Hálózatok II
2005/2006

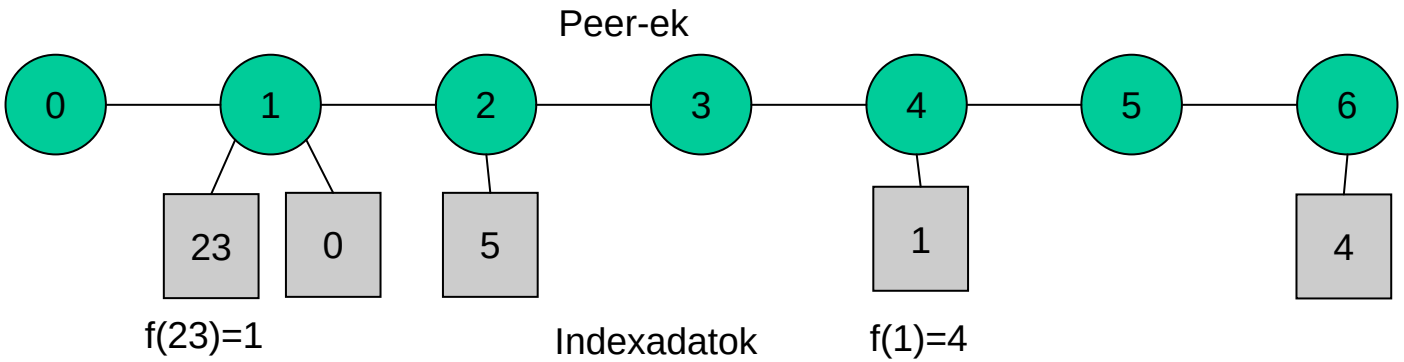
7: Peer-To-Peer Hálózatok II

P2P hálózatok kritériumai

- Kezelhetőség
  - Milyen nehéz a hálózatot működtetni
- Információ-koherencia
  - Mennyire jól osztja el az információt?
- Bővíthetőség
  - Milyen könnyen tud növekedni?
- Hibatűrés
  - Milyen könnyen hárítja el a hibákat?
- Biztonság
  - Mennyire nehezen rombolható szét tudatosan?
- Védelem politikai üldözéssel szemben
  - Milyen nehéz a hálózatot lekapcsolani?
- Skálázhatóság
  - Milyen nagyra tud a hálózat növekedni?

Egy hash-tábla mint Peer-to-Peer hálózat

- Minden Peer egy tárhelyet reprezentál 0,1,2,..,n-1
  - Egy minden Peers számára ismert hash-függvény, pl. n = 7 esetén
    - f(x) = (3x+1 mod 23) mod 7
  - A Peer-eket láncként kössük össze



- Keresés
  - Számítsuk ki f(x)-et
  - Menjünk oda ahhoz a Peer-hez a láncon, amely f(x)-et reprezentálja

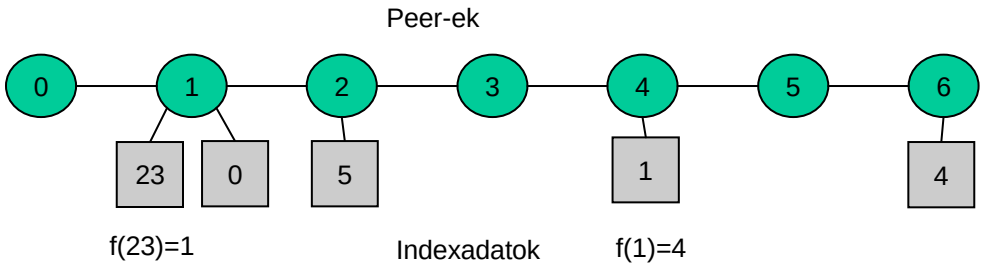
Hash-táblától az elosztott hash-táblához (DHT)

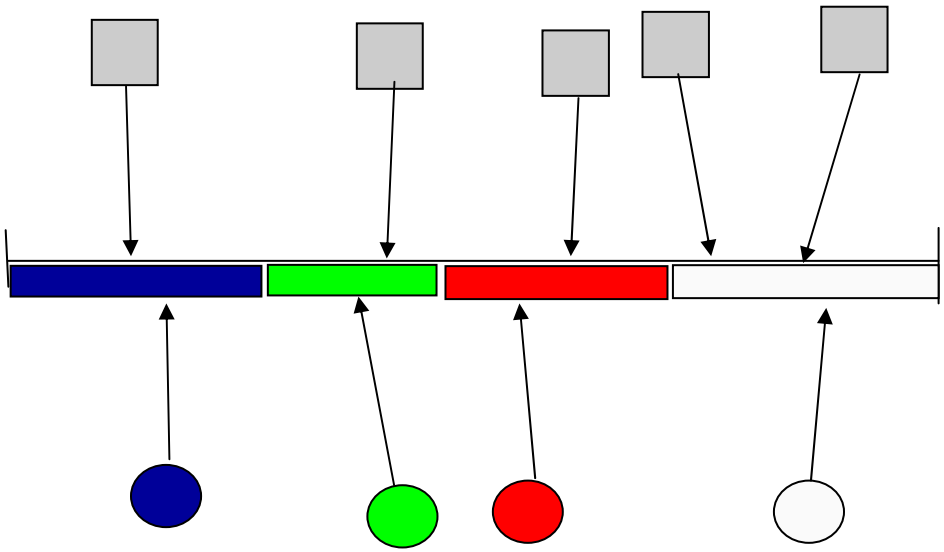
Hash-Táblák

- Előny
  - A keresés egyszerű
- Hátrány
  - Egy új Peer kapcsolódásakor új hash-függvényt kell választani
  - Hosszú utak, nagy élterhelés

Elosztott Hash-Tábla
(Distributed Hash Table, DHT)

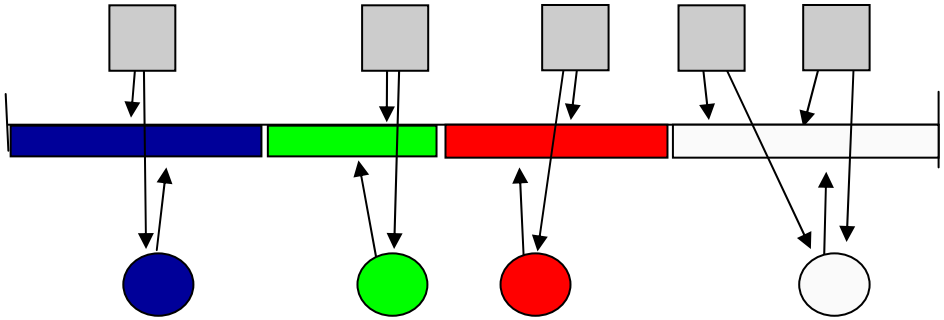
- A Peer-eket leképezzük (hashing) egy helyre és minden Peer-hez hozzárendeljük a hash-függvény értéktartományának egy részét
- Az adatokat is hash-eljük
  - A hash-függvény értéké alapján a tartományért felelős Peer-en tároljuk



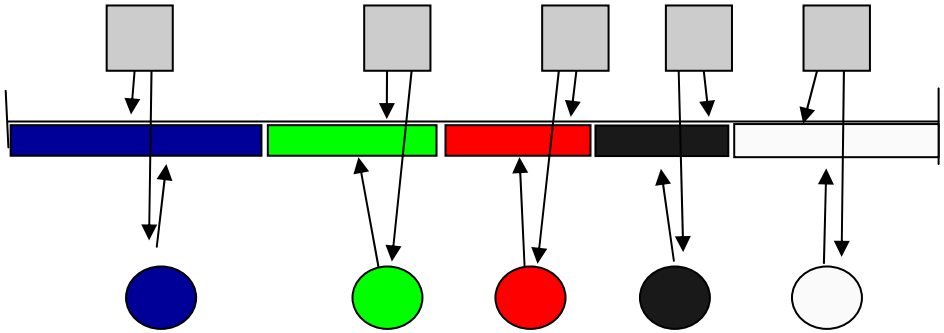


Befűzés a DHT-ba

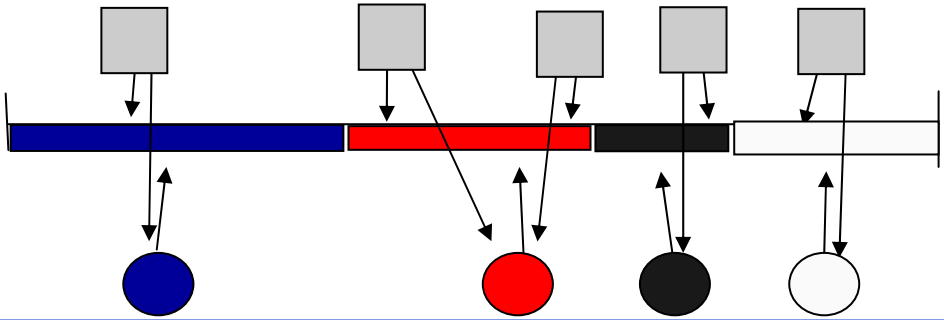
- Elosztott Hash-tábla
  - Peer-ek et hash-eljük egy helyre
  - Mindegyik egy tartományért felelős
  - Adatokat szintén hash-eljük



- Bekapcsolódik egy új Peer (csomópont)
  - A szomszédok átadják a tartományuk egy részét és a hozzátartozó adatokat

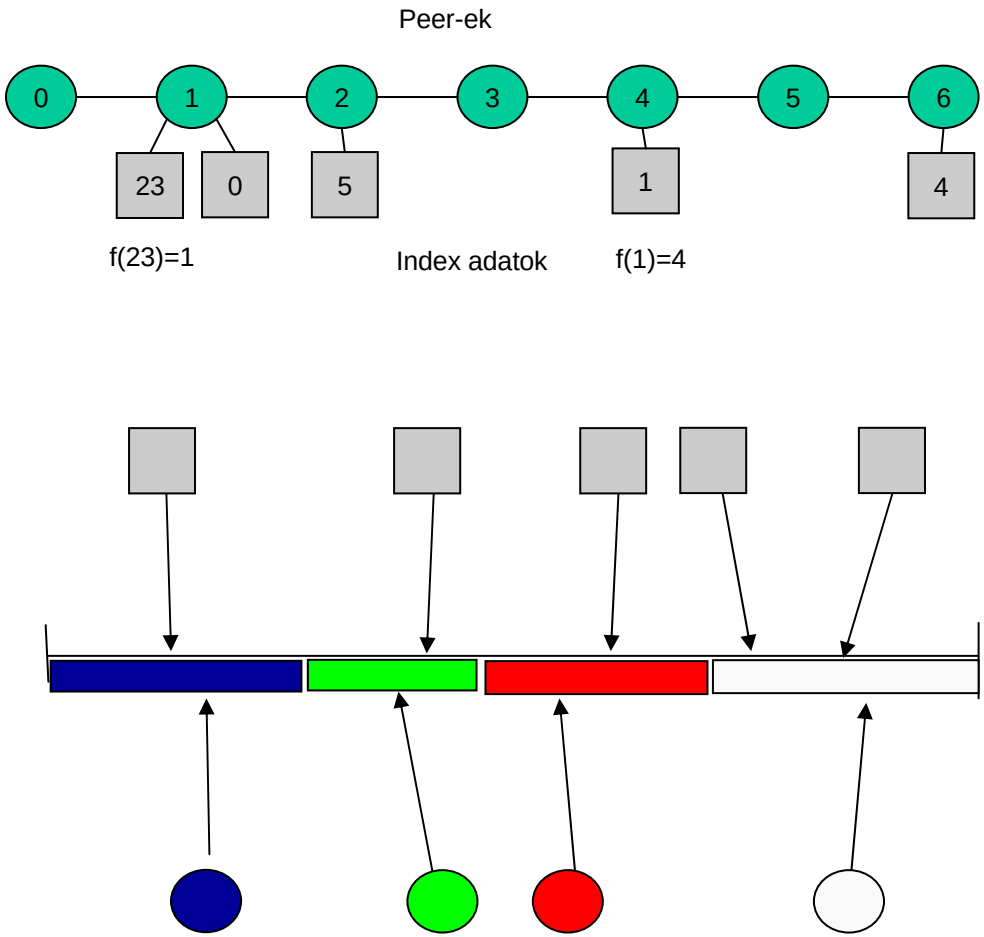


- Egy Peer elhagyja a hálózatot
  - A szomszédai átveszik a tartományát és a hozzátartozó adatokat



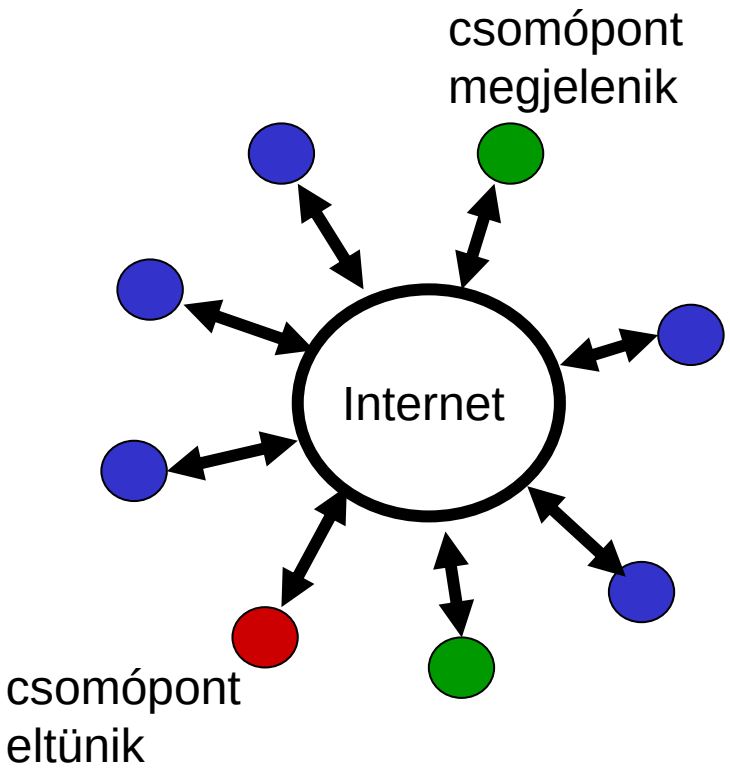
DHT tulajdonságai

- Előny
  - Minden adatot egyértelműen hozzá lehet rendelni egy Peer-hez
  - Egy Peer csatlakozása vagy kilépése a hálózatból csak a szomszédainál okoz változást
- DHT-t sok P2P hálózat használ
- Még tisztázni kell:
  - Az összeköttetések struktúráját



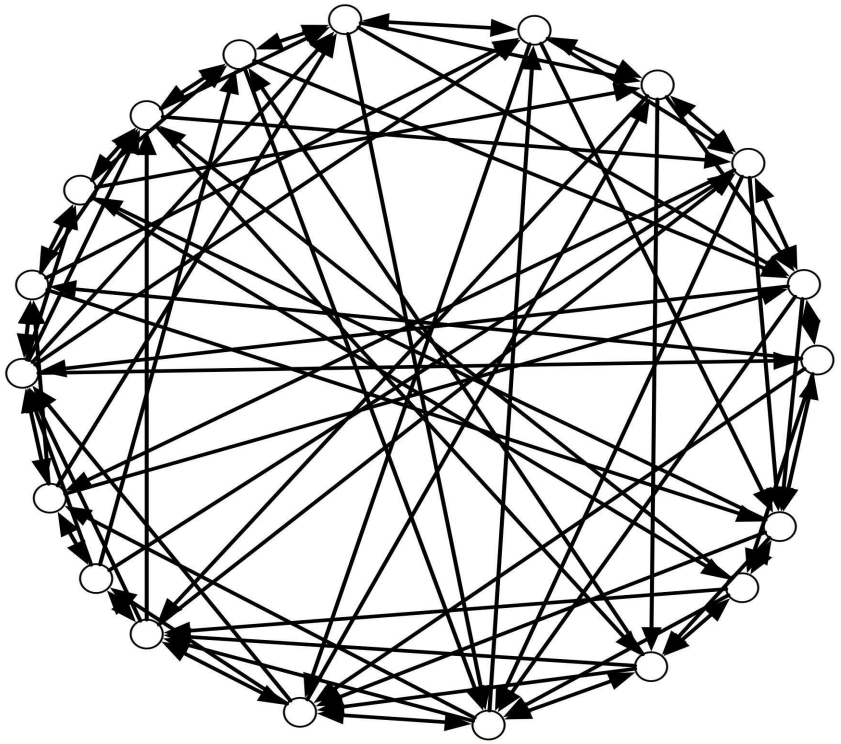
Peer-to-peer Netzwerke

- Peer-to-peer halózatok elosztott rendszerek
  - Központi kontrol és hierarchikus struktúrák nélkül
  - Azonos szoftware-rel
  - Nagyfokú dinamikával, azaz csomópontok megjelennek és eltünnek
  - Sok csomóponttal
  - Kevés hálózat-információval



Chord

- DHT (elosztott hash-tábla), ahol a leképezési tartomány {0,..,2^m-1}
  - m elegendően nagy
- A Peer-ek egy gyűrűt formálnak
- A gyűrűben útrövidítések vannak bizonyos utódokhoz, melyek távolsága exponenciálisan növekszik

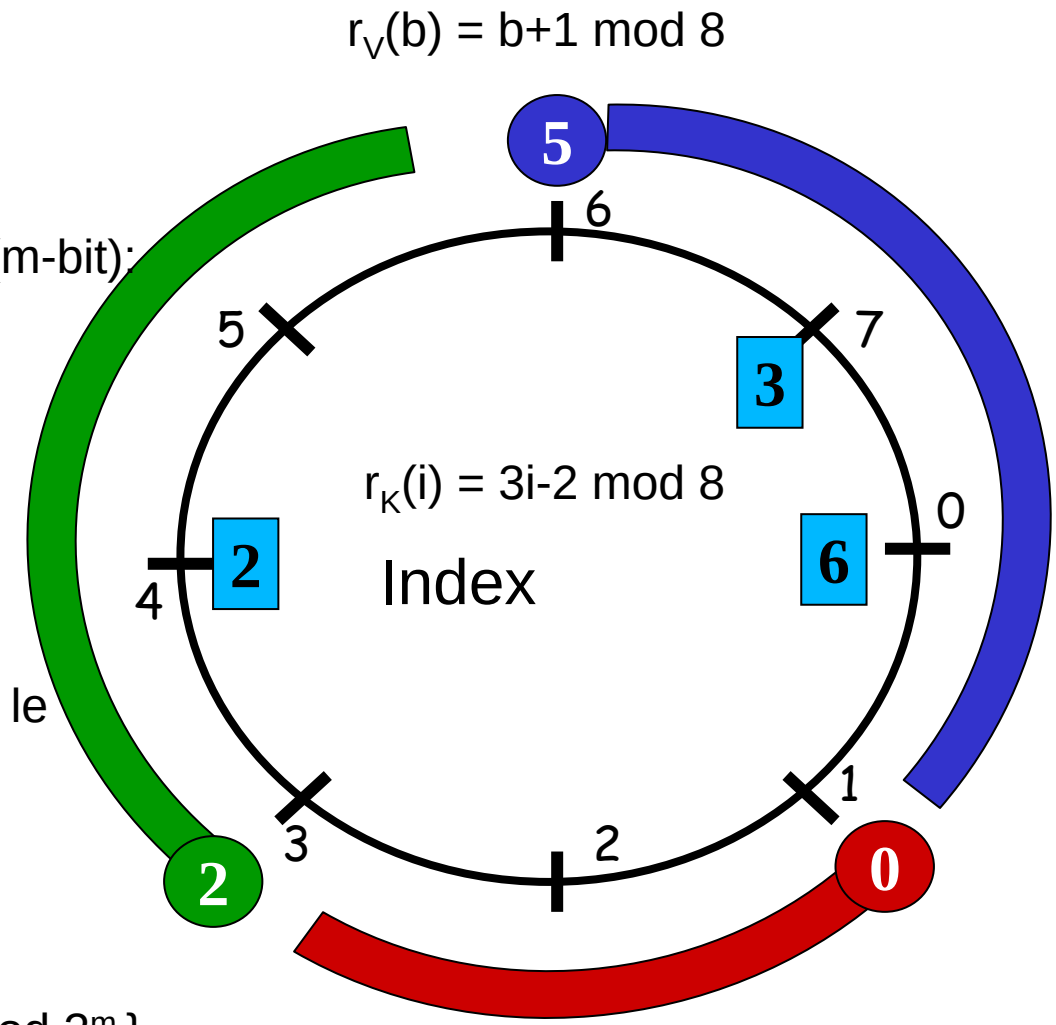


Chord mint DHT

- V: csomópontok halmaza, |V| = n
- K: kulcsok halmza (adatok), |K| = k
- m: hash-függvény értékének hossza (m-bit): m >> log max{k,n}

- Két hash-függvény, ami {0,..,2^m-1}–re képez
  - $r_V(b)$: b ∈ V Peer-t képezi le {0,..,2^m-1}-re véletlenül
  - $r_K(i)$: i ∈ K kulcsot (adatot) képezi le {0,..,2^m-1}-re véletlenül

- Egy i kulcs leképezése egy b Peer-re b = $f_V(i)$
  - $f_V(i)$:= arg $\min_{b \in V}\{ (r_K(i) - r_V(b)) \bmod 2^m \}$



Chord adatstruktúrája

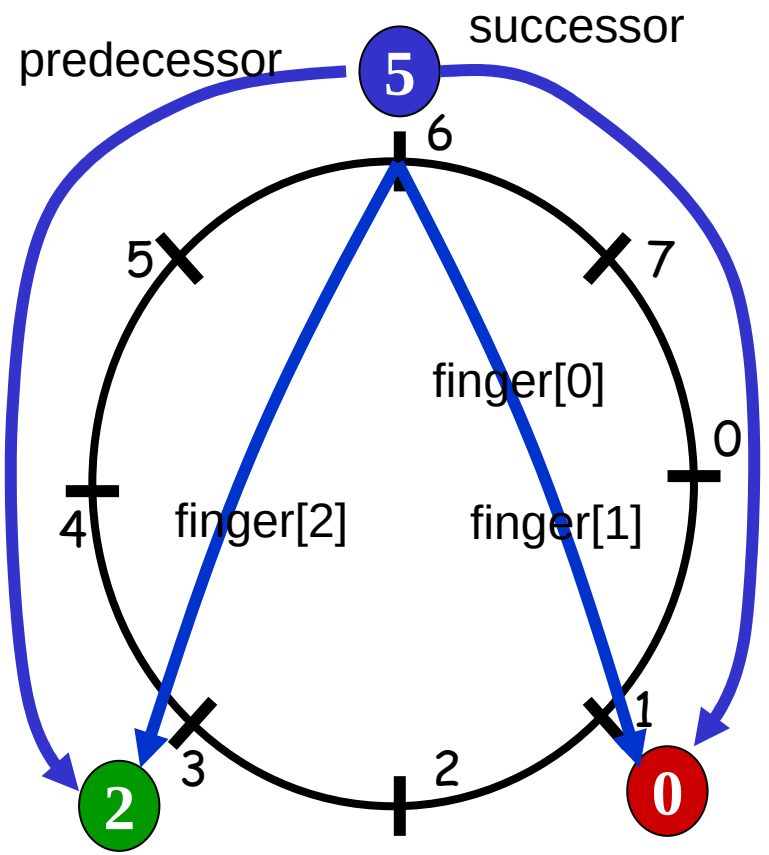
Minden b csomóponthoz tároljuk:

- successor: következő csomópont a gyűrűn
- predecessor: megelőző csomópont
- minden i ∈ {0,..m-1} -re
  - finger[i] : az a csomópont, amely képe $r_V(b)+2^i \bmod 2^m$ értékét követi, azaz legalább $r_V(b)+2^i \bmod 2^m$ és azok között a legkisebb
  - egyszerűbb jelölés miatt: finger[m] := b
- Kicsi i esetén a finger-bejegyzések gyakran azonosak
  - csak különböző finger-bejegyzéseket tárolunk

Lemma:

A különböző finger-bejegyzések száma b csomóponton O(log n) nagy valószínűséggel.

nagy valószínűséggel = $1 - n^{-c}$



Keresés a Chord-ban

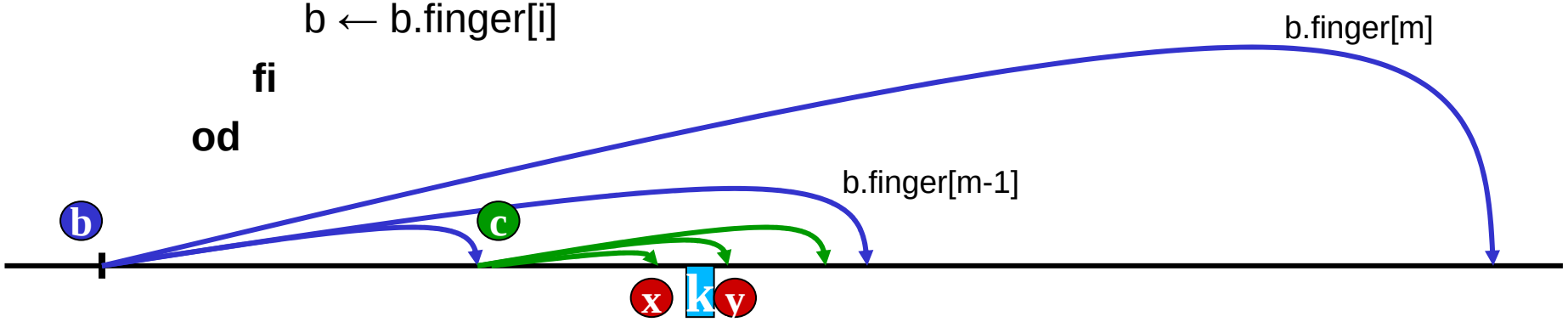
Tétel 1:

Egy kulcs keresése O(log n) ugrást igényel nagy valószínűséggel. Kulcs befűzése és törlése O(log n) üzenetcserét igényel nagy valószínűséggel.

Keresőalgoritmus k kulcshoz:

- Főrutin: Kezdjünk egy tetszőleges b csomópontnál

  while not $r_K(k) \in [\, r_V(b), r_V(b.successor)\,)$ **do**
  for i=m-1 **downto** 0 **do**
  if $r_K(k) \in [\, r_V(b.finger[i]), r_V(finger[i+1])\,)$ **then**
  b ← b.finger[i]
  fi
  od



Egyensúly a Chord-ban

- n: csomópontok száma a P2P-hálózatban
- k: kulcsok száma ≥ 1

Tétel 2:

A Chord adatstruktúra a következő tulajdonságokkal rendelkezik:

1. **Egyensúly és Terhelés**: nagy valószínűséggel ($1-n^{-c}$) minden csomópont legfeljebb $O(\frac{k \log n}{n})$ kulcsot tárol.
2. **Dinamika:** Ha egy új csomópont kapcsolódik a hálózathoz, vagy egy csomópont elhagyja a hálózatot, nagy valószínűséggel legfeljebb $O(\frac{k \log n}{n})$ kulcsot kell mozgatni (a szomszéd csomóponton tárolni).

Biz.:

- …

A Cord adatstruktúra tulajdonságai

Lemma 1:

A $r_V(b.succ) - r_V(b) \bmod 2^m$ távolság

1. várható értéke $2^m/n$,
2. nagy valószínűséggel legfeljebb $O((2^m/n) \log n)$ és
3. nagy valószínűséggel legalább $(2^m/n) / n^c$ egy konstans $c > 0$ –ra.
4. Egy w $2^m/n$ hosszú intervallumban a csomópontok száma nagy valószínűséggel
   a) $\Theta(w)$, ha $w=\Omega(\log n)$
   b) legfeljebb $O(w \log n)$, ha $w=o(\log n)$

Lemma 2:

Azon csomópontok száma, melyek egy finger-mutatója b-re mutat

1. várható érték $O(\log n)$,
2. nagy valószínűséggel legfeljebb $O(\log n)$

Bizonyítások?

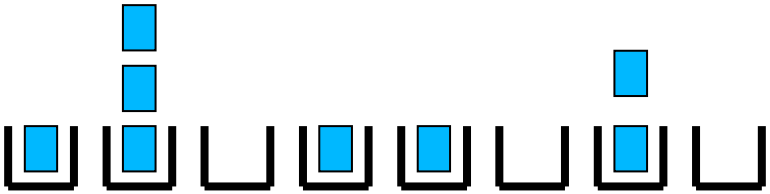
- Hogy lehet ezeket a tulajdonságokat bebizonyítani
- Abstrakt modell: labdák és kosarak (bins and balls)
- Chernoff-egyenlőtlenség felhasználása

Labdák és kosarak

Lemma 3:

Ha m labdát véletlenül n kosárba dobunk, akkor:

1. A labdák számának várható értéke egy kosaranként m/n.
2. Annak a valószínűsége, hogy k labda esik egy bizonyos kosárba: $\binom{m}{k}(\frac{1}{n})^k(1-\frac{1}{n})^{m-k}$.



Lemma 4:

Ha Ha m=n labdát véletlenül n kosárba dobunk, akkor:

1. Annak a valószínűsége, hogy egy bizonyos kosár üres marad, konstans (konvergál 1/e -hez). Az üres kosarak számának várható értéke konvergal n/e –hez.
2. Annak a valószínűsége, hogy több mint k ln n labda esik egy bizonyos kosárba, legfeljebb $O(n^{-c})$ egy konstans k-ra és c-re.

Biz.: 1.: Lemma 3, pont 2 szerint: $\binom{n}{0}(\frac{1}{n})^0(1-\frac{1}{n})^n = (1-\frac{1}{n})^n \approx \frac{1}{e}$.

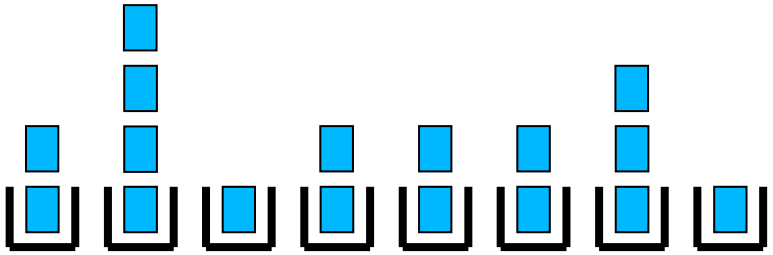
2. Chernoff-korlát

Labdák és kosarak

Lemma 5

Legyen k>0 egy fix konstans. Ha m= k n log n labdát véletlenül n kosárba dobunk, akkor a következő érvényes:

1. Minden c>k –ra annak a valószínűsége, hogy több mint c log n labda esik egy kosárba legfeljebb $O(n^{-c'})$, ahol c‘>0 egy konstans.
2. Minden c<k –ra annak a valószínűsége, hogy egy kosárba kevesebb mint c log n labda esik, legfeljebb $O(n^{-c'})$, ahol c‘>0 egy konstans.



Biz.:

Chernoff-korlát

Kitérés: Valószínűségszámítás – Markov egyenlőtlenség

- Egy diszkrét valósz. változó X várható értéke: $\mathbf{E}[X] = \sum_{x \in \mathbb{R}} \mathbf{P}[X = x]x$
- **Markov egyenlőtlenség:** Legyen k>0. Egy X≥0 diszkrét valósz. változóra, amire E[X]>0:

$$\mathbf{P}[X \geq k \cdot \mathbf{E}[X]] \leq \frac{1}{k}$$

- **Biz.:**

$$\begin{aligned}
\mathbf{E}[X] &= \sum_{x \in \mathbb{R}} \mathbf{P}[X = x]x \\
&\geq \sum_{x \geq k\mathbf{E}[x]} \mathbf{P}[X = x]x \\
&\geq \sum_{x \geq k\mathbf{E}[x]} \mathbf{P}[X = x] \cdot k \cdot \mathbf{E}[X] \\
&= k \cdot \mathbf{E}[X] \cdot \sum_{x \geq k\mathbf{E}[x]} \mathbf{P}[X = x] = k \cdot \mathbf{E}[X] \cdot \mathbf{P}[x \geq k \cdot \mathbf{E}[X]].
\end{aligned}$$

Chebyshev egyenlőtlenség

- Erősebb korlát: **Chebyshev egyenlőtlenség :**

$$\mathrm{P}[|X - \mathbf{E}[X]| \geq k] \leq \frac{V[X]}{k^2}$$

Ha a szórás (variance) ismert:

$$V[X] := \mathbf{E}[(X - \mathbf{E}[X])^2]$$

Chernoff egyenlőtlenség

- Bernoulli kisérlet
  - p valószínűséggel 1
  - 1-p valószínűséggel 0

- **Tétel 3. Chernoff egyenlőtlenség :**
  Legyenek $x_1,\ldots,x_n$ független Bernoulli kisérletek, melyekre
  $$\mathbf{P}[x_i = 1] = p,\ \mathbf{P}[x_i = 0] = 1 - p.$$
  Legyen $S_n = \sum_{i=1}^{n} x_i$.

  Ekkor:

  1. c>0: $\mathbf{P}[S_n \geq (1+c)\mathbf{E}[S_n]] \leq e^{-\frac{min\{c,c^2\}}{3}pn}$.
  2. c∈[0,1]: $\mathbf{P}[S_n \leq (1-c)\mathbf{E}[S_n]] \leq e^{-\frac{c^2}{2}pn}$.

Chernoff egyenlőtlenség bizonyítása

- t>0-ra (Markov korlátból): $\mathbf{P}[e^{tS_n} \geq k\mathbf{E}[e^{tS_n}]] = \mathbf{P}[tS_n \geq k\mathbf{E}[tS_n]] \leq \frac{1}{k}$

ahol

$$\begin{aligned}
\mathbf{E}[e^{tS_n}] &= \mathbf{E}\left[e^{t\sum_{i=1}^{n} x_i}\right] \\
&= \mathbf{E}\left[\prod_{i=1}^{n} e^{tx_i}\right] \\
&= \prod_{i=1}^{n} \mathbf{E}\left[e^{tx_i}\right] \\
&= \prod_{i=1}^{n} (e^0(1-p) + e^t p) \\
&= (1 - p + e^t p)^n \\
&= (1 + (e^t - 1)p)^n
\end{aligned}$$

válasszuk t –t és k –t:

$$t = \ln(1+c) > 0$$
$$k = e^{t(1+c)pn} / E[e^{t \cdot S_n}]$$

Chernoff egyenlőtlenség 1. rész bizonyítása

$$\begin{aligned}
\mathbf{P}[S_n \geq (1+c)pn] &\leq e^{-t(1+c)pn} \cdot (1+p(e^t-1))^n \\
&\leq e^{-t(1+c)pn} \cdot e^{pn(e^t-1)} \\
&= e^{-t(1+c)pn+pn(e^t-1)} \\
&= e^{-(1+c)\ln(1+c)pn+cpn} \\
&= e^{(c-(1+c)\ln(1+c))pn}
\end{aligned}$$

Azt kell tehát megmutatni, hogy

$$(1+c)\ln(1+c) \geq c + \tfrac{1}{3}\min\{c, c^2\}$$

Chernoff egyenlőtlenség 1. rész 1.eset: $c \leq c^2$ (azaz $c \geq 1$)

Meg kell mutatni $c \geq 1$-re, hogy:

$$(1+c)\ln(1+c) \geq c + \frac{1}{3}c$$

Ha c=1: 2 ln(2) > 4/3

Ha c>1: deriválva:

- Bal oldal deriváltja: ln(1+c) + 1
- Jobb oldal deriváltja: 4/3

- c>1 esetén a bal oldal növekedése nagyobb mint a jobb oldalé, mivel
  - ln(1+c) + 1 > ln (2) + 1 > 4/3
- Az egyenlőtlenség érvényes c>0-ra is.

Chernoff egyenlőtlenség 1. rész 2.eset: $c>c^2$ (azaz $0<c<1$)

Meg kell mutatni $0<c<1$-re, hogy $(1+c)\ln(1+c) \geq c + \frac{1}{3}c^2$

$x \in (0,1)$–re teljesül: $\sum_{i=0}^{\infty}(-x)^i = \frac{1}{1+x}$.

Így $\frac{d\ln(1+x)}{dx} = \frac{1}{1+x} = 1 - x + x^2 - x^3 + x^4 - \ldots$

Ebből következik: $\ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \ldots$

(1+x) ln(1+x)-re ebből azt kapjuk, hogy

$$(1+x)\ln(1+x) = x + \left(1 - \frac{1}{2}\right)x^2 - \left(\frac{1}{2} - \frac{1}{3}\right)x^3 + \left(\frac{1}{3} - \frac{1}{4}\right)x^4 + \ldots$$

(1+c) ln(1+c) –t behelyettesítve $c \in (0,1)$-ra megkapjuk, hogy:

$$(1+c)\ln(1+c) \geq c + \frac{1}{2}c^2 - \frac{1}{6}c^3 \geq c + \frac{1}{3}c^2$$

Chernoff egyenlőtlenség 2. rész

Meg kell mutatni, hogy:

$$e^{-t(1-c)pn} \cdot (1 + p(e^t - 1))^n \leq e^{-\frac{c^2}{2}pn}$$

ahol: $t = \ln(1 - c)$

$$\begin{aligned} e^{-t(1-c)pn} \cdot (1 + p(e^t - 1))^n &\leq e^{-t(1-c)pn} \cdot e^{pn(e^t-1)} \\ &= e^{-t(1-c)pn+pn(e^t-1)} \\ &= e^{-(1-c)\ln(1-c)pn-cpn} \end{aligned}$$

Így azt kell megmutatni, hogy:

$$-c - (1 - c)\ln(1 - c) \leq -\frac{1}{2}c^2$$

Chernoff egyenlőtlenség 2. rész

Miért igaz $-c-(1-c)\ln(1-c) \leq -\frac{1}{2}c^2$?

c=0 esetén a két oldal egyenlő (=0).

A bal oldal deriváltja ln(1-c); a jobb oldal deriváltja -c

Mivel $\ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \ldots$

következik, hogy $\ln(1-c) = -c - \frac{1}{2}c^2 - \frac{1}{3}c^3 - \ldots < -c$

Így következik a kívánt egyenlőtlenség.

Chernoff egyenlőtlenség

- Bernoulli kisérlet
  - p valószínűséggel 1
  - 1-p valószínűséggel 0

- **Tétel 3. Chernoff egyenlőtlenség :**
  Legyenek $x_1,\ldots,x_n$ független Bernoulli kisérletek, melyekre
  $$\mathbf{P}[x_i = 1] = p,\ \mathbf{P}[x_i = 0] = 1 - p.$$
  Legyen $S_n = \sum_{i=1}^{n} x_i$.

  Ekkor:

  1. c>0: $\mathbf{P}[S_n \geq (1+c)\mathbf{E}[S_n]] \leq e^{-\frac{min\{c,c^2\}}{3}pn}$.
  2. c∈[0,1]: $\mathbf{P}[S_n \leq (1-c)\mathbf{E}[S_n]] \leq e^{-\frac{c^2}{2}pn}$.

A Chord adatstruktúrájának tulajdonságai

Lemma 6

1. Két szomszédos Peer p és q közötti távolság a gyűrűn (azaz $r_V(p)-r_V(q) \bmod 2^m$)
   a) várható érték: $2^m/n$,
   b) $< O((2^m/n) \log n)$ (nagy valószínűséggel) és
   c) $> 2^m/n^c$ (nagy valószínűséggel) egy konstans $c>1$ -re
2. A gyűrű egy w $2^m/n$ hosszú intervallumába (nagy valószínűséggel)
   a) $O(\log n + w \log n)$ Peer esik, ha $w=O(\log n)$
   b) $O(w)$ Peer esik, ha $w=\Omega(\log n)$

A Chord adatstruktúrájának tulajdonságai

Lemma 6

1. Két szomszédos Peer p és q közötti távolság a gyűrűn
   a) várható érték: $2^m/n$,
   b) < $O((2^m/n) \log n)$ (nagy valószínűséggel) és
   c) > $2^m/n^c$ (nagy valószínűséggel) egy konstans c>1 -re

Biz.:

1a) Az összes távolság összege 2^m, a Peerek száma n

1b) Tekintsünk egy c $((2^m/n) \log n)$ hosszú intervallumot a gyűrűn

- A valószínűség, hogy egy Peer ebbe az intervallumba esik: c (log n)/n
- A valószínűség, hogy minden n Peer kívül esik az intervallumon:

$$\left(1 - \frac{c \ln n}{n}\right)^n \leq e^{-\frac{c \ln n}{n} \cdot n} = n^{-c}$$

- Tehát egy ilyen intervallum nem marad üres és így a távolság két szomszédos Peer között nagy valószínűséggel ≤ $2c((2^m/n) \log n)$

1c) A valószínűség, hogy egy Peer egy adott $2^m/n^c$ hosszú intervallumba esik n^{-c}

- Tehát a Peer-ek nagy valószínűséggel nem esnek túl közel egy másik Peer-hez

A Chord adatstruktúrájának tulajdonságai

Lemma 6

2. A gyűrű egy w $2^m/n$ hosszú intervallumába (nagy valószínűséggel)
   a) O(log n + w log n) Peer esik, ha w=O(log n)
   b) O(w) Peer esik, ha w=Ω(log n)

Biz.: (Chernoff korláttal). Tekintsünk egy w $2^m/n$ hosszú intervallumot

- A valószínűség, hogy egy Peer ebbe esik: p = w / n
- A Peer-ek számának várható értéke az intervallumban: p n = w

2a) 1.eset: p n ≥ 1, c>1: $\mathrm{P}[X \geq (1 + c \ln n)pn] \leq e^{-\frac{1}{3}(c \ln n)pn} \leq n^{-\frac{1}{3}c}$

2.eset: p n < 1, c>1: $\mathrm{P}[X \geq pn + c \ln n] = \mathrm{P}[X \geq (1 + \frac{c \ln n}{pn})pn]$
$\leq e^{-\frac{1}{3}\frac{c \ln n}{pn}pn} \leq n^{-\frac{1}{3}c}$

2b) p n > k ln n, c > 1: $\mathrm{P}[X \geq (1 + c)pn] \leq e^{-\frac{1}{3}ck \ln n} = n^{-\frac{1}{3}ck}$

Egyensúly a Chord-ban

- n: Peer-ek száma a P2P hálózatban
- k: kulcsok száma ≥ 1 (a tárolt adatok kulcsainak száma)

Tétel 4.

Az elemek eloszlására a Peer-eken a következő igaz:

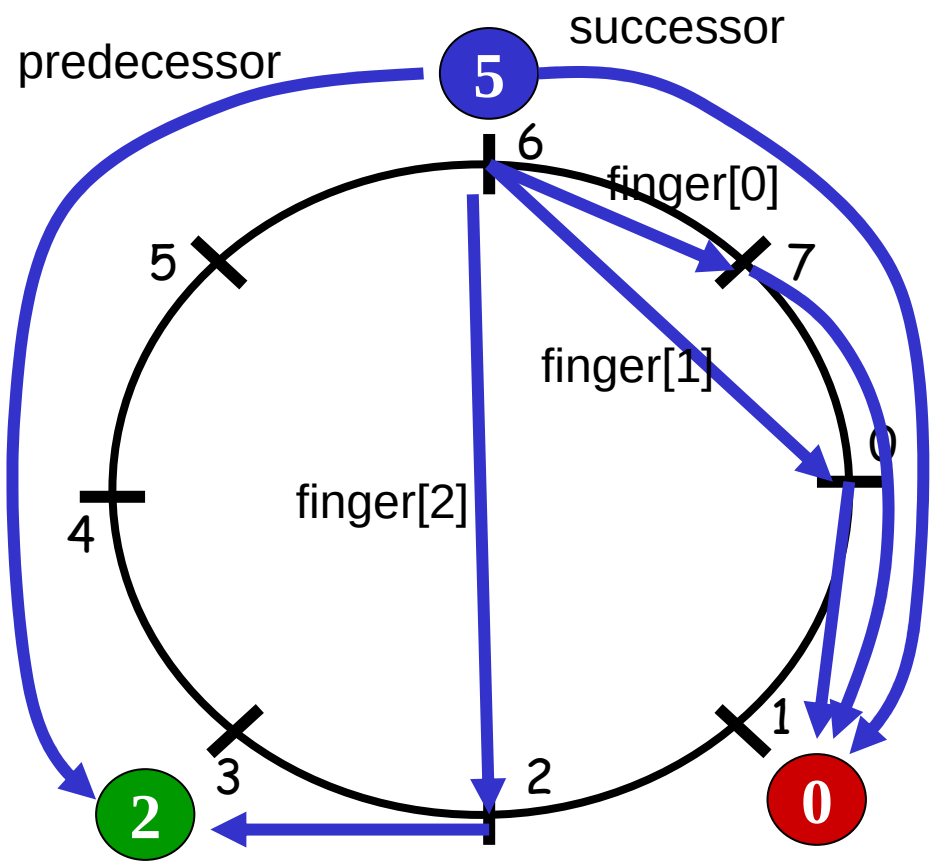
- Ha k=O(n log n):
  Minden csomópont legfeljebb O(log n + k/n $\log^2$ n) kulcsot tárol nagy valószínűséggel
- Ha k=Ω(n log n):
  Minden csomópont legfeljebb O(k/n log n) kulcsot tárol nagy valószínűséggel

- **Biz.:**
  - Chernoff korlát

Chord adatstruktúrája

Minden b csomóponthoz tároljuk:

- successor: következő csomópont a gyűrűn
- predecessor: megelőző csomópont
- minden i ∈ {0,..m-1} -re
  - az i-edik ujj: finger[i]: az a csomópont, amely képe $r_V(b)+2^i \bmod 2^m$ értékét követi, azaz legalább $r_V(b)+2^i \bmod 2^m$ és azok között a legkisebb
  - jelölje finger[m]:=b
- Kicsi i esetén a finger-bejegyzések gyakran azonosak
  - csak különböző finger-bejegyzéseket tárolunk



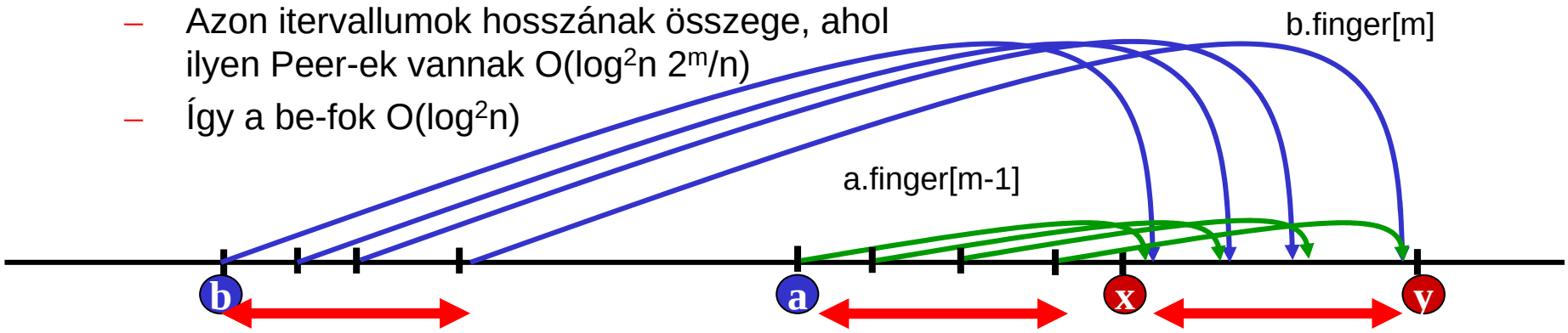
Az ujjak száma

Lemma 7

1. A CHORD hálózatban a ki-fok O(log n) nagy valószínűséggel
2. A CHORD hálózatban a be-fok O($\log^2$ n) nagy valószínűséggel

Biz.:

1. A minimális távolság két Peer között $2^m/n^c$ (nagy valószínűséggel)
   - Így a ki-fok legfeljebb c log n (nagy valószínűséggel)
2. A maximális távolság két szomszédos Peer x,y között O(log n $2^m/n$)
   - Minden Peer a, amelynek egy ujja a.finger[i] erre az intervallumra mutat, növeli y be-fokát egyel.
   - Azon itervallumok hosszának összege, ahol ilyen Peer-ek vannak O($\log^2 n$ $2^m/n$)
   - Így a be-fok O($\log^2 n$)



Keresés a Chord-ban

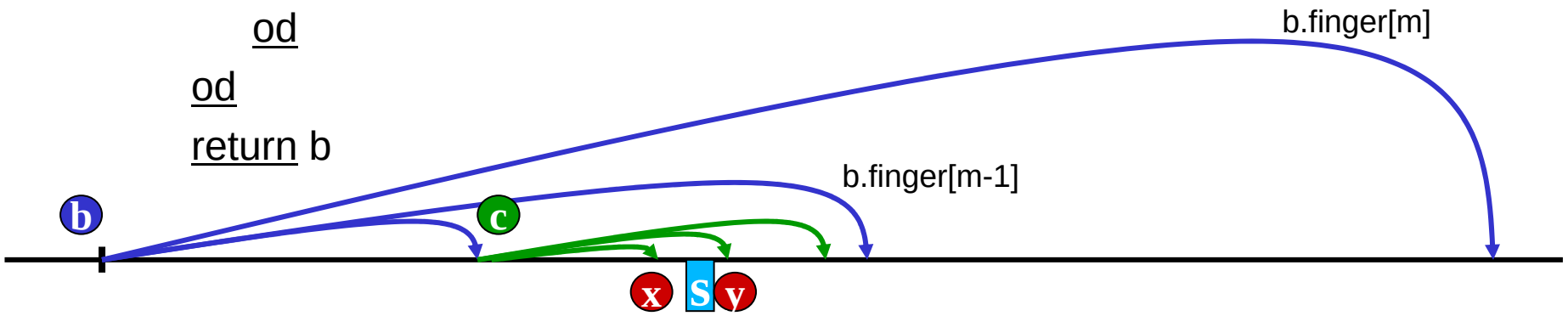
Tétel 5

A keresés nagy valószínűséggel O(log n) ugrást tartalmaz

Kereső algoritmus s kulccsal:

- Főrutin: Legyen b egy csomópont (a keresés elindítója)

```
while not r_K(s) ∈ [ r_V(b),r_V(b.successor) ) do
    for i=m-1 downto 0 do
        if r_K(s) ∈ [ r_V(b.finger[i]), r_V(finger[i+1]) ) then
            b ← b.finger[i]
        fi
    od
od
return b
```



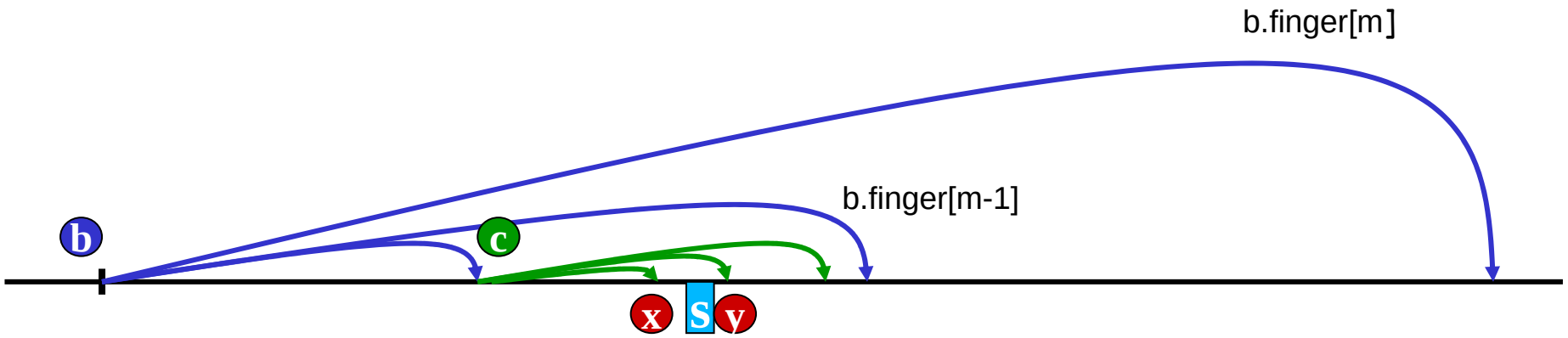
Suchen in Chord

Tétel 5

A keresés nagy valószínűséggel O(log n) ugrást tartalmaz

Biz.:

- A távolság a célhoz minden ugrással legalább feleződik
- A távolság a keresés legelején legfeljebb 2^m
- A távolság két szomszédos Peer között legalább $2^m/n^c$ nagy valószínűséggel
- Így a keresés ideje legfeljebb c log n



Peer-ek hozzáadása

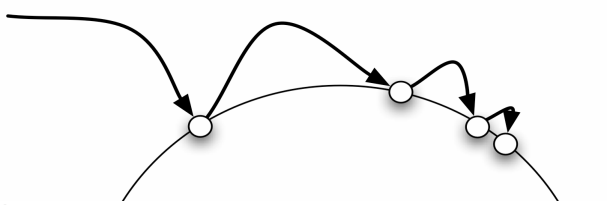
Tétel 6

Nagy valószínűséggel $O(\log^2 n)$ üzenet elegendő egy új Peer-t a Chord-ba felvenni

