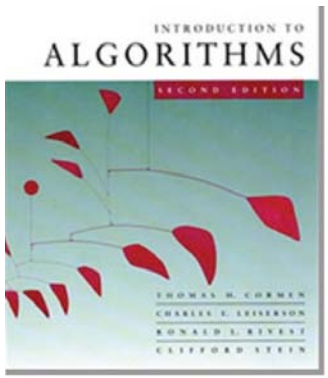
Algoritmalara Giriş

6.046J/18.401J

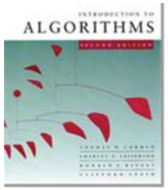


DERS 14

Yarışmacı Çözümleme

- Kendini Düzenleyen Listeler
- Öne Taşıma - Buluşsal Yaklaşım
- Öne Taşımanın Yarışmacı Çözümlemesi

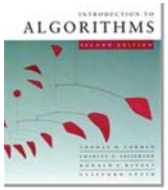
Prof. Charles E. Leiserson



Kendini Düzenleyen Listeler

n elemanlı L listesi

- Erişim(x) işlemi $\text{seviye}_L(x)$ = x'in L'nin başına olan uzaklığına mal olur.
- L listesi, komşu elemanların 1 maliyetle yer değiştirmesi ile yeniden düzenlenebilir.



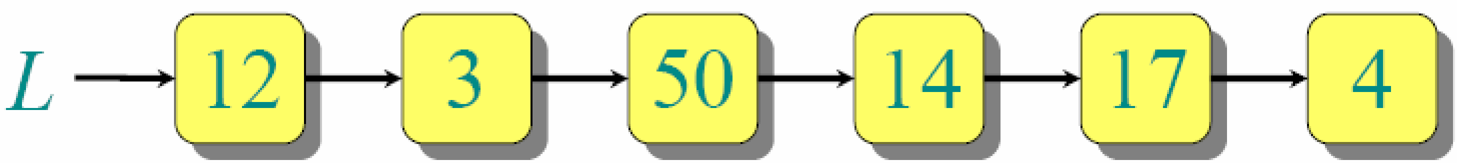
Kendini Düzenleyen Listeler

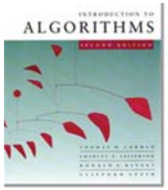
n elemanlı *L* listesi

- Erişim(x) işlemi $\text{seviye}_L(x)$ = x'in L'nin başına olan uzaklığına mal olur.

- L listesi, komşu elemanların 1 maliyetle yer değiştirmesi ile yeniden düzenlenebilir.

Örnek :





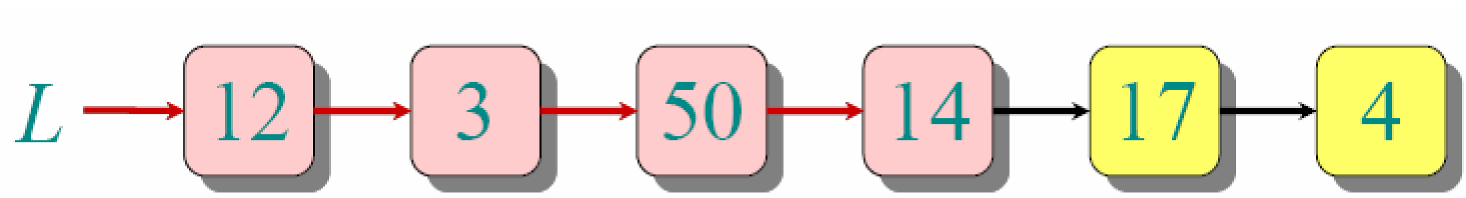
Kendini Düzenleyen Listeler

n elemanlı *L* listesi

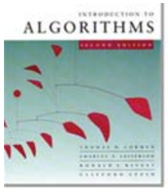
• Erişim(x) işlemi $\text{seviye}_L(x)$ = x'in *L*'nin başına olan uzaklığına mal olur.

• *L* listesi, komşu elemanların 1 maliyetle yer değiştirmesi ile yeniden düzenlenebilir.

Örnek :



14 anahtarlı elemana erişim 4'e mal olur.



