Floyd Warshall

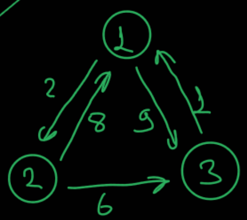
→ Bu algoritma bir grafta bir düğümden diğer herhangi bir düğüme gitmek için kullanılabilecek yollardan en kısasını tespit etmek için kullanılır. Dinamik bir yaklaşımdır.

→ $d^{k}_{i,j}$ → i'den j'ye en kısa yolun uzunluğudur.

$$d^{k}_{i,j} = \begin{cases} l(i,j) & \text{if } k=0 \\ \min\left\{ d^{k-1}_{i,j} \,,\, d^{k-1}_{i,k} + d^{k-1}_{k,j} \right. & \text{if } 1 \le k \le n \end{cases}$$

Örnek

Yönlü grafın tablosunu oluşturalım.



1'den 1'e
1'den 2'ye
1'den 3'e
3'den 2'ye

Adım 1 → $D_0 = \begin{bmatrix} 0 & 2 & 9 \\ 8 & 0 & 6 \\ 1 & \infty & 0 \end{bmatrix}$

k=0

i=1, k=1, j=1'den n'e kadar

3

$D(1,1) = \min\{ \overset{0}{D_0(1,1)} , \overset{0}{D_0(1,1)} + \overset{0}{D_0(1,1)} \} = 0$

$D(1,2) = \min(\overset{2}{D_0(1,2)} , \overset{0}{D_0(1,1)} + \overset{2}{D_0(1,2)} \} = 2$

$D(1,3) = \min\{ \underset{9}{D_0(1,3)} , \underset{0}{D_0(1,1)} + \underset{9}{D_0(1,3)} \} = 9$

i=2, k=1, j=1'den n'e kadar

$D(2,1) = \min\{ \overset{8}{D_0(2,1)} , \overset{8}{D_0(2,1)} + \overset{0}{D(1,1)} \} = 8$

$D(2,2) = \min\{ \overset{0}{D_0(2,2)} , \overset{8}{D_0(2,1)} + \overset{2}{D_0(1,2)} \} = 0$

$D(2,3) = \min\{ \overset{6}{D_0(2,3)} , \overset{8}{D_0(2,1)} + \overset{9}{D_0(1,3)} \} = 6$

i=3, k=1, j=1'den n'e kadar

$D(3,1) = \min\{ \overset{1}{D_0(3,1)} , \overset{1}{D_0(3,1)} + \overset{0}{D_0(1,1)} \} = 1$

$D(3,2) \min\{ \overset{\infty}{D_0(3,2)} , \underline{\overset{1}{D_0(3,1)} + \overset{2}{D_0(1,2)}} \} = 3$ yeni

$D(3,3) \min\{ \overset{0}{D_0(3,3)} , \overset{1}{D_0(3,1)} + \overset{9}{D_0(1,3)} \} = 0$

1 nolu düğüme uğrayarak hedef düğüme gideceğiz k=1

$D_1 = \begin{bmatrix} 0 & 2 & 9 \\ 8 & 0 & 6 \\ 1 & 3 & 0 \end{bmatrix}$

Bu eleman değişti.

i = 1'den n'e kadar değişecek ve her i değeri için j'de 1'den n'e kadar gidecek. Ve bu hesap k=1'den k=n'e kadar yapılacak.

Hesap yükü = $O(n^3)$ olur

→ $k=1$ için hesap yaptık ve D_0 matrisini D_1 matrisine güncelledik. Şimdi aynı hesapları $k=2$ için yapacağız ve D_2 matrisini oluşturacağız. Güncel matrisimiz D_1 olduğu için hesaplarda onu kullanacağız.

$i=1$, $k=2$ için $j=1$'den n'e kadar

$$D(1,1) = \min\{D_1(1,1),\ D_1(1,2)+D_1(2,1)\} = 0$$
$$D(1,2) = \min\{D_1(1,2),\ D_1(1,2)+D_1(2,2)\} = 2$$
$$D(1,3) = \min\{D_1(1,3),\ \underline{D_1(1,2)+D_1(2,3)}\} = 8$$

yeni

$i=2$, $k=2$ için $j=1$'den n'e kadar

$$D(2,1) = \min\{D_1(2,1),\ D_1(2,2)+D_1(2,1)\} = 8$$
$$D(2,2) = \min\{D_1(2,2),\ D_1(2,2)+D_1(2,2)\} = 0$$
$$D(2,3) = \min\{D_1(2,3),\ D_1(2,2)+D_1(2,3)\} = 6$$

$i=3$, $k=2$ için $j=1$'den n'e kadar

$$D(3,1) = \min\{D_1(3,1),\ D_1(3,2)+D_1(2,1)\} = 1$$
$$D(3,2) = \min\{D_1(3,2),\ D_1(3,2)+D_1(2,2)\} = 3$$
$$D(3,3) = \min\{D_1(3,3),\ D_1(3,2)+D_1(2,3)\} = 0$$

$$\rightarrow D_2 = \begin{bmatrix} 0 & 2 & 8 \\ 8 & 0 & 6 \\ 1 & 3 & 0 \end{bmatrix}$$

$k=3$ için, D_2 tablosuna göre D_3 hesabı yapılacak

$i=1$, $k=3$ için $j=1$'den n'e kadar

$$D(1,1) = \min\{D_2(1,1),\ D(1,3)+D(3,1)\} = 0$$
$$D(1,2) = \min\{D_2(1,2),\ D(1,3)+D(3,2)\} = 2$$
$$D(1,3) = \min\{D_2(1,3),\ D(1,3)+D(3,3)\} = 8$$

$i=2$, $k=3$ için $j=1$'den 3'e

$$D(2,1) = \min\{D_2(2,1),\ \underline{D_2(2,3)+D_2(3,1)}\} = 7$$

Yeni

$$D(2,2) = \min\{D_2(2,2),\ D_2(2,3)+D_2(3,2)\} = 0$$
$$D(2,3) = \min\{D_2(2,3),\ D_2(2,3)+D_2(3,3)\} = 6$$

$i=3$, $k=3$ için $j=1$'den 3'e kadar

$$D(3,1) = \min\{D_2(3,1),\ D_2(3,3)+D_2(3,1)\} = 1$$
$$D(3,2) = \min\{D_2(3,2),\ D_2(3,3)+D_2(3,2)\} = 3$$
$$D(3,3) = \min\{D_2(3,3),\ D_2(3,3)+D_3(3,3)\} = 0$$

$$\rightarrow \underline{D_3} = \begin{bmatrix} 0 & 2 & 8 \\ 7 & 0 & 6 \\ 1 & 3 & 0 \end{bmatrix}$$

D_n olduğunda işlemlerin sonuna geldik demektir. Güncel tablomuz bu. Bu en kısa yolları içermektedir.

Bu soru için $3^3=27$ maliyet yaptık. Bu yaklaşımın maliyeti $O(n^3)$ ile sabittir. Sabitlikten dolayı $\Theta(n^3)$ ile gösterilir. (teta) Bellek maliyeti: $i \times j$ ($= n^2$) 'dir. Bu soru için $\Theta(n^2)$ 'den gider.

Θn^3 ve Θn^2 = Sabit

Brut Force = Her iki düğüm arasındaki, olası bütün yollar önce hesaplar ve sadece iki düğüm için 2^n maliyet ve her iki düğüm arasında hesaplanan tüm yollardan en kısa olanı arama algoritması kullanarak yapılacaktır