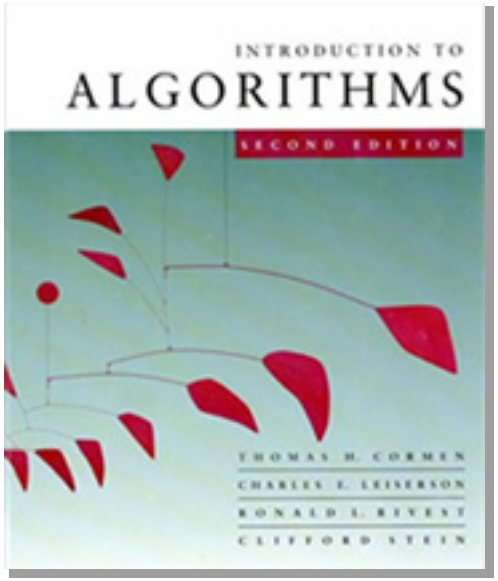
Algoritmalara Giriş

6.046J/18.401J

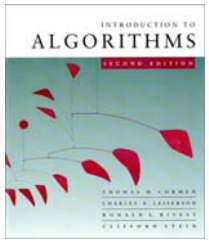


Ders 18

En Kısa Yollar II

- Bellman-Ford algoritması
- Doğrusal Programlama ve fark kısıtları
- VLSI yerleşimi küçültülmesi

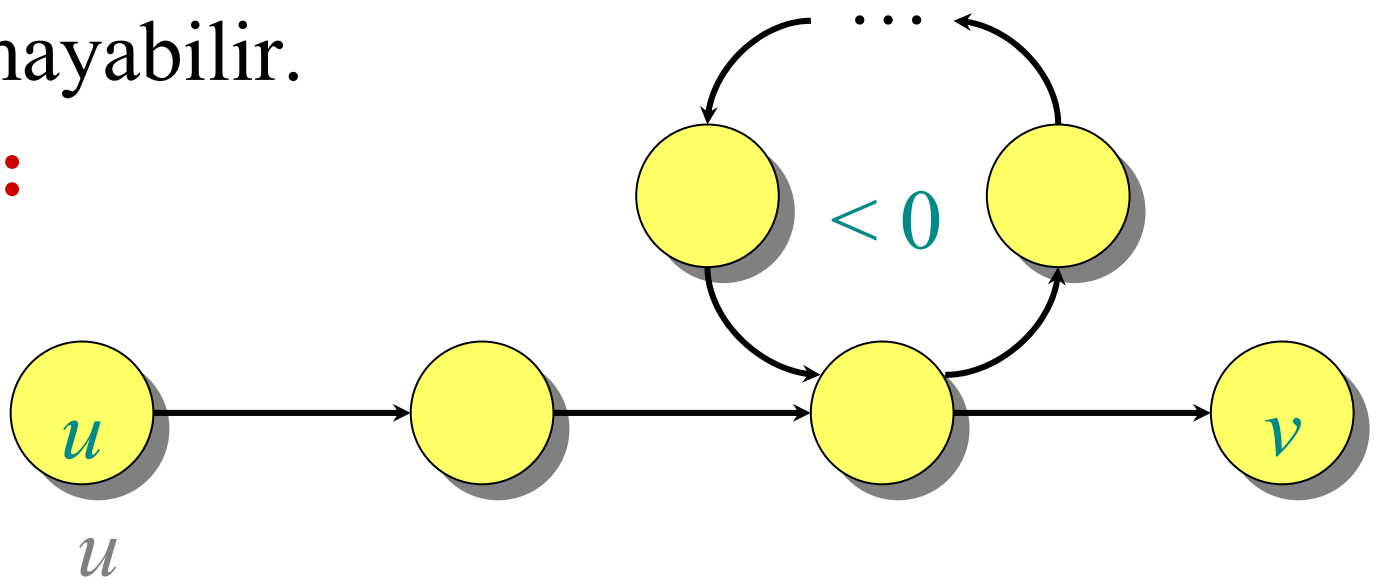
Prof. Erik Demaine

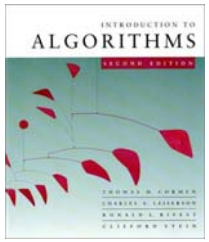


Negatif-ağırlık çevrimleri

Hatırlatma: Eğer grafik $G = (V, E)$ negatif ağırlık çevrimi içeriyorsa, en kısa yollardan bazıları bulunmayabilir.

Örnek:

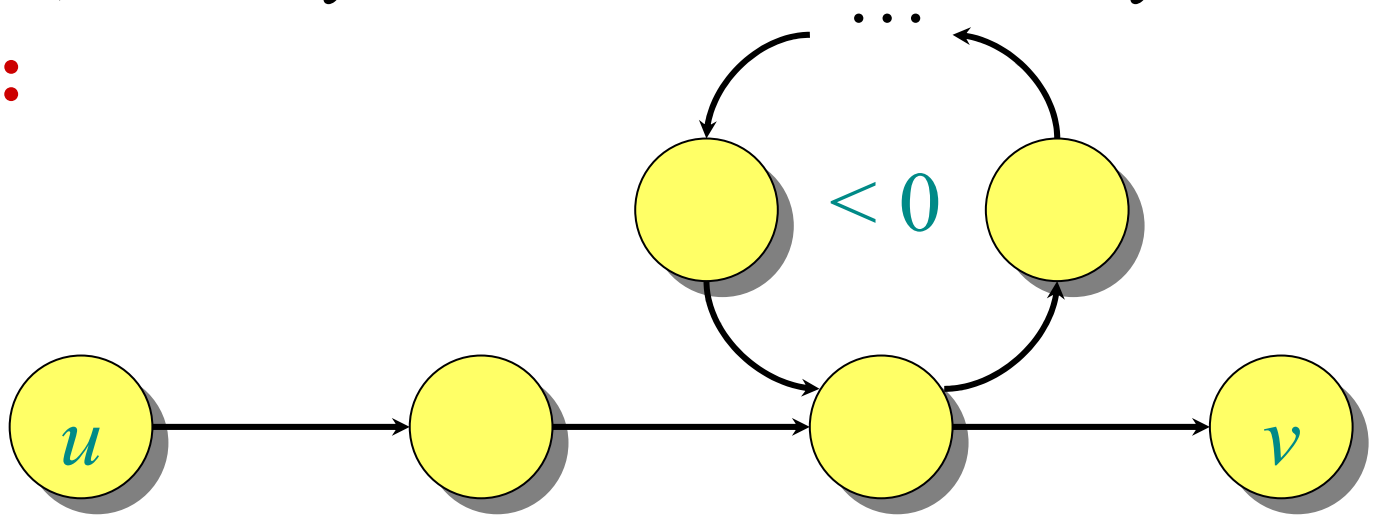




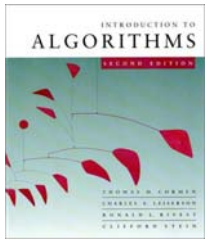
Negatif-ağırlık çevrimleri

Hatırlatma: Eğer grafik $G = (V, E)$ negatif ağırlık çevrimi içeriyorsa, en kısa yollardan bazıları bulunmayabilir.

Örnek:



Bellman-Ford algoritması: Bir $s \in V$ kaynağından tüm $v \in V'$ lere bütün kısa yol uzunluklarını bulur ya da bir negatif ağırlık çevrimi olduğunu saptar.



Bellman-Ford algoritması

$d[s] \leftarrow 0$
for each(her bir) $v \in V - \{s\}$(için)
    do(yap) $d[v] \leftarrow \infty$

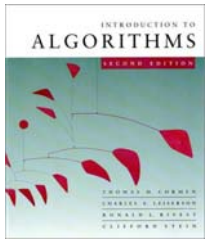
} ilklendirme

for(için) $i \leftarrow 1$ **to** $|V| - 1$ ' (e)
    do for each edge(her kenar için yap) $(u, v) \in E$
        do(yap) **if**(eğer) $d[v] > d[u] + w(u, v)$
            then(sonra) $d[v] \leftarrow d[u] + w(u, v)$

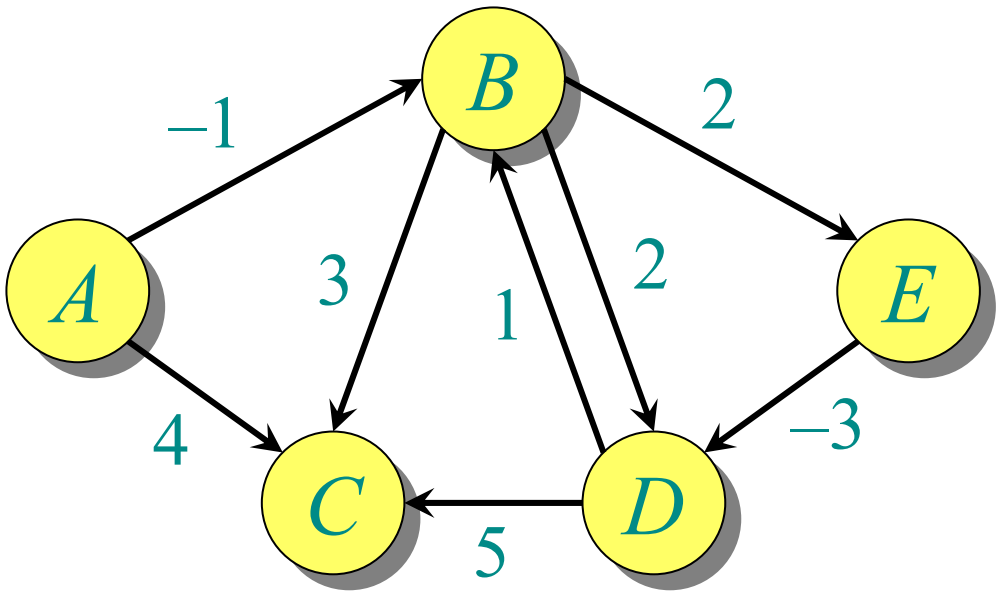
} ***Gevşetme adımı***

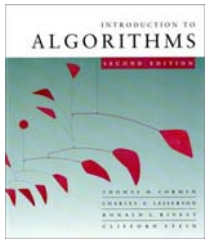
(her)**for** each edge $(u, v) \in E$ (kenarı için)
(yap eğer)**do if** $d[v] > d[u] + w(u, v)$
    sonra bunu negatif ağırlık çevrimi var diyerek raporla

Sonunda, $d[v] = \delta(s, v)$, negatif ağırlık çevrimi yoksa.
Süre $= O(VE)$.

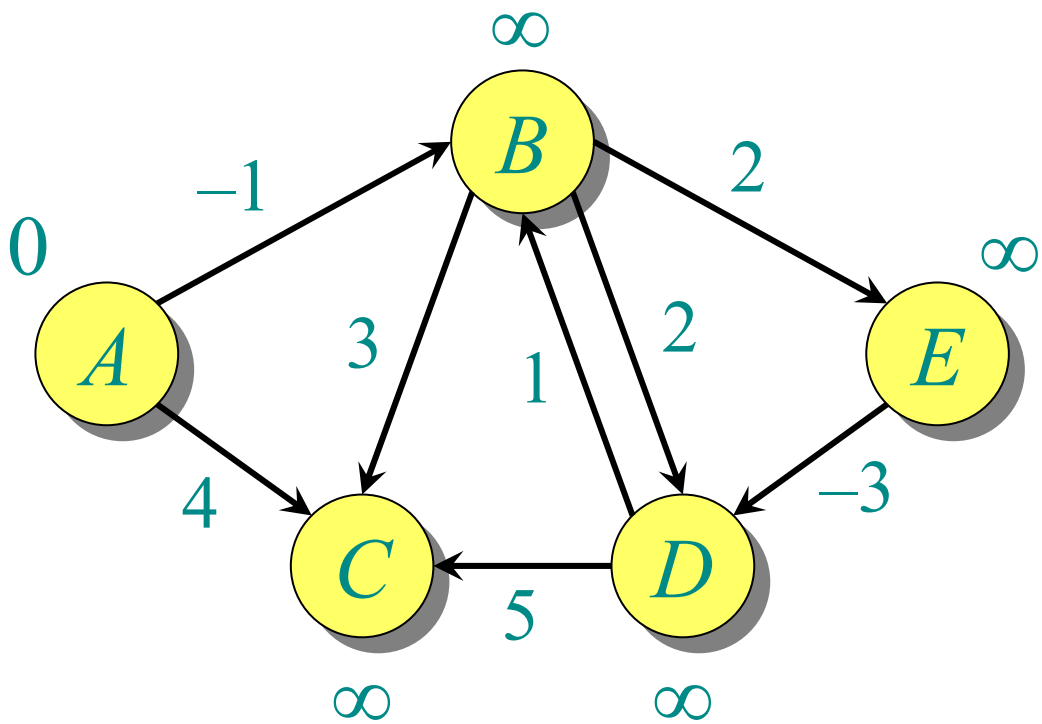


Bellman-Ford örneği

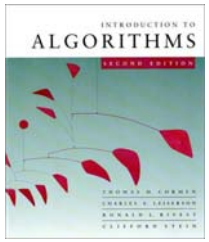




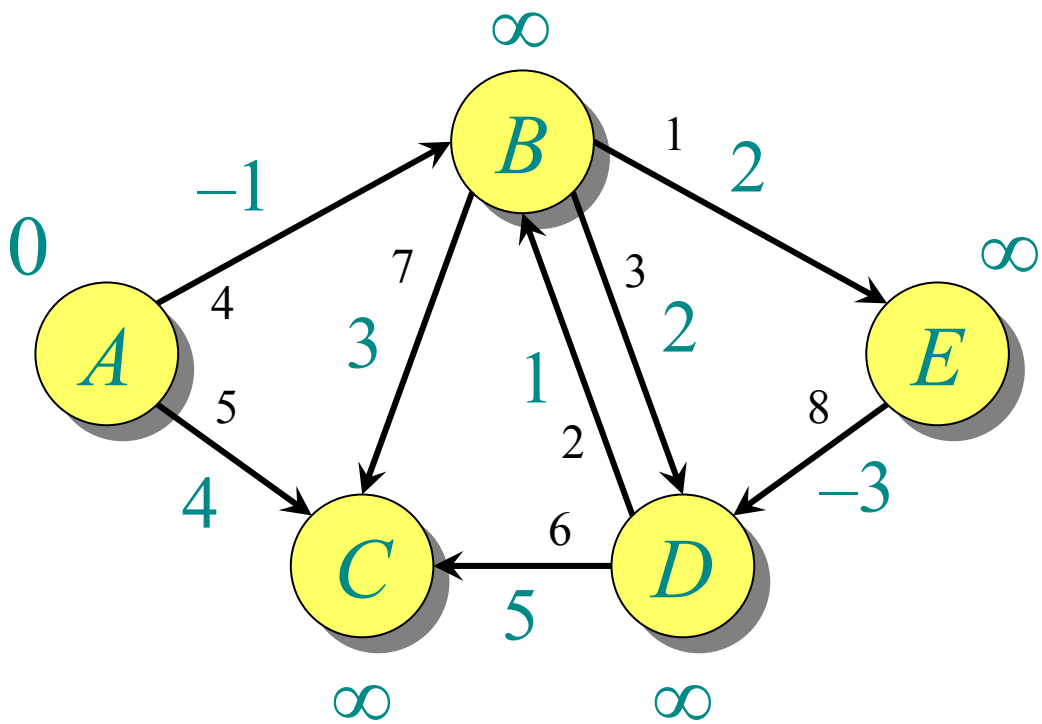
Bellman-Ford örneği



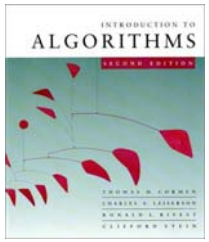
İlklendirme.



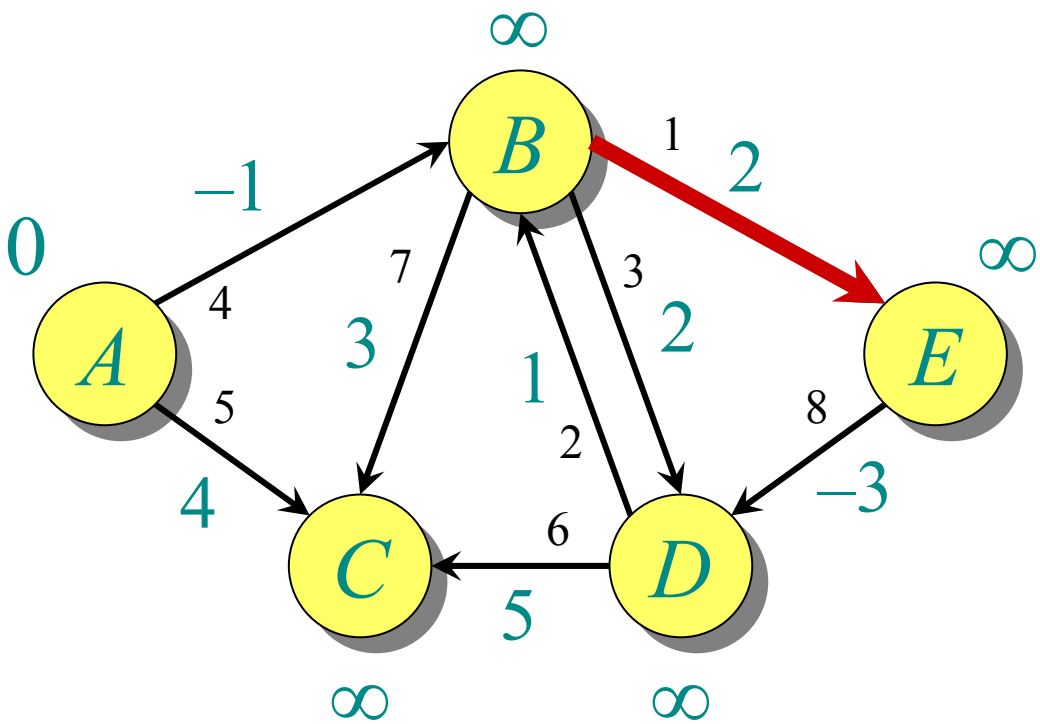
Bellman-Ford örneği

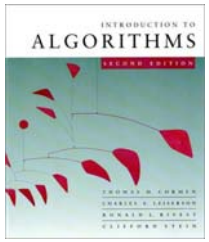


Köşe gevşetme düzeni

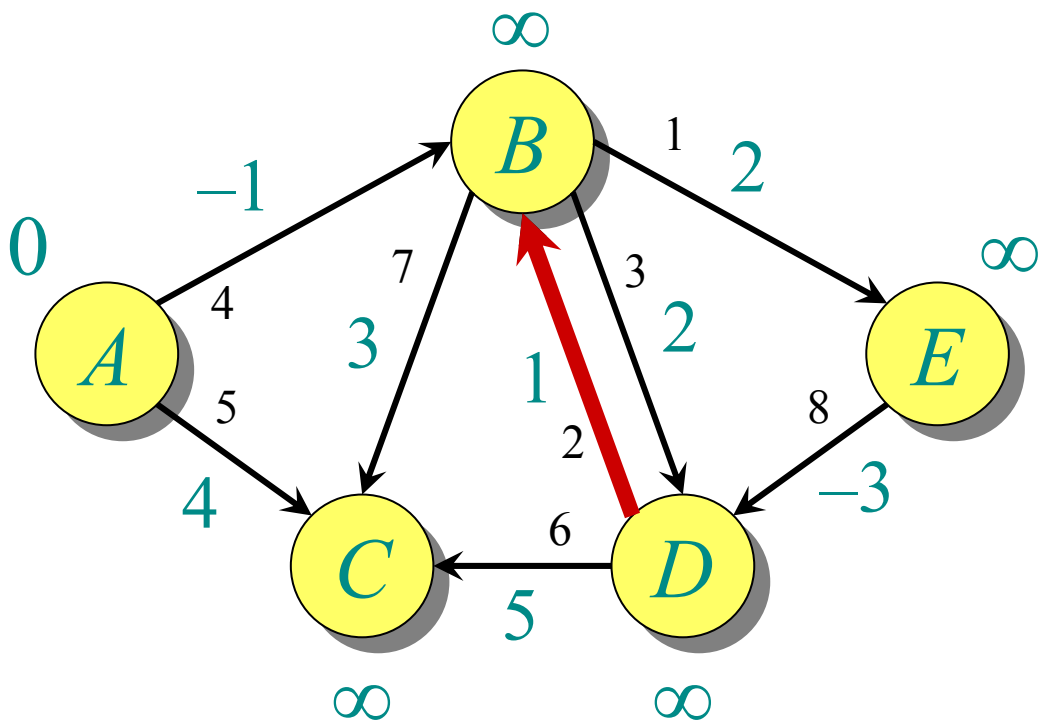


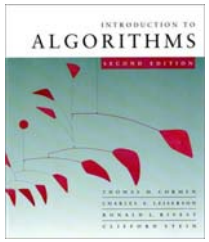
Bellman-Ford örneği



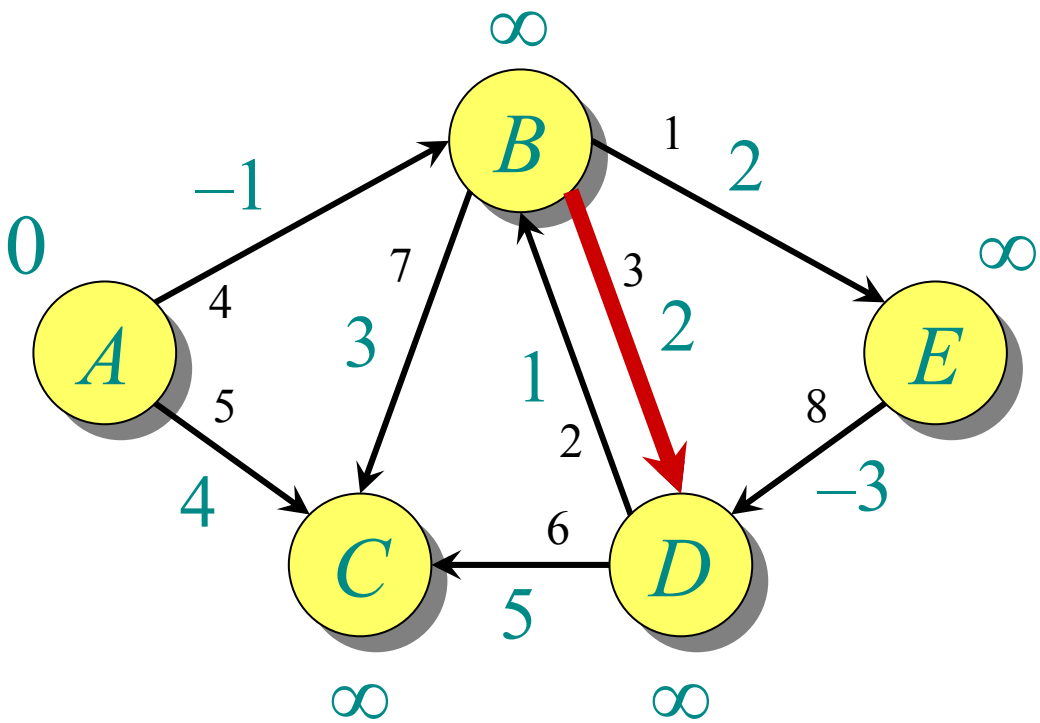


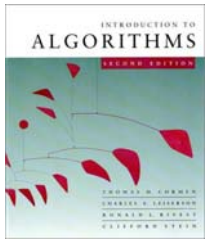
Bellman-Ford örneği



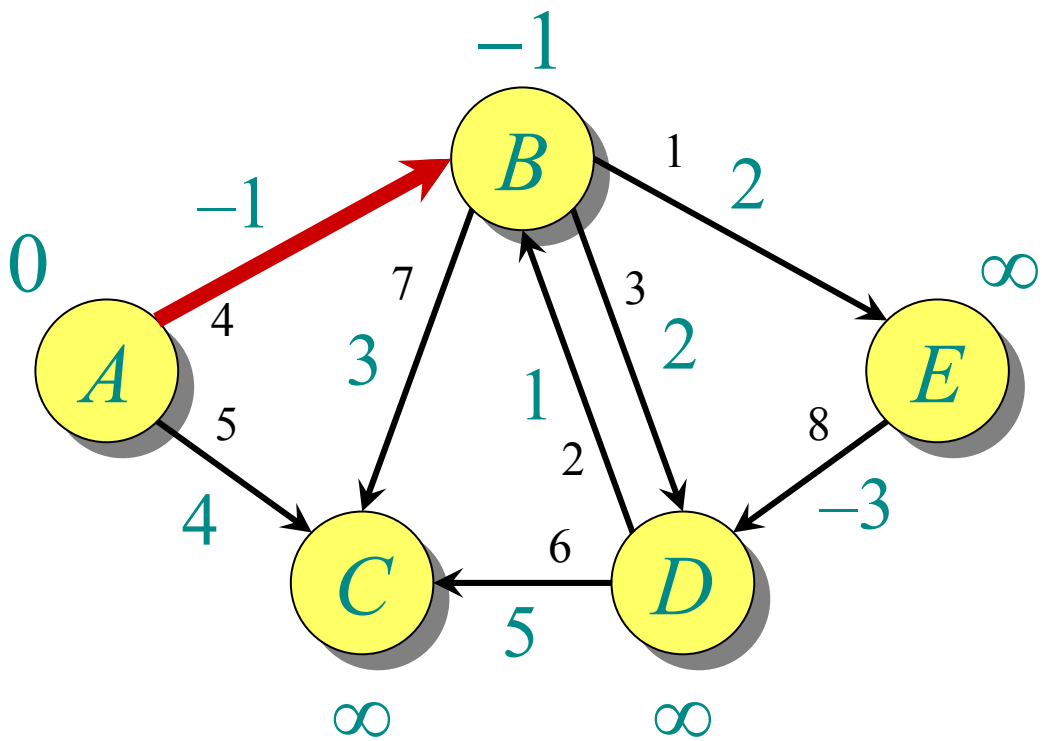


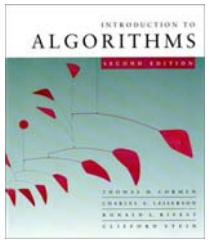
Bellman-Ford örneği



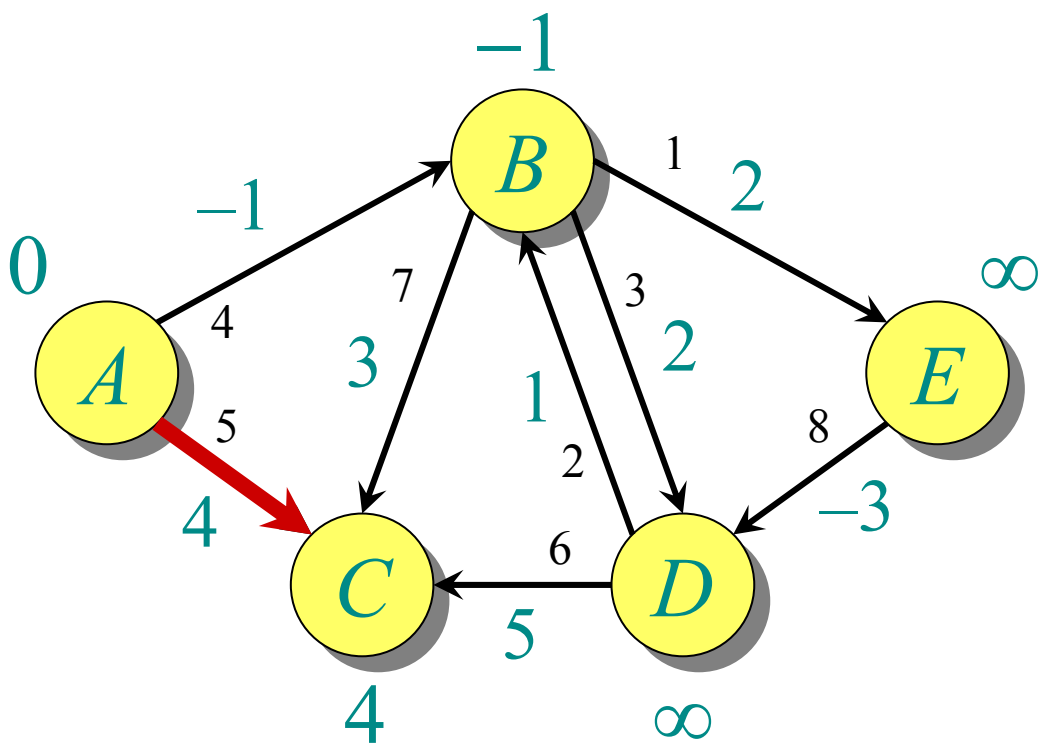


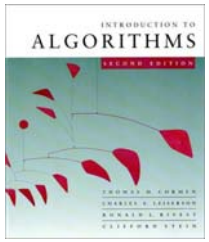
Bellman-Ford örneği



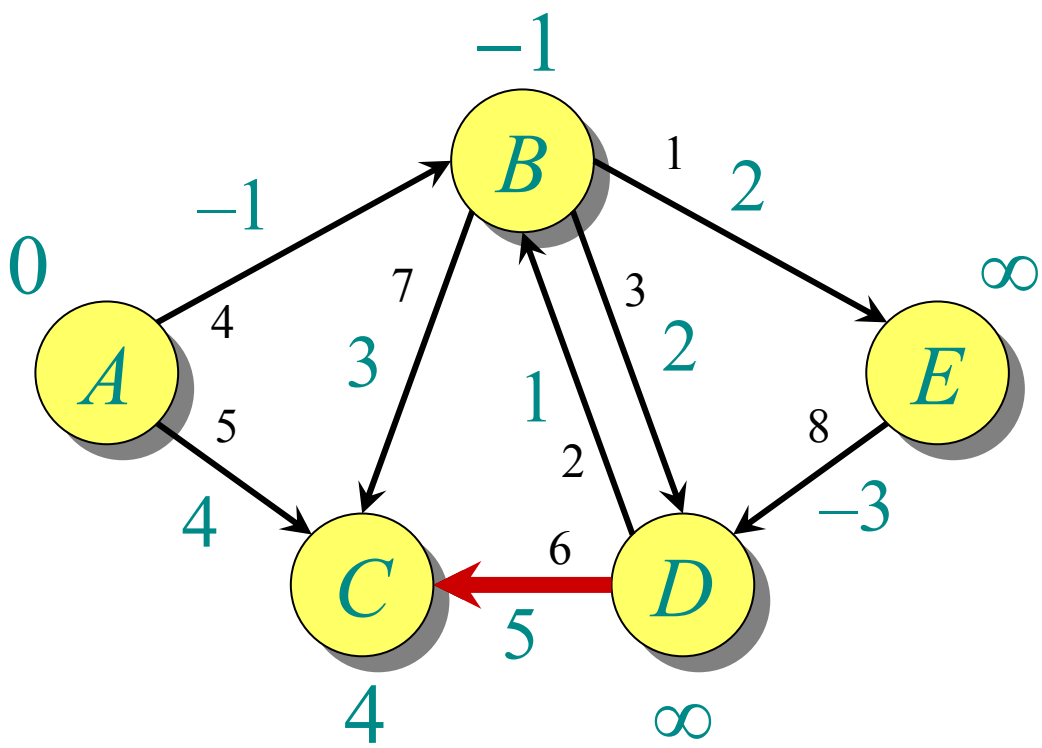


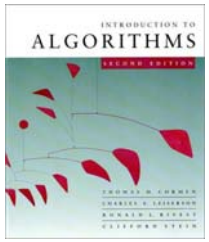
Bellman-Ford örneği



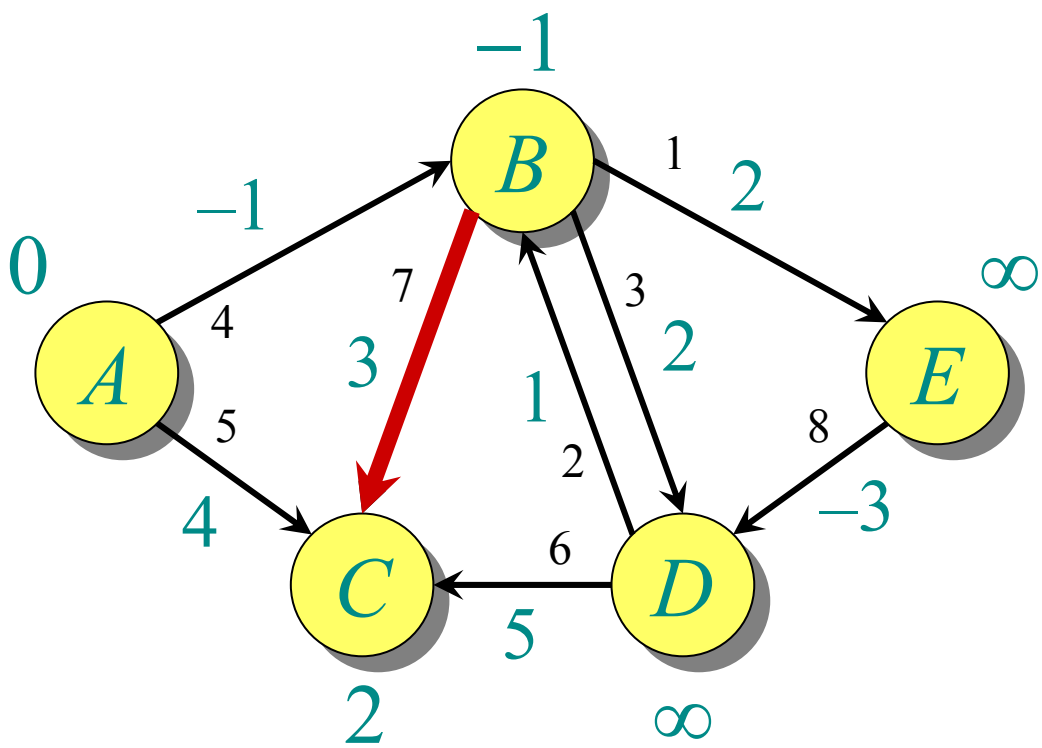


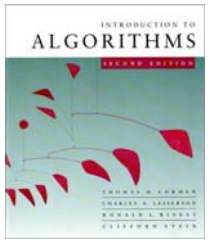
Bellman-Ford örneği



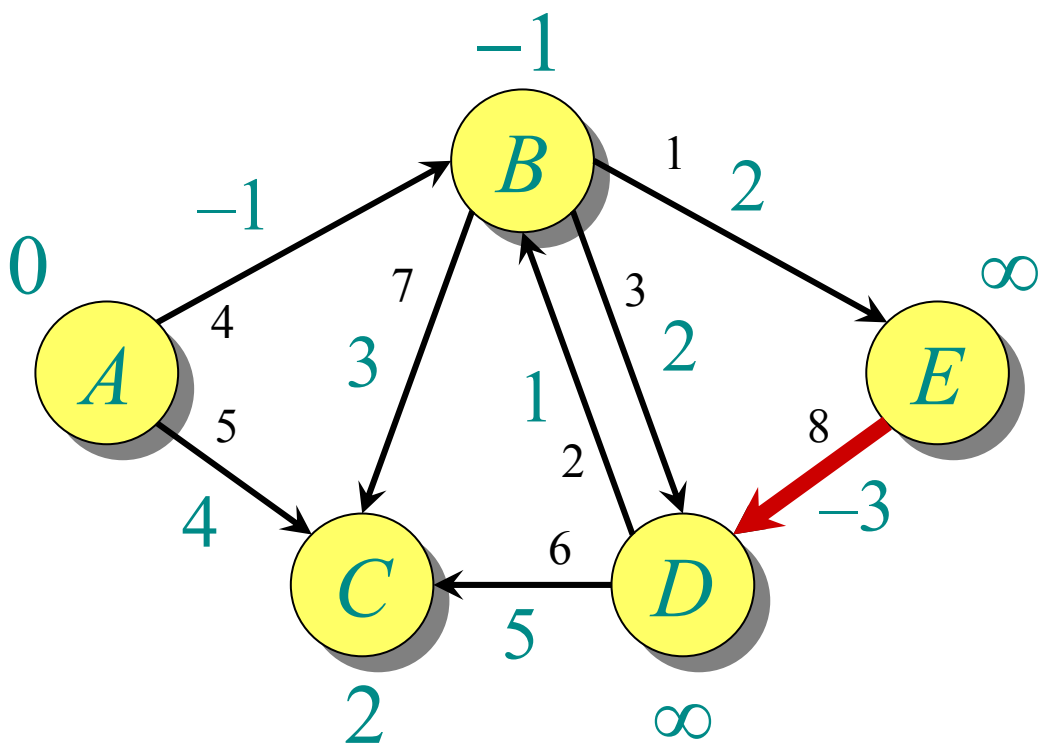


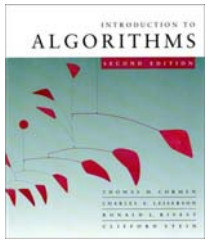
Bellman-Ford örneği



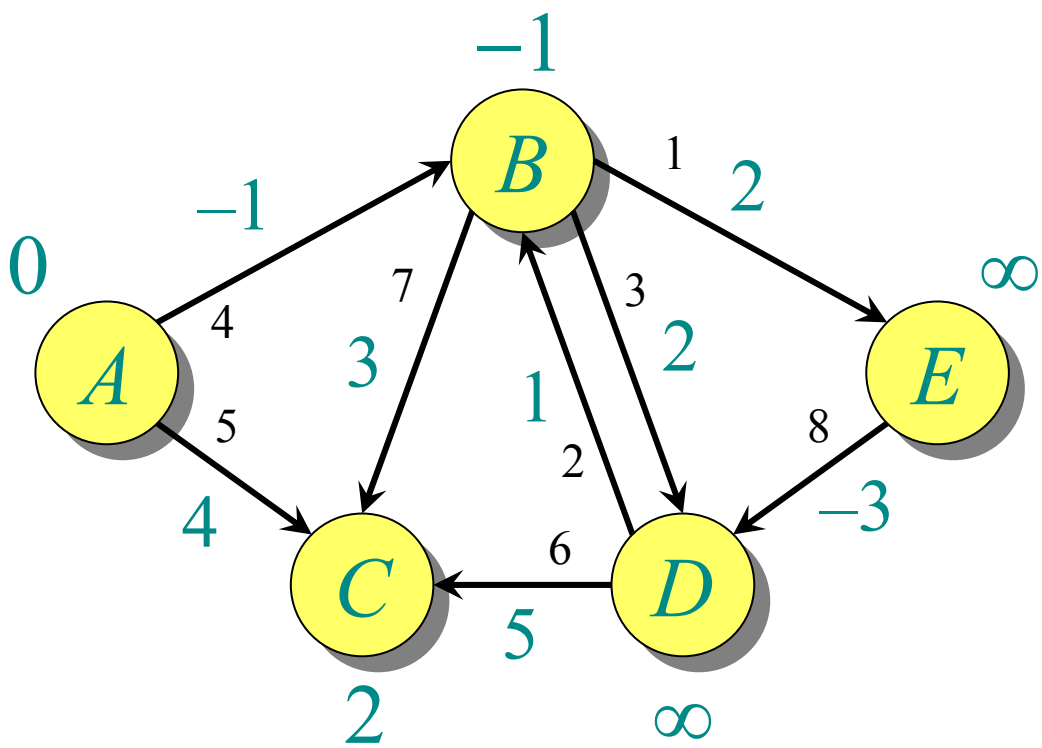


Bellman-Ford örneği

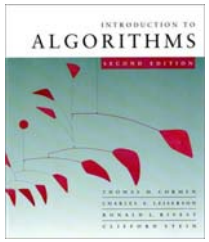




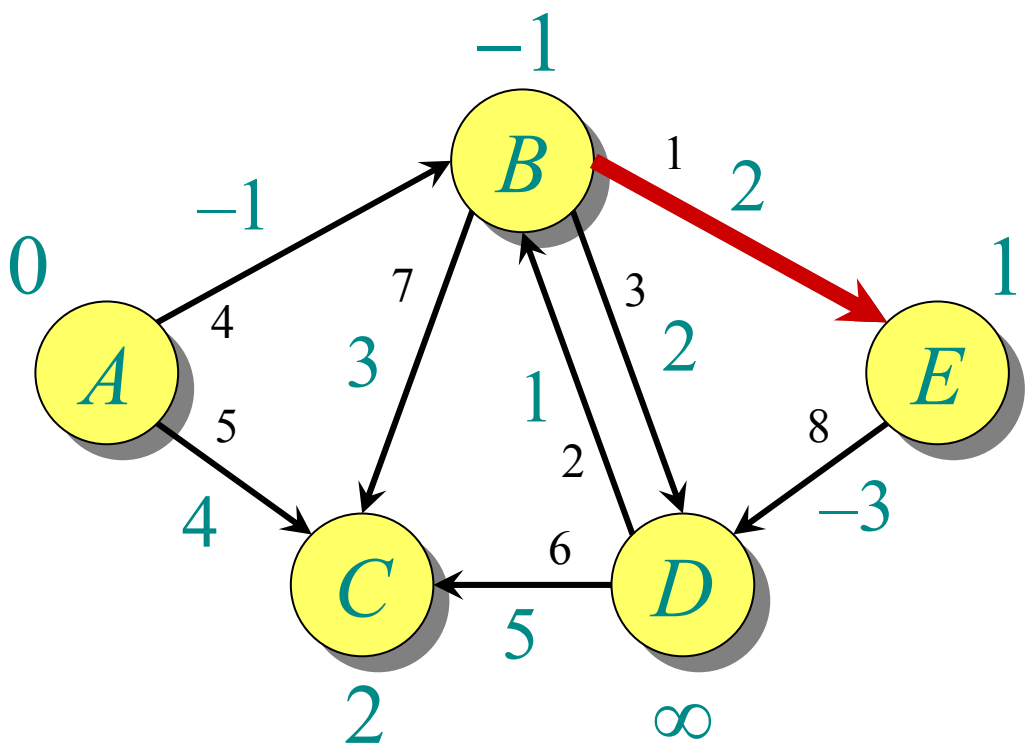
Bellman-Ford örneği

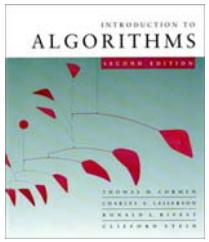


1. geçişin sonunda

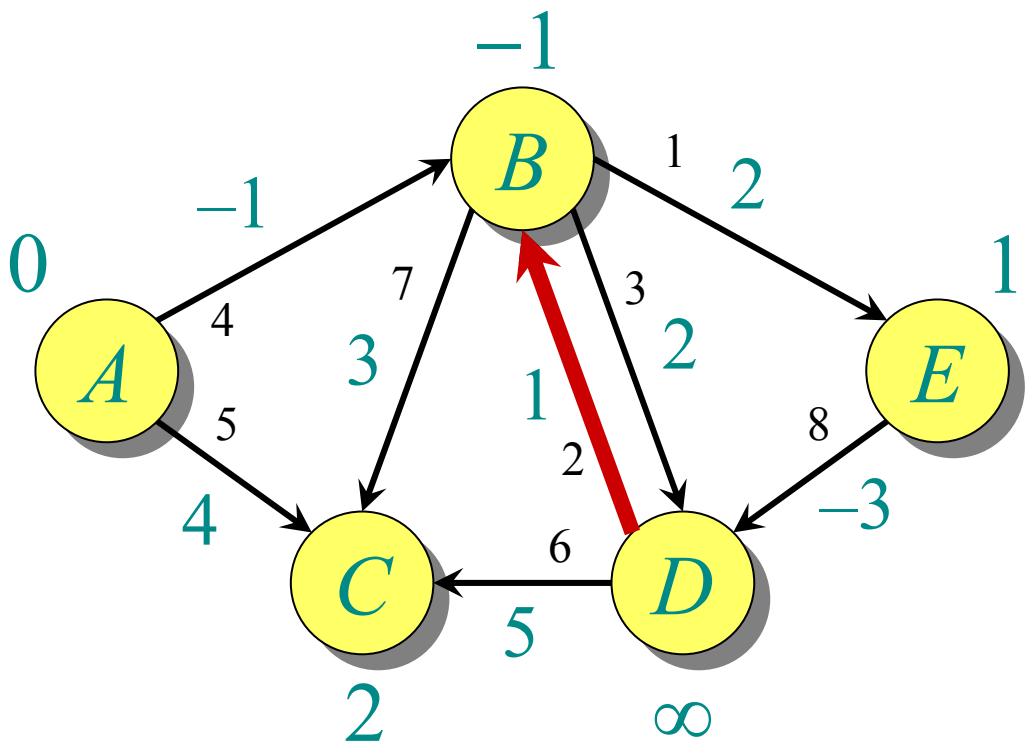


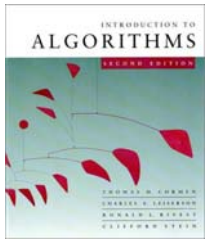
Bellman-Ford örneği



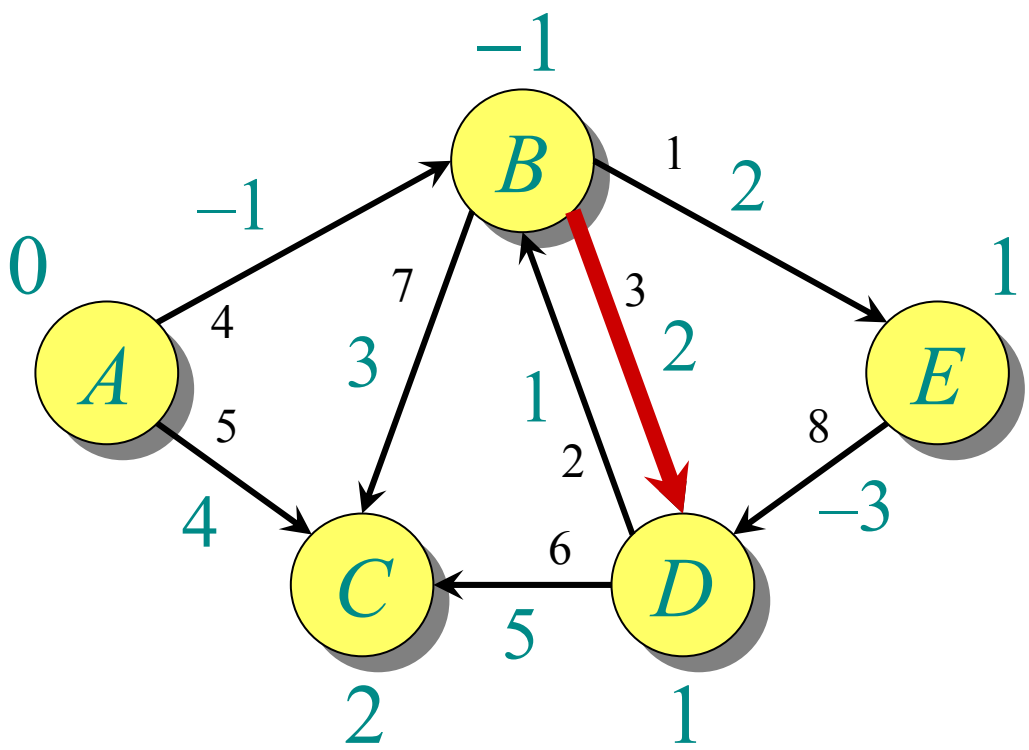


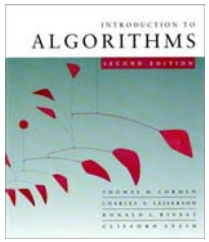
Bellman-Ford örneği



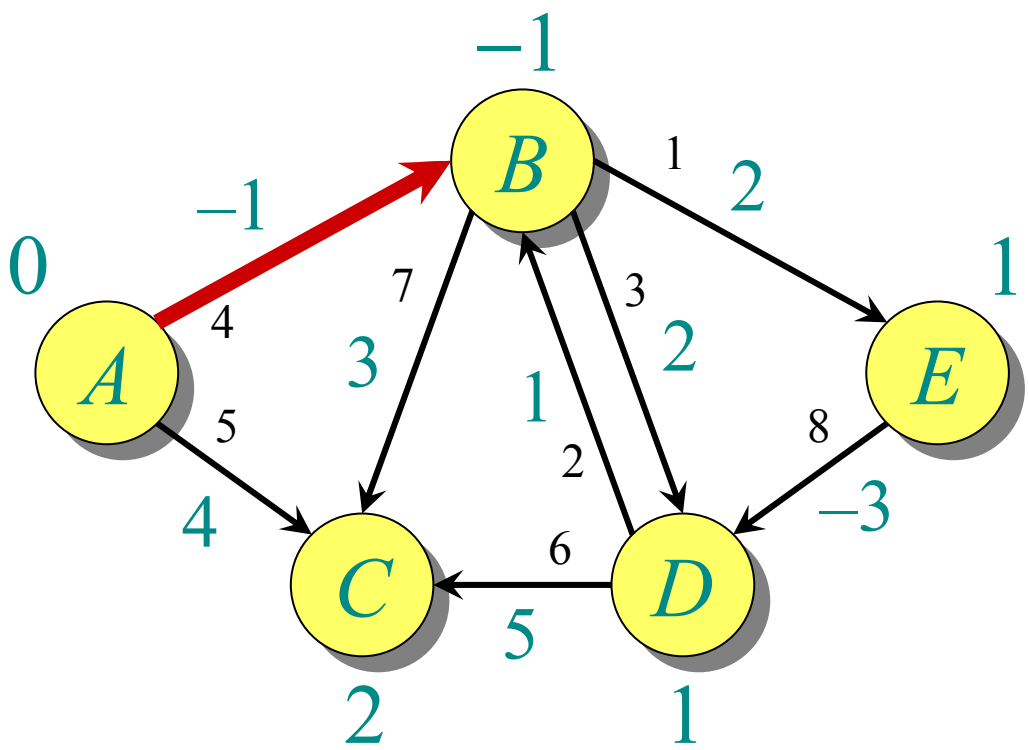


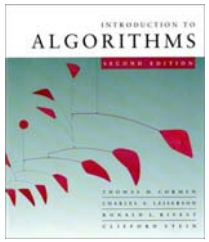
Bellman-Ford örneği



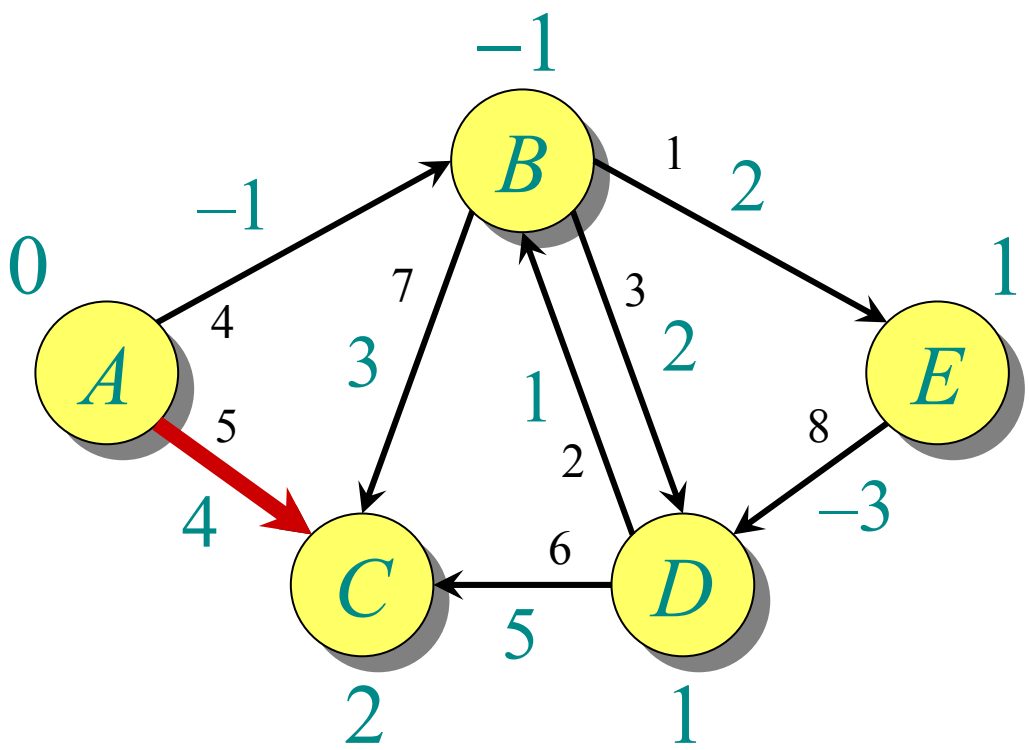


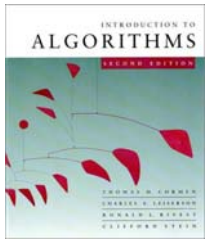
Bellman-Ford örneği



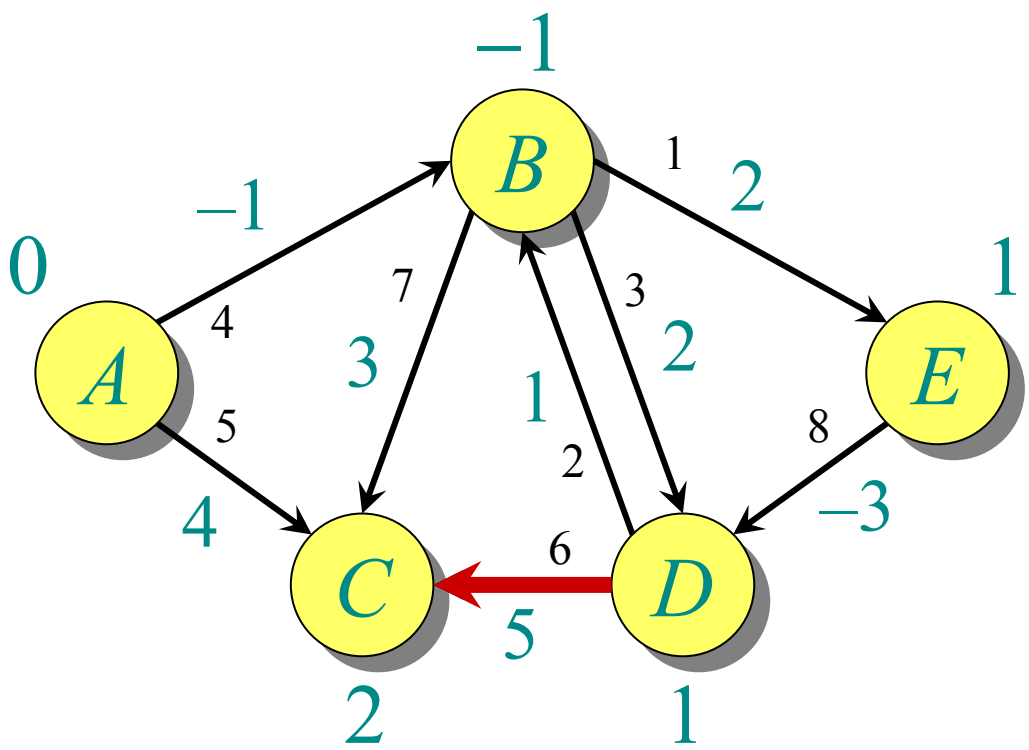


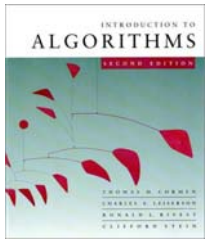
Bellman-Ford örneği



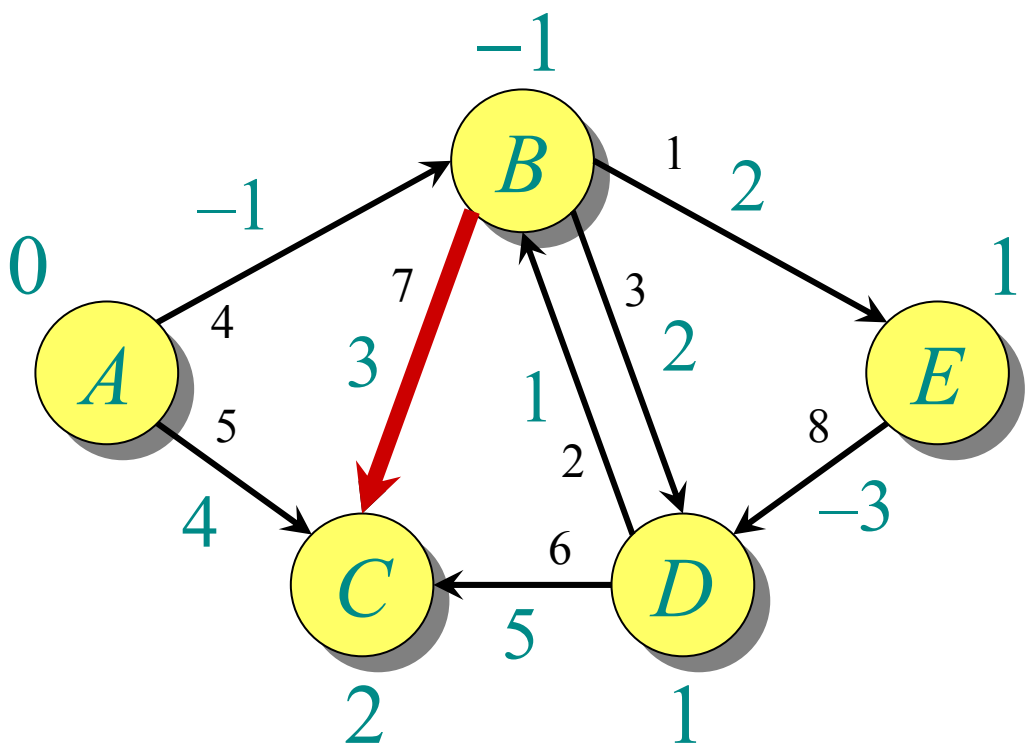


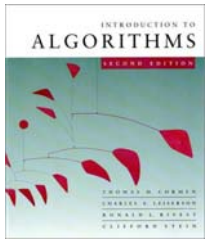
Bellman-Ford örneği



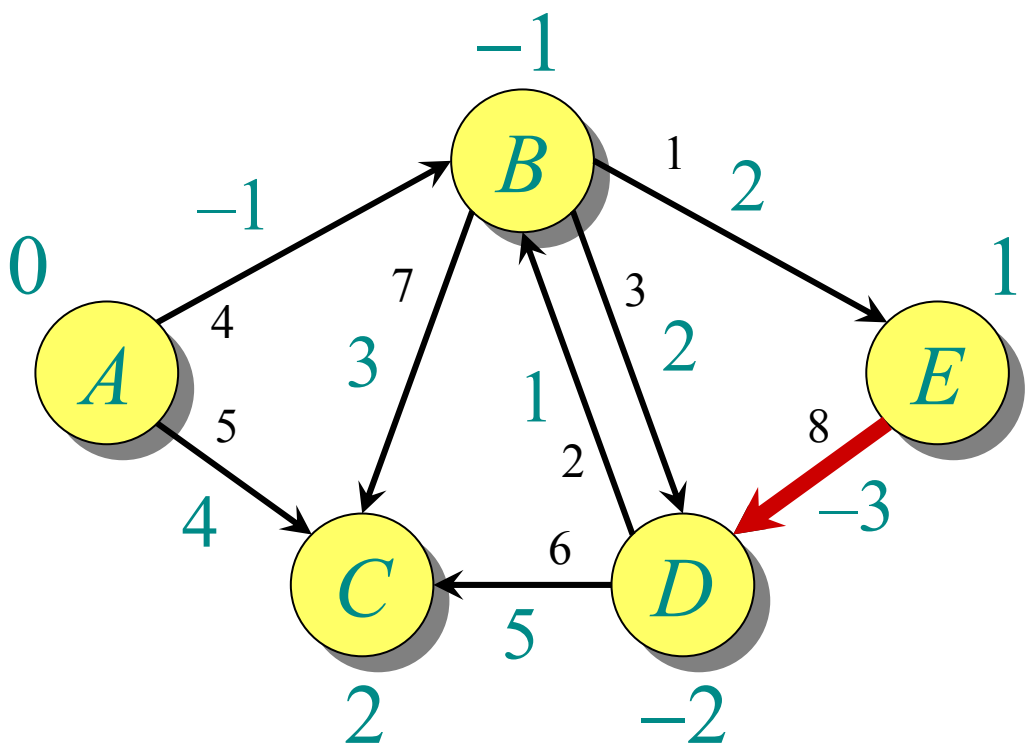


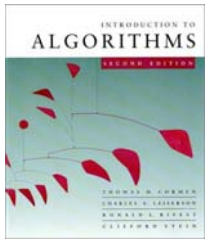
Bellman-Ford örneği



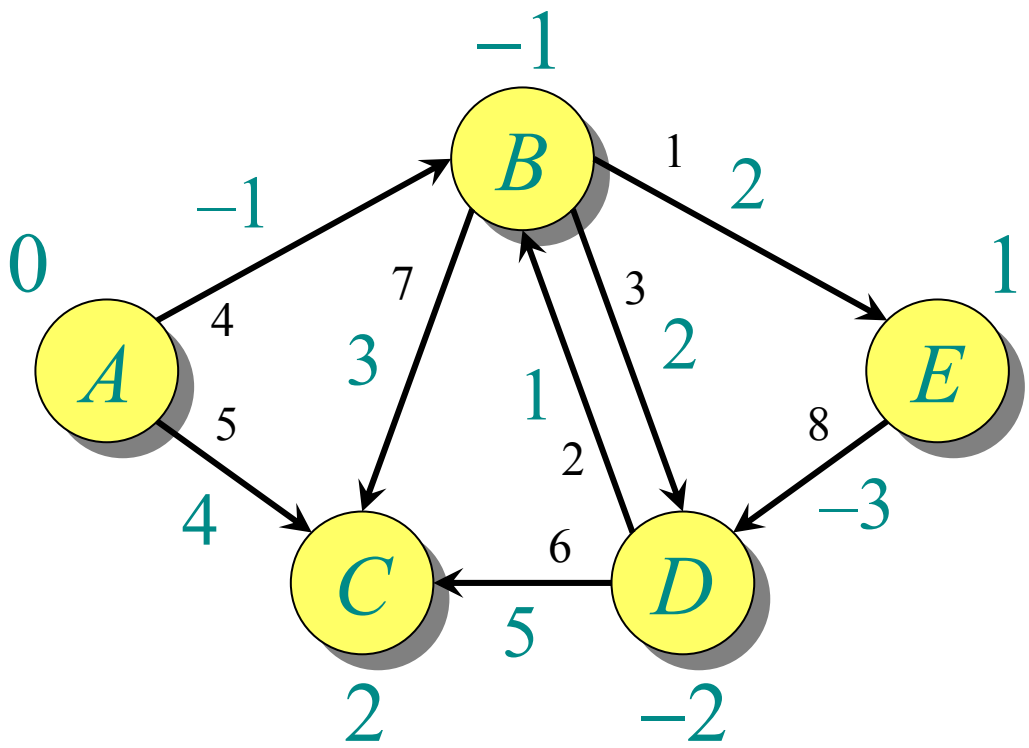


Bellman-Ford örneği

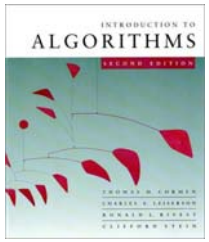




Bellman-Ford örneği

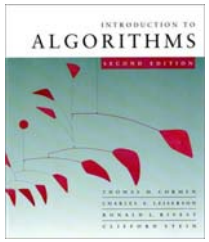


2. geçişin sonu (ve 3 ve 4).



Doğruluk

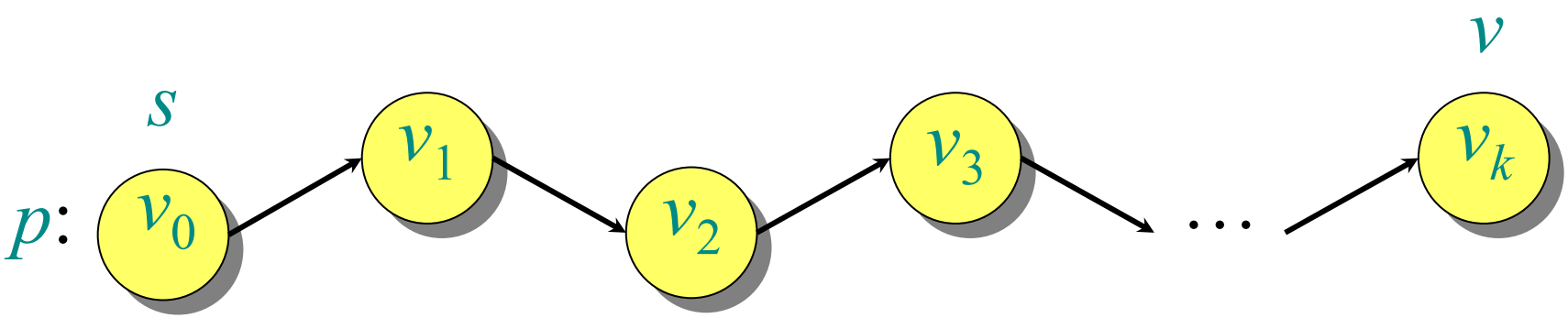
Teorem. Eğer $G = (V, E)$ hiç negatif ağırlık çevrimi içermiyorsa, sonrasında Bellman-Ford algoritması, bütün $v \in V'$ ler için $d[v] = \delta(s, v)'$ yi çalıştırır.



Doğruluk

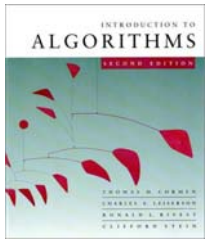
Teorem. Eğer $G = (V, E)$ hiç negatif ağırlık çevrimi içermiyorsa, sonrasında Bellman-Ford algoritması bütün $v \in V$ ler için $Vd[v] = \delta(s, v)$' yi çalıştırır.

Kanıt. $v \in V$ herhangi bir köşe olsun ve s' den v' ye, üzerinde en az sayıda köşe olan en kısa yolun p olduğunu farzedin.

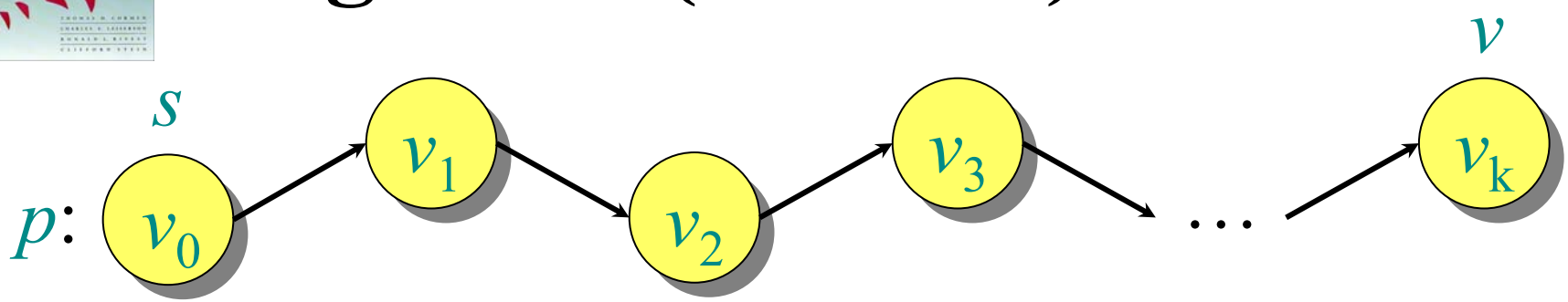


p en kısa yol ise,

$$\delta(s, v_i) = \delta(s, v_{i-1}) + w(v_{i-1}, v_i).$$



Doğruluk (Devamı)



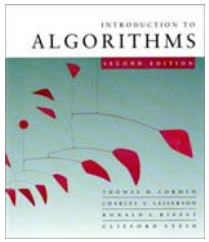
İlk olarak, $d[v_0] = 0 = \delta(s, v_0)$ ve $d[v_0]$sonraki gevşetmeler tarafından değiştirilmemiş.
(Ders 14' teki $d[v] \geq \delta(s, v)$ kuramı sebebiyle).

- E' den 1 geçiş sonra, $d[v_1] = \delta(s, v_1)$.
- E' den 2 geçiş sonra, $d[v_2] = \delta(s, v_2)$.

  ⋮

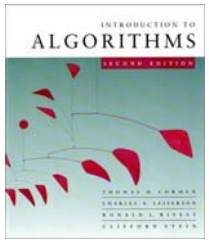
- E' den k geçiş sonra, $d[v_k] = \delta(s, v_k)$.

Eğer G negatif ağırlık çevrimi içermiyorsa, p basittir.
En uzun basit yolun $\leq |V| - 1$ kadar kenarı vardır. □



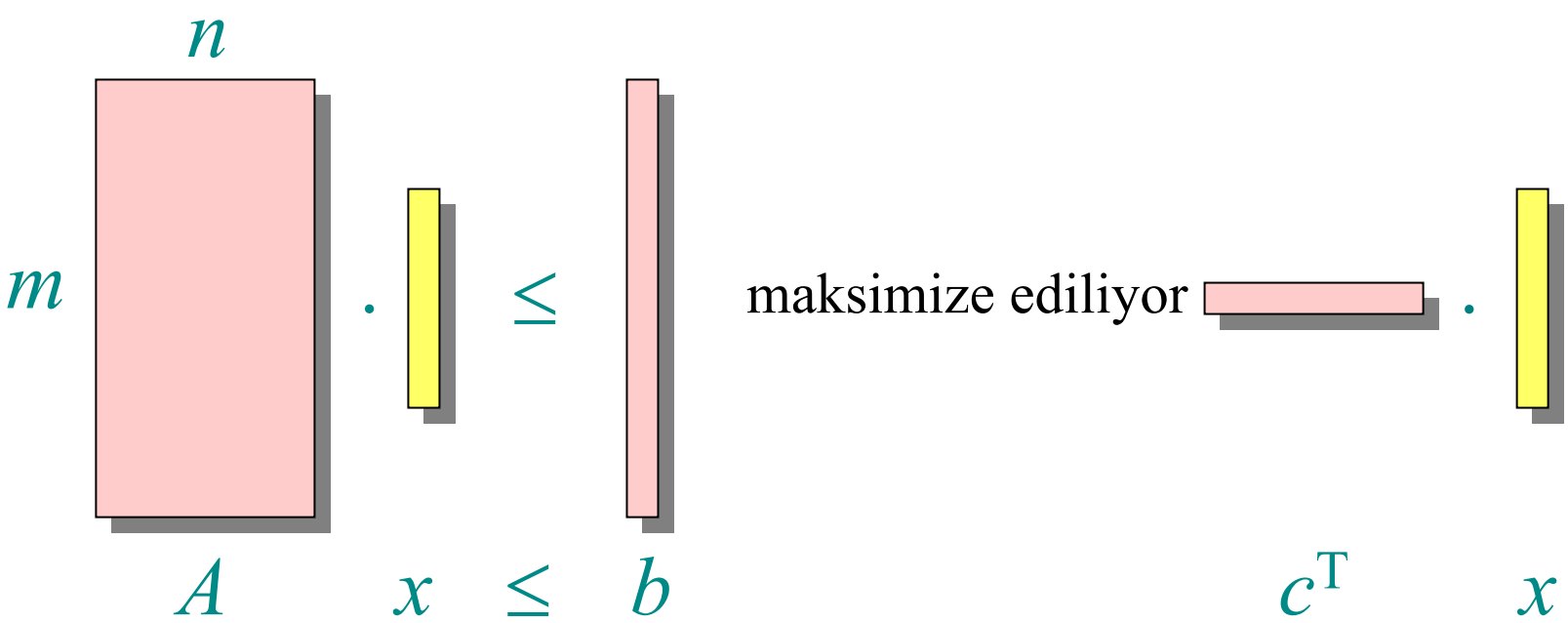
Negatif ağırlık çevrimlerini bulma

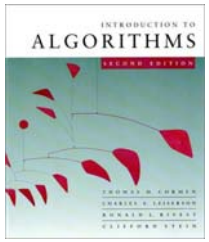
Doğal Sonuç. $|V|-1$ geçiş sonra $d[v]$ değeri birleşmede başarısız olursa, G' de s' den erişilebilir bir negatif ağırlık çevrimi vardır. □



Doğrusal programlama

A, $m \times n$ bir matris olsun, b, m-vektörü ve c n-vektörü olsun. $c^{\mathrm{T}}x$ öznesini $Ax \leq b$ ' ye maksimize eden n-vektör x' i bulun ya da böyle bir çözümün bulunmadığını belirleyin.

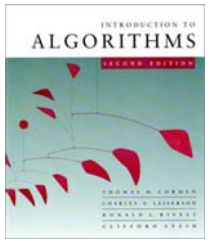




Doğrusal programlama algoritmaları

Genel problem için algoritmalar

- Tek yönlü (Simplex) yöntemler —Pratik, ama en kötü koşma süresi, üstel zamanlı.
- Dahili-nokta yöntemi — polinomsal zamanlı ve tek yönlü (simplex) yöntemle yarışır.



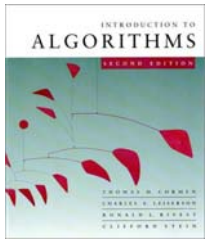
Doğrusal programlama algoritmaları

Genel problem için algoritmalar

- Tek yönlü (Simplex) yöntemler —Pratik, ama en kötü koşma süresi, üstel zamanlı.
- Dahili-nokta yöntemi — polinomsal zamanlı ve tek yönlü (simplex) yöntemle yarışır.

Fizibilite problemi: Optimizasyon kriteri yok.
$Ax \leq b$ için bir x bulun.

- Genellikle, alışılmış LP (doğrusal programlama) kadar zordur.

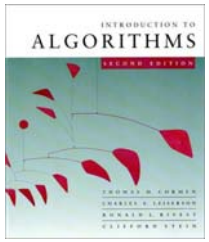


Fark kısıtlarının sisteminin çözümlemesi

Her satırı sadece bir tane 1, bir tane -1 içeren ve kalanı 0 olan A için doğrusal programlama.

Örnek:

$$\left.\begin{aligned} x_1 - x_2 &\le 3 \\ x_2 - x_3 &\le -2 \\ x_1 - x_3 &\le 2 \end{aligned}\right\} \quad x_j - x_i \le w_{ij}$$



Fark kısıtları sisteminin çözümlemesi

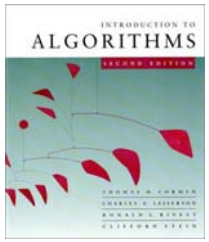
Her satırı sadece bir tane 1, bir tane -1 içeren ve kalanı 0 olan A için doğrusal programlama.

Örnek:

$$\left.\begin{aligned} x_1 - x_2 &\le 3 \\ x_2 - x_3 &\le -2 \\ x_1 - x_3 &\le 2 \end{aligned}\right\} \quad x_j - x_i \le w_{ij}$$

Çözüm:

$$\begin{aligned} x_1 &= 3 \\ x_2 &= 0 \\ x_3 &= 2 \end{aligned}$$



Fark kısıtları sisteminin çözümlemesi

Her satırı sadece bir tane 1, bir tane -1 içeren ve kalanı 0 olan A için doğrusal programlama.

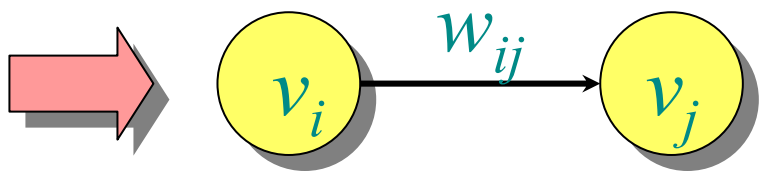
Örnek:

$$\left.\begin{aligned} x_1 - x_2 &\le 3 \\ x_2 - x_3 &\le -2 \\ x_1 - x_3 &\le 2 \end{aligned}\right\} \quad x_j - x_i \le w_{ij}$$

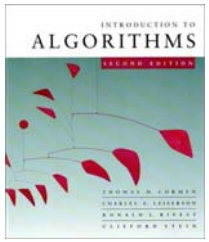
Çözüm:

$$\begin{aligned} x_1 &= 3 \\ x_2 &= 0 \\ x_3 &= 2 \end{aligned}$$

Kısıt grafiği:

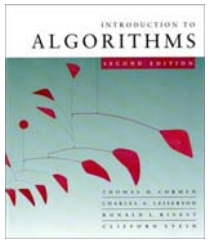
$x_j - x_i \le w_{ij}$ 

("A" matrisinin boyutları: $|E| \times |V|$.)



Karşılanamaz kısıtlar

Teorem. Eğer kısıt grafiği bir negatif ağırlık çevrimi içeriyorsa, farklar sistemi karşılanamaz.

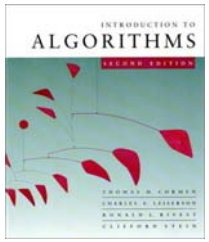


Karşılanamaz kısıtlar

Teorem. Eğer kısıt grafiği bir negatif ağırlık çevrimi içeriyorsa, farklar sistemi karşılanamaz.

Kanıt. Negatif ağırlık çevrimi şöyle olsun: $v_1 \to v_2 \to \cdots \to v_k \to v_1$. sonrasında

$$\begin{aligned} x_2 - x_1 &\le w_{12} \\ x_3 - x_2 &\le w_{23} \\ &\vdots \\ x_k - x_{k-1} &\le w_{k-1,\,k} \\ x_1 - x_k &\le w_{k1} \end{aligned}$$



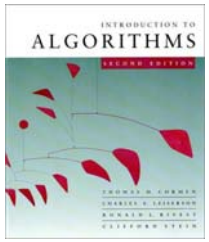
Kısıtların karşılanması

Teorem. Eğer kısıt grafiği bir negatif ağırlık çevrimi içeriyorsa, farklar sistemi karşılanamaz.

Kanıt. Negatif ağırlık çevrimi şöyle olsun: $v_1 \to v_2 \to \cdots \to v_k \to v_1$. sonrasında

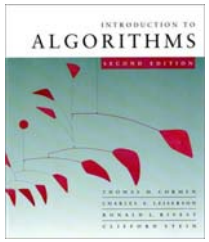
$$
\begin{aligned}
x_2 - x_1 &\le w_{12} \\
x_3 - x_2 &\le w_{23} \\
&\vdots \\
x_k - x_{k-1} &\le w_{k-1,\,k} \\
x_1 - x_k &\le w_{k1} \\
\hline
0 &\le \text{çevrimin ağırlığı} \\
&< 0
\end{aligned}
$$

bu nedenle x_i için kısıtları karşılayabilecek bir değer yoktur. □



Kısıtların karşılanması

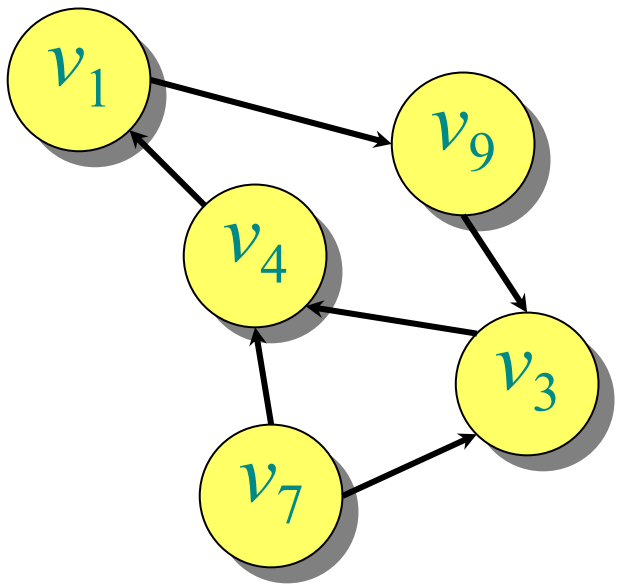
Teorem. Kısıt grafiğinde negatif ağırlık çevrimi bulunmadığını farzedin. O zaman kısıtlar karşılanabilir.

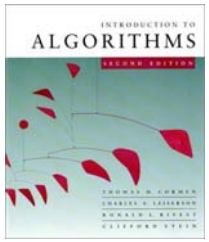


Kısıtların karşılanması

Teorem. Kısıt grafiğinde negatif ağırlık çevrimi bulunmadığını farzedin. O zaman kısıtlar karşılanabilir.

Kanıt. V' ye yeni bir s köşesi ekleyin ve bu $v_i \in V$ 'deki her köşeye 0- kenar ağırlığı ile gelsin.

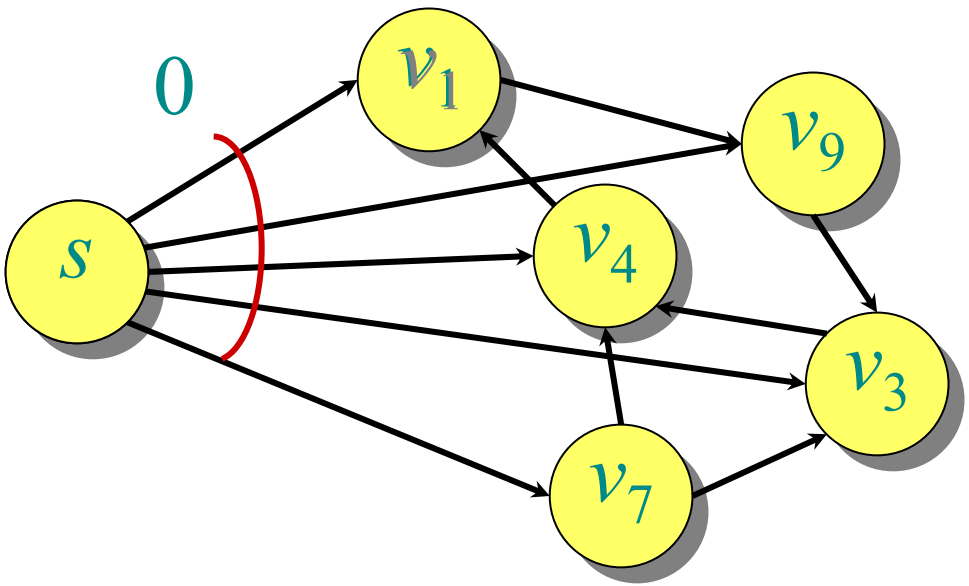




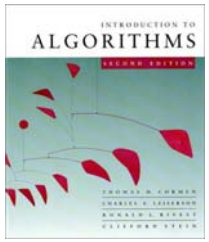
Kısıtların karşılanması

Teorem. Kısıt grafiğinde negatif ağırlık çevrimi bulunmadığını farzedin. O zaman kısıtlar karşılanabilir.

Kanıt. V' ye yeni bir s köşesi ekleyin ve bu $v_i \in V$ 'deki her köşeye 0- kenar ağırlığı ile gelsin.



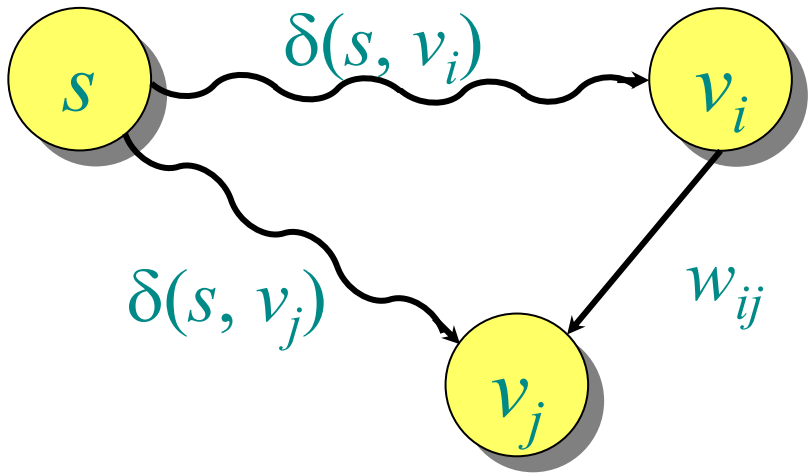
Not: Negatif ağırlık çevrimi yok, yani en kısa yollar var.



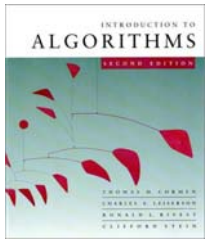
Kanıt (Devamı)

İddia: $x_i = \delta(s, v_i)$ ataması kısıtları karşılar.

Bir $x_j - x_i \le w_{ij}$ kısıtı olduğunu ve
s den v_j ve v_i ye en kısa yollar olduğunu düşünün.



Üçgen eşitsizliği bize $\delta(s,v_j) \le \delta(s, v_i) + w_{ij}$ ' yi verir.
$x_i = \delta(s, v_i)$ ve $x_j = \delta(s, v_j)$ olduğunda, kısıt $x_j - x_i \le w_{ij}$ karşılanır. □



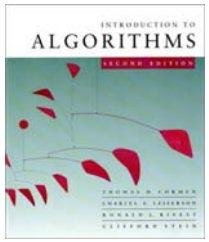
Bellman-Ford ve Doğrusal programlama

Doğal Sonuç. Bellman-Ford algoritması m fark kısıtının olduğu bir sistemi n değişkenli olarak $O(mn)$ süresinde çözebilir. □

Tek kaynaklı en kısa yollar basit bir LP problemidir.

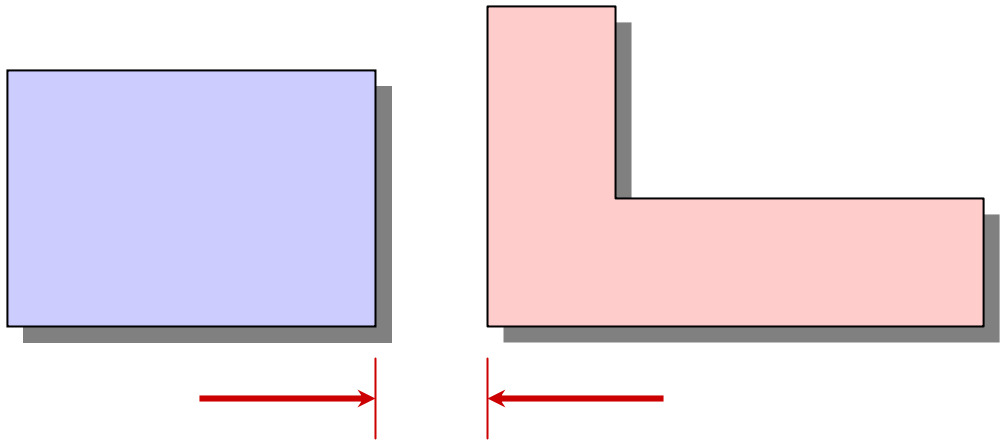
Aslında, Bellman-Ford $x_1 + x_2 + \cdots + x_n$ öğesini, $x_j - x_i \le w_{ij}$ ve $x_i \le 0$ kısıt koşullarında maksimize eder (egzersiz).

Bellman-Ford aynı zamanda $\max_i\{x_i\} - \min_i\{x_i\}$' yi minimize eder (egzersiz).



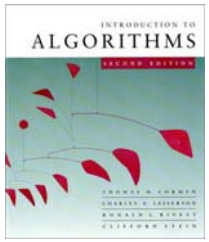
VLSI yerleşimi küçültme uygulaması

Entegre devre özellikleri:

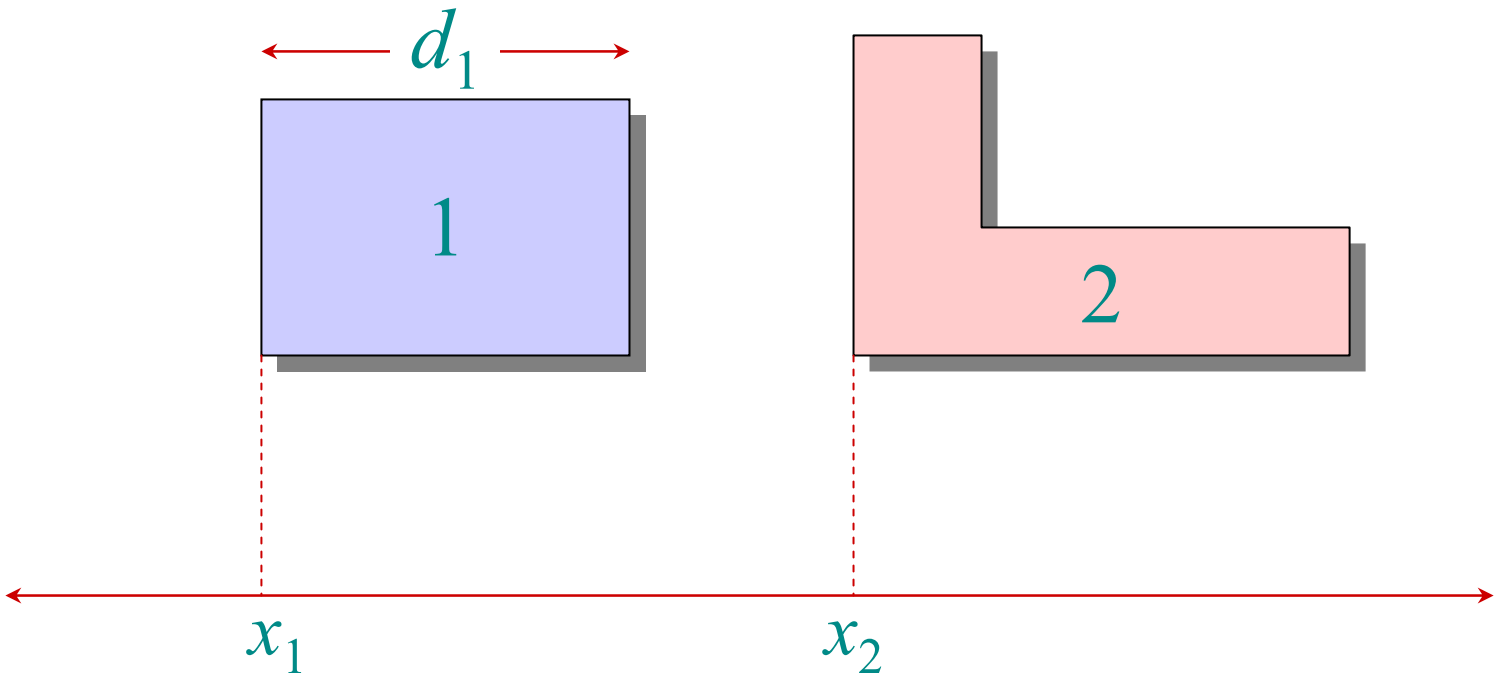


minimum ayrılma λ

Problem: VLSI yerleşiminde, tüm nitelikleri çok yakınlaştırmadan, nitelikler arasındaki mesafeyi tek boyutta sıkıştırın.



VLSI Yerleşimi küçültmesi



Kısıt: $x_2 - x_1 \geq d_1 + \lambda$

Bellman-Ford $\max_i\{x_i\} - \min_i\{x_i\}$, x-boyutunda yerleşimi en aza indirir.