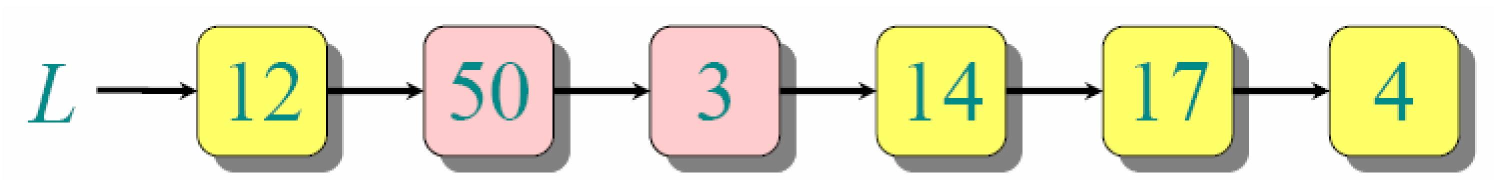
Kendini Düzenleyen Listeler

n elemanlı *L* listesi

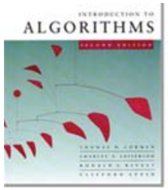
• Erişim(x) işlemi seviye$_L$(x) = x'in L'nin başına olan uzaklığına mal olur.

• L listesi, komşu elemanların 1 maliyetle yer değiştirmesi ile yeniden düzenlenebilir.

Örnek :



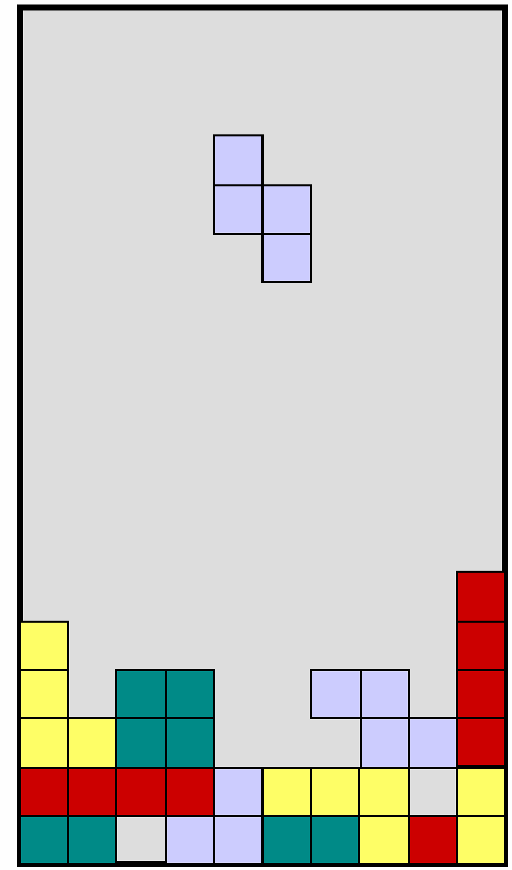
3 ile 50'nin yer değiştirmesi 1'e mal olur.



Çevrimiçi ve Çevrimdışı Problemler

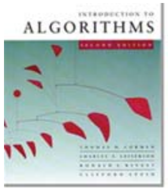
Tanım. İşlemler serisi *S* her seferinde bir kez oluşturulur. Her işlem için, gelecekle ilgili herhangi bir bilgi olmaksızın bir ***çevrimiçi*** algoritma *A*, işlemi çalıştırır. (örn. *Tetris*)

Çevrimdışı bir algoritma bütün *S* serisini en baştan görür.



Tetris Oyunu

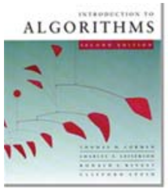
Amaç : Toplam maliyet $C_A(S)$'yi en aza indirgemektir.



Kendini Düzenleyen Listelerin En Kötü Durum Analizi

Rakip her zaman, L listesinin kuyruğundaki n'inci elemana erişir. Herhangi bir çevrimiçi A algoritması için en kötü durum :

$$C_A(S) = \Omega(|S| \cdot n)$$



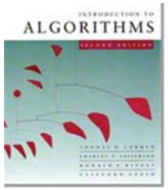
Kendini Düzenleyen Listelerin Ortalama Durum Analizi

x elemanının $p(x)$ ihtimali ile erişildiğini varsayalım. Bu durumda;

$$\mathrm{E}[C_A(S)] = \sum_{x \in L} p(x) \cdot \textit{\textbf{seviye}}_L\textbf{\textit{(x)}}$$

Bu, L listesi p'ye göre azalan şekilde sıralı olursa en aza indirgenir.

Buluş : L listesindeki her elemanın kaç kere erişildiğini gösteren bir liste tutalım, L'yi bu listeye göre sıralayalım.



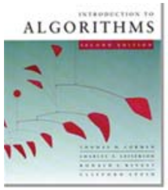
Öne Taşıma Buluşu

Pratik : Geliştiriciler, ***Öne Taşıma (ÖT)*** Buluşunun iyi sonuçlar verdiğini tecrübe ettiler.

Fikir : *x*'e eriştikten sonra, *x*'i listenin en başına sıra değiştirmelerle taşımak.

$$\text{maliyet} = 2 \cdot \text{seviye}_L(x)$$

ÖT Buluşu, *S* dizisindeki erişim yerelliğine iyi tepki verir.



Yarışmacı Çözümleme

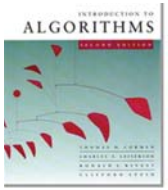
Tanım : Eğer bir *k* sabiti, herhangi bir *S* işlemler dizisi için;

$$C_A(S) \leq \alpha \cdot C_{\mathrm{OPT}}(S) + k$$

durumunu sağlıyorsa, çevrimiçi *A* algoritması

α – *yarışmacıdır*.

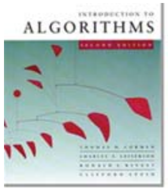
Burada *OPT* en uygun çevrimdışı algoritmadır. ("Tanrının algoritması")



Öne Taşıma O(1) yarışmacıdır

Teorem : ÖT, kendini düzenleyen listeler için

4-yarışmacıdır.



Öne Taşıma O(1) yarışmacıdır

Teorem : ÖT, kendini düzenleyen listeler için

4-yarışmacıdır.

İspat. L_i, i'inci erişimden sonraki ÖT listesi olsun, L_i* ise i'inci erişimden sonra en uygun liste olsun.

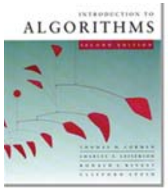
c_i = ÖT'deki i'inci işlem için maliyet

= $2 \cdot \text{seviye}_{L_{i-1}}(x)$, eğer x'e erişiliyorsa.

c_i^* = ÖT'deki i'inci işlem için maliyet

= $\text{seviye}_{L_{i-1}*}(x) + t_i$

t_i en uygun sıra değiştirme sayısıdır.



Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

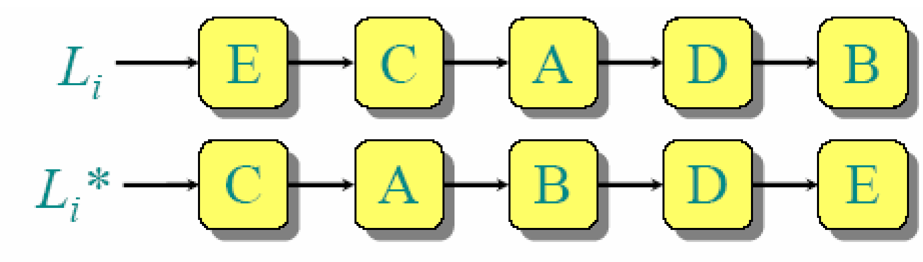
$$\begin{aligned}\Phi(L_i) &= 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i{}^*} x\}| \\ &= 2 \cdot \# \text{ Ters çevirme}\end{aligned}$$

Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\Phi(L_i) = 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i^*} x\}|$$
$$= 2 \cdot \# \text{ Ters çevirme}$$

Örnek :

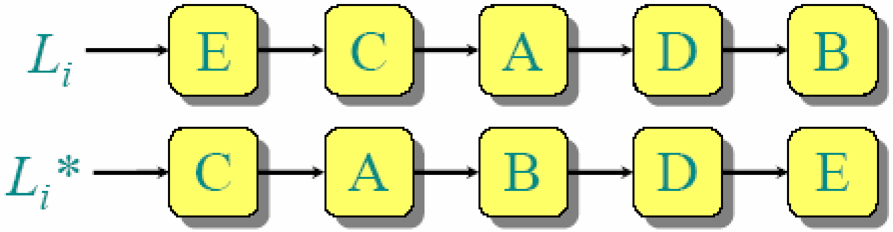


Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \rightarrow \mathbb{R}$

$$\begin{aligned}\Phi(L_i) &= 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i^*} x\}| \\ &= 2 \cdot \# \text{ Ters çevirme}\end{aligned}$$

Örnek :



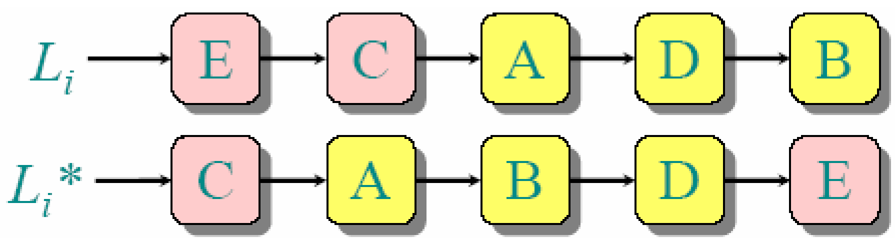
$$\Phi(L_i) = 2 \cdot |\{\ldots\}|$$

Potansiyel Fonksiyonu

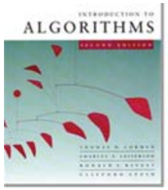
Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\Phi(L_i) = 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i{}^*} x\}|$$
$$= 2 \cdot \# \text{ Ters çevirme}$$

Örnek :



$$\Phi(L_i) = 2 \cdot |\{(\mathrm{E},\mathrm{C}), \ldots\}|$$



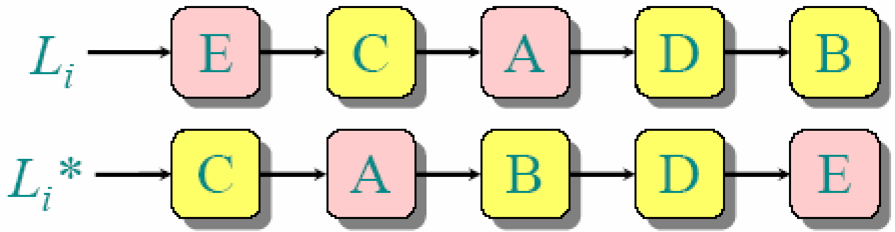
Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\Phi(L_i) = 2 \cdot |\{(x, y) : x \prec_{L_i} y \ \text{ ve } \ y \prec_{L_i^*} x\}|$$

$$= 2 \cdot \# \ \text{Ters çevirme}$$

Örnek :



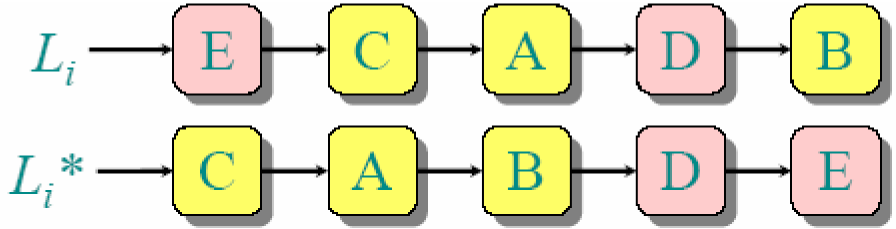
$$\Phi(L_i) = 2 \cdot |\{(E,C), (E,A), \ldots\}|$$

Potansiyel Fonksiyonu

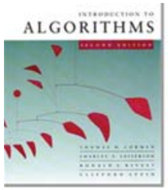
Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \rightarrow \mathbb{R}$

$$\begin{aligned}\Phi(L_i) &= 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i^*} x\}| \\ &= 2 \cdot \# \text{ Ters çevirme}\end{aligned}$$

Örnek :



$$\Phi(L_i) = 2 \cdot |\{(E,C), (E,A), (E,D), \ldots\}|$$

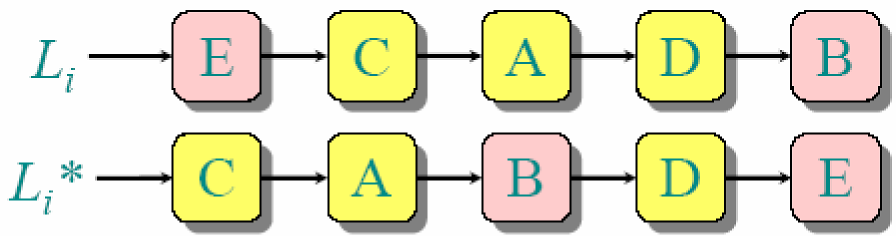


Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\Phi(L_i) = 2 \cdot |\{(x, y) : x \prec_{L_i} y \ \text{ ve } \ y \prec_{L_i{}^*} x\}|$$
$$= 2 \cdot \# \ \text{Ters çevirme}$$

Örnek :



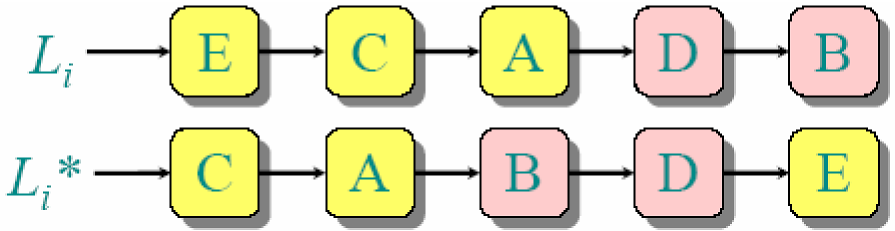
$$\Phi(L_i) = 2 \cdot |\{(E,C), (E,A), (E,D), (E,B), \ldots\}|$$

Potansiyel Fonksiyonu

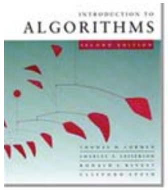
Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \rightarrow \mathbb{R}$

$$\Phi(L_i) = 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i{*}} x\}|$$
$$= 2 \cdot \# \text{ Ters çevirme}$$

Örnek :



$$\Phi(L_i) = 2 \cdot |\{(E,C), (E,A), (E,D), (E,B), (D,B)\}|$$

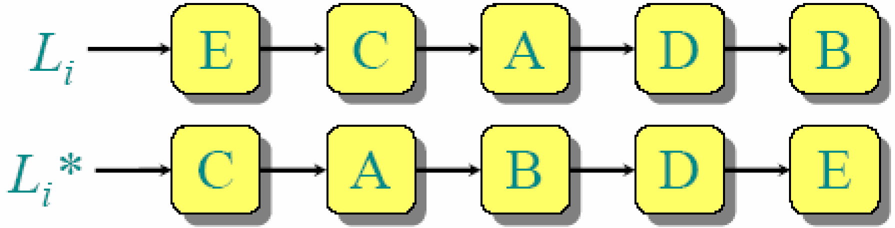


Potansiyel Fonksiyonu

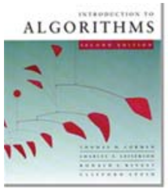
Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\begin{aligned}\Phi(L_i) &= 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i^*} x\}| \\ &= 2 \cdot \# \text{ Ters çevirme}\end{aligned}$$

Örnek :



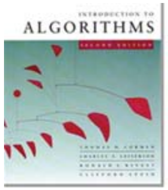
$$\begin{aligned}\Phi(L_i) &= 2 \cdot |\{(E,C), (E,A), (E,D), (E,B), (D,B)\}| \\ &= 10 .\end{aligned}$$



Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\Phi(L_i) = 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i^*} x\}|$$
$$= 2 \cdot \# \text{ Ters çevirme}$$

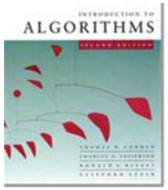


Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\begin{aligned}\Phi(L_i) &= 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i{}^*} x\}| \\ &= 2 \cdot \# \text{ Ters çevirme}\end{aligned}$$

- *i=0, 1, ...,* için $\Phi(L_i) \geq 0$
- Eğer ÖT ve en uygun algoritma aynı liste ile başlıyorsa; $\Phi(L_0) = 0$



Potansiyel Fonksiyonu

Potansiyel Fonksiyonu tanımlayın : $\Phi : \{L_i\} \to \mathbb{R}$

$$\Phi(L_i) = 2 \cdot |\{(x, y) : x \prec_{L_i} y \text{ ve } y \prec_{L_i^*} x\}|$$
$$= 2 \cdot \# \text{ Ters çevirme}$$

- *i=0, 1, ...,* için $\Phi(L_i) \geq 0$
- Eğer ÖT ve en uygun algoritma aynı liste ile başlıyorsa; $\Phi(L_0) = 0$

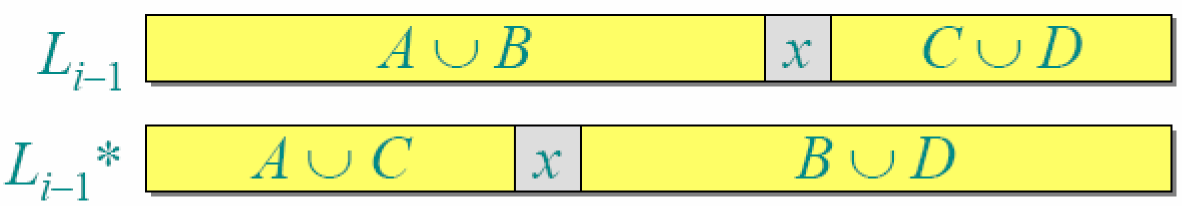
1 değişimle **Φ** ne kadar değişir?

- Bir değişim 1 ters çevirme yaratır veya yok eder.
- $\Delta\Phi = \pm 2$

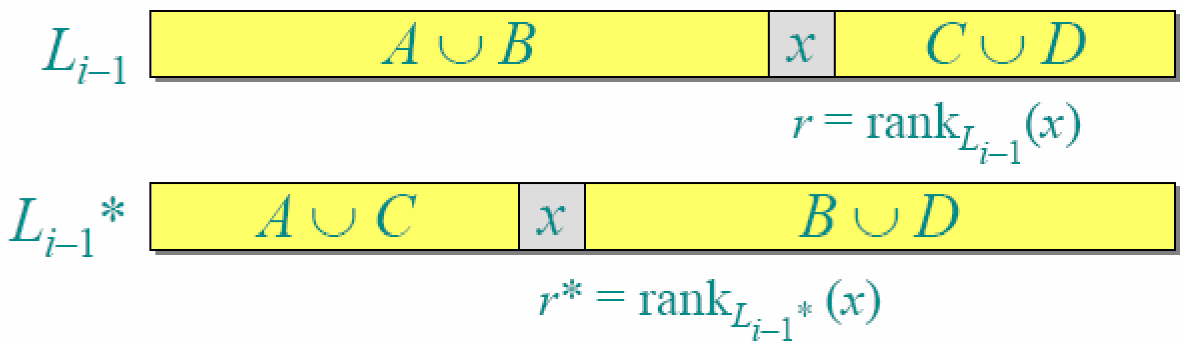
Erişim sırasında ne olur?

i'inci işlemin x'e eriştiğini düşünün ve şunu tanımlayın;

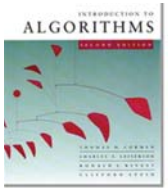
$$
\begin{aligned}
A &= \{y \in L_{i-1} : y \prec_{L_{i-1}} x \text{ ve } y \prec_{L_{i-1}^*} x\}, \\
B &= \{y \in L_{i-1} : y \prec_{L_{i-1}} x \text{ ve } y \succ_{L_{i-1}^*} x\}, \\
C &= \{y \in L_{i-1} : y \succ_{L_{i-1}} x \text{ ve } y \prec_{L_{i-1}^*} x\}, \\
D &= \{y \in L_{i-1} : y \succ_{L_{i-1}} x \text{ ve } y \succ_{L_{i-1}^*} x\}.
\end{aligned}
$$



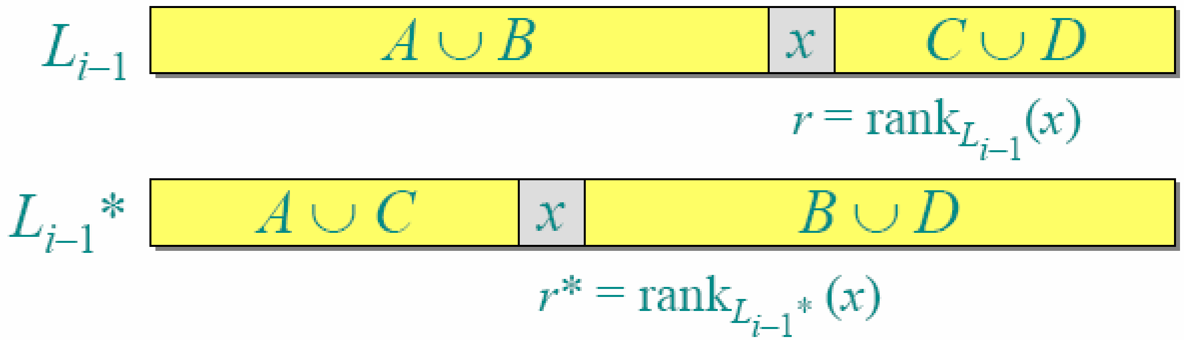
Erişim sırasında ne olur?



$$r = |A| + |B| + 1 \qquad r^* = |A| + |C| + 1 .$$



Erişim sırasında ne olur?



$$r = |A| + |B| + 1 \qquad r^* = |A| + |C| + 1.$$

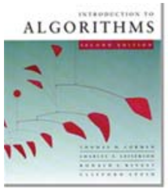
ÖT x’i öne hareket ettirdiğinde, $|A|$ ters çevirme yaratır ve $|B|$ ters çevirmeyi yok eder. En uygun algoritmada, her bir yer değiştirme ≤ 1 ters çevirme yaratır. Bu durumda;

$$\Phi(L_i) - \Phi(L_{i-1}) \leq 2(|A| - |B| + t_i)$$

Amortize edilmiş maliyet

ÖT'nin *i*'inci işlemi için Φ göre amortize edilmiş maliyet :

$$\hat{c}_i = c_i + \Phi(L_i) - \Phi(L_{i-1})$$



Amortize edilmiş maliyet

ÖT'nin i'inci işlemi için Φ göre amortize edilmiş maliyet :

$$\begin{aligned}\hat{c}_i &= c_i + \Phi(L_i) - \Phi(L_{i-1}) \\ &\le 2r + 2(|A| - |B| + t_i)\end{aligned}$$

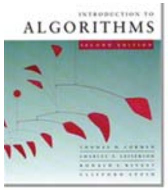
Amortize edilmiş maliyet

ÖT’nin *i*’inci işlemi için Φ göre amortize edilmiş maliyet :

$$\hat{c}_i = c_i + \Phi(L_i) - \Phi(L_{i-1})$$
$$\leq 2r + 2(|A| - |B| + t_i)$$

$$= 2r + 2(|A| - (r - 1 - |A|) + t_i)$$

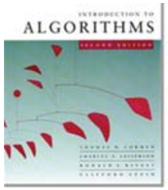
$r = |A| + |B| + 1$ iken.



Amortize edilmiş maliyet

ÖT'nin i'inci işlemi için Φ göre amortize edilmiş maliyet :

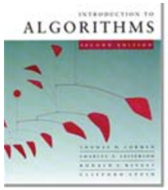
$$
\begin{aligned}
\hat{c}_i &= c_i + \Phi(L_i) - \Phi(L_{i-1}) \\
&\le 2r + 2(|A| - |B| + t_i) \\
&= 2r + 2(|A| - (r - 1 - |A|) + t_i) \\
&= 2r + 4|A| - 2r + 2 + 2t_i
\end{aligned}
$$



Amortize edilmiş maliyet

ÖT'nin i'inci işlemi için Φ göre amortize edilmiş maliyet :

$$
\begin{aligned}
\hat{c}_i &= c_i + \Phi(L_i) - \Phi(L_{i-1}) \\
&\le 2r + 2(|A| - |B| + t_i) \\
&= 2r + 2(|A| - (r - 1 - |A|) + t_i) \\
&= 2r + 4|A| - 2r + 2 + 2t_i \\
&= 4|A| + 2 + 2t_i
\end{aligned}
$$

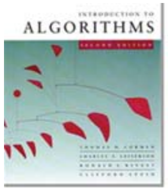


Amortize edilmiş maliyet

ÖT’nin i’inci işlemi için Φ göre amortize edilmiş maliyet :

$$\begin{aligned}
\hat{c}_i &= c_i + \Phi(L_i) - \Phi(L_{i-1}) \\
&\le 2r + 2(|A| - |B| + t_i) \\
&= 2r + 2(|A| - (r - 1 - |A|) + t_i) \\
&= 2r + 4|A| - 2r + 2 + 2t_i \\
&= 4|A| + 2 + 2t_i \\
&\le 4(r^* + t_i)
\end{aligned}$$

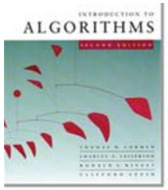
$r^* = |A| + |C| + 1 \ge |A| + 1$ iken..



Amortize edilmiş maliyet

ÖT'nin i'inci işlemi için Φ göre amortize edilmiş maliyet :

$$
\begin{aligned}
\hat{c}_i &= c_i + \Phi(L_i) - \Phi(L_{i-1}) \\
&\le 2r + 2(|A| - |B| + t_i) \\
&= 2r + 2(|A| - (r - 1 - |A|) + t_i) \\
&= 2r + 4|A| - 2r + 2 + 2t_i \\
&= 4|A| + 2 + 2t_i \\
&\le 4(r^* + t_i) \\
&= 4c_i^*.
\end{aligned}
$$



Büyük Final

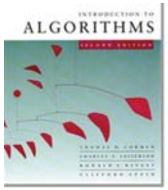
Bu durumda, şuna sahibiz;

$$C_{\mathrm{MTF}}(S) = \sum_{i=1}^{|S|} c_i$$

Büyük Final

Bu durumda, şuna sahibiz;

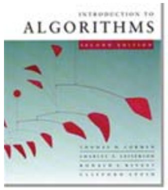
$$
\begin{aligned}
C_{\mathrm{MTF}}(S) &= \sum_{i=1}^{|S|} c_i \\
&= \sum_{i=1}^{|S|} \left(\hat{c}_i + \Phi(L_{i-1}) - \Phi(L_i) \right)
\end{aligned}
$$



Büyük Final

Bu durumda, şuna sahibiz;

$$
\begin{aligned}
C_{\mathrm{MTF}}(S) &= \sum_{i=1}^{|S|} c_i \\
&= \sum_{i=1}^{|S|} \left(\hat{c}_i + \Phi(L_{i-1}) - \Phi(L_i) \right) \\
&\le \left(\sum_{i=1}^{|S|} 4c_i{}^* \right) + \Phi(L_0) - \Phi(L_{|S|})
\end{aligned}
$$



Büyük Final

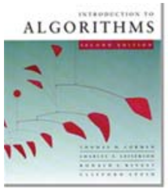
Bu durumda, şuna sahibiz;

$$
\begin{aligned}
C_{\mathrm{MTF}}(S) &= \sum_{i=1}^{|S|} c_i \\
&= \sum_{i=1}^{|S|} \left(\hat{c}_i + \Phi(L_{i-1}) - \Phi(L_i) \right) \\
&\le \left(\sum_{i=1}^{|S|} 4 c_i{}^* \right) + \Phi(L_0) - \Phi(L_{|S|}) \\
&\le 4 \cdot C_{\mathrm{OPT}}(S),
\end{aligned}
$$

$\Phi(L_0) = 0$ ve $\Phi(L_{|S|}) \ge 0$ iken.

EK

Eğer x'i öne taşıyan yer değiştirmeleri bedava kabul edersek, bu durumda öne taşıma 2-yarışmacı olur.



EK

Eğer x'i öne taşıyan yer değiştirmeleri bedava kabul edersek, bu durumda öne taşıma 2-yarışmacı olur.

Eğer $\boldsymbol{L}_0 \neq \boldsymbol{L}_0{}^*$ ise ne olur?

- Bu durumda, $\Phi(L_0)$ en kötü durumda $\Theta(n^2)$ olur.
- Ayrıca, $C_{\mathrm{MTF}}(S) \leq 4$ olur. n^2, $|S| \to \infty$ bir sabitse, $C_{\mathrm{OPT}}(S) + \Theta(n^2)$ ise hala 4-yarışmacıdır.