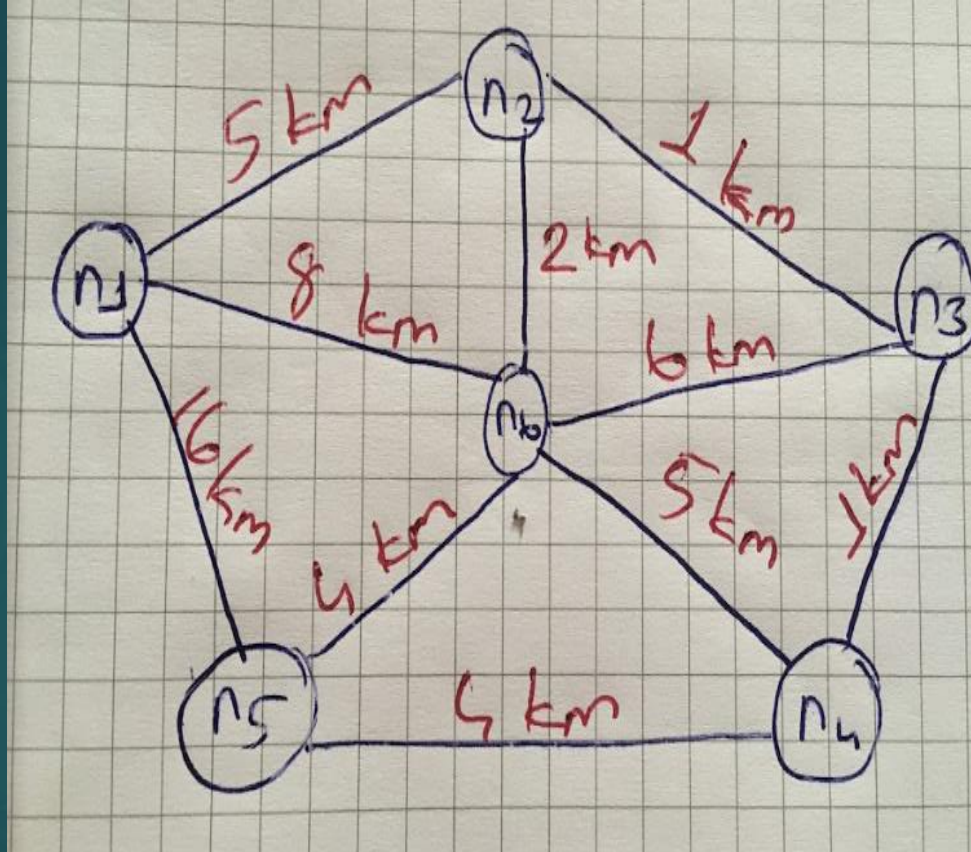
- Belirli bir graf üzerinde, **bir başlangıçtan**(source) bir **bitiş düğümüne** (sink, end, target) **en kısa yoldan** (shortest path) ulaşmaktır.
- Bu özelliğinden dolayı, **maksimum akış** (maximum flow) problemlerinde bu yöntem kullanılabilir.
- Örneğin bir dağıtım şebekesinde bir kaynaktan bir hedefe gönderilebilecek azami miktarı sevk eden problemlerin çözümünde kullanılabilen bir algoritmadır.
- Algoritmayı bulan iki kişinin ismi ile anılmaktadır.

Algoritma Nasıl Çalışır?

- Algoritma basitçe bir grafta gidilebilecek düğümlerin **komşuluk listesini (adjacency list)** çıkararak bu düğümlere olan mesafeyi tutan bir matris üzerinden çalışır.
- Algoritmanın her adımında matris üzerinde çeşitli işlemler yapılarak en kısa yol bulunmaya çalışılır.
- Algoritma matris üzerinde çalışan ve düğümler arası mesafeleri her adımda güncelleyen bir algoritma olduğu için iterative, döngü ile yazılabilir.

Shortest Path Problem

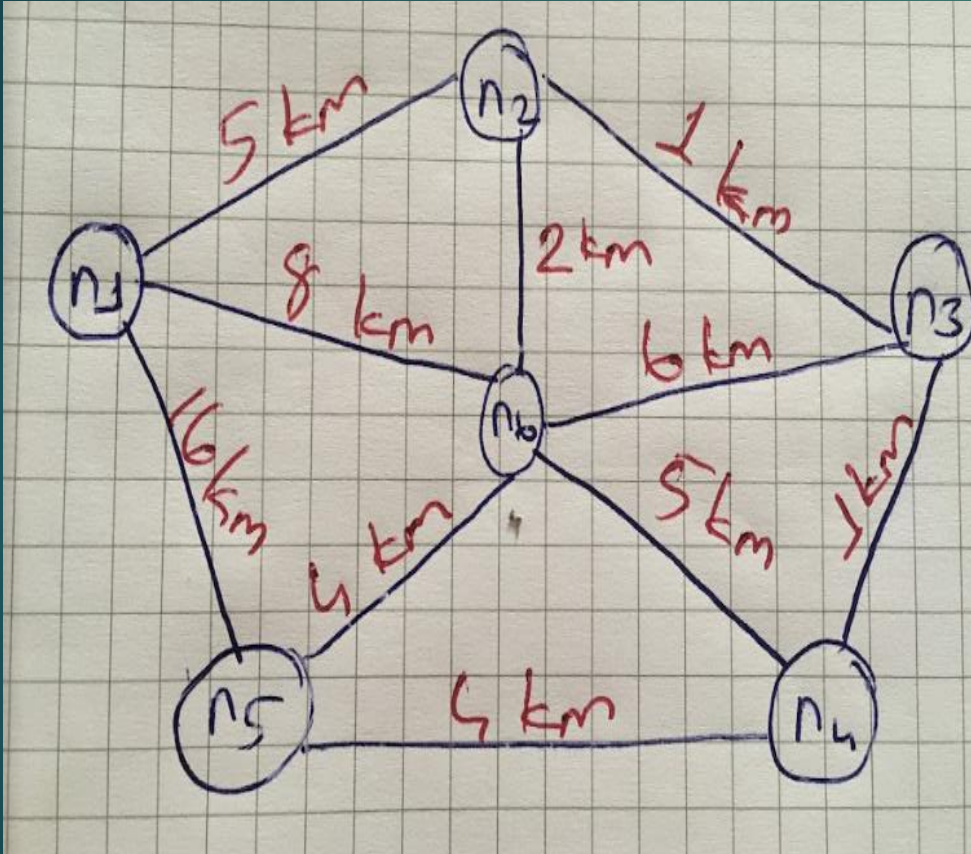
1.Aşama : İlk olarak yakınlık matrisinin oluşturalım.



	n1	n2	n3	n4	n5	n6
n1	0	5	∞	∞	16	8
n2	5	0	1	∞	∞	2
n3	∞	1	0	1	∞	6
n4	∞	∞	1	0	4	5
n5	∞	∞	∞	4	0	4
n6	8	2	6	5	4	0

Shortest Path Problem

2.Aşama : 2.adımda sıra matrisini oluşturulalım.



	n1	n2	n3	n4	n5	n6
n1	0	5	$\infty/6$	$\infty/7$	16/11	8/7
n2	5	0	1	$\infty/2$	$\infty/6$	2
n3	$\infty/6$	1	0	1	$\infty/5$	6/3
n4	$\infty/7$	$\infty/2$	1	0	4	5/4
n5	$\infty/11$	$\infty/6$	$\infty/5$	4	0	4
n6	8/7	2	6/3	5/4	4	0

Piseudo Code

1

```
int yol[][];  
procedure FloydWarshall ()  
    for k: = 1 to n  
        for each (i,j) in {1,..,n}2  
            yol[i][j] = min ( yol[i][j], yol[i][k]+yol[k][j] );
```

2

```
for i = 1 to N  
    for j = 1 to N  
        if (i'den j'ye bir yol varsa)  
            yol[0][i][j] = i ile j arasındaki mesafe  
        else  
            yol[0][i][j] = sonsuz  
  
for k = 1 to N  
    for i = 1 to N  
        for j = 1 to N  
            yol[k][i][j] = min(yol[k-1][i][j], yol[k-1][i][k]  
+ yol[k-1][k][j])
```

Dijkstra ,Bellman-Ford ve Floyd Warshall Algoritması

- Dijkstra ve bellman-ford algoritmalarına göre daha avantajlıdır.
- Dijkstra ve bellman-ford algortimalarında bir tane başlangıç düğümü belirlenip diğer düğümlere olan en kısa yolu buluyoruz.
- Floyd warshall da ise ; bir matris şeklinde bütün olasılıkları değerlendirerek en kısa yolu buluyoruz. Yani sadece bir noktan bir nokta giden değerleri değil, diğer noktalardan da hedef düğüme giden yolları buluyoruz.



DİNLEDİĞİNİZ İÇİN
TEŞEKKÜR EDERİM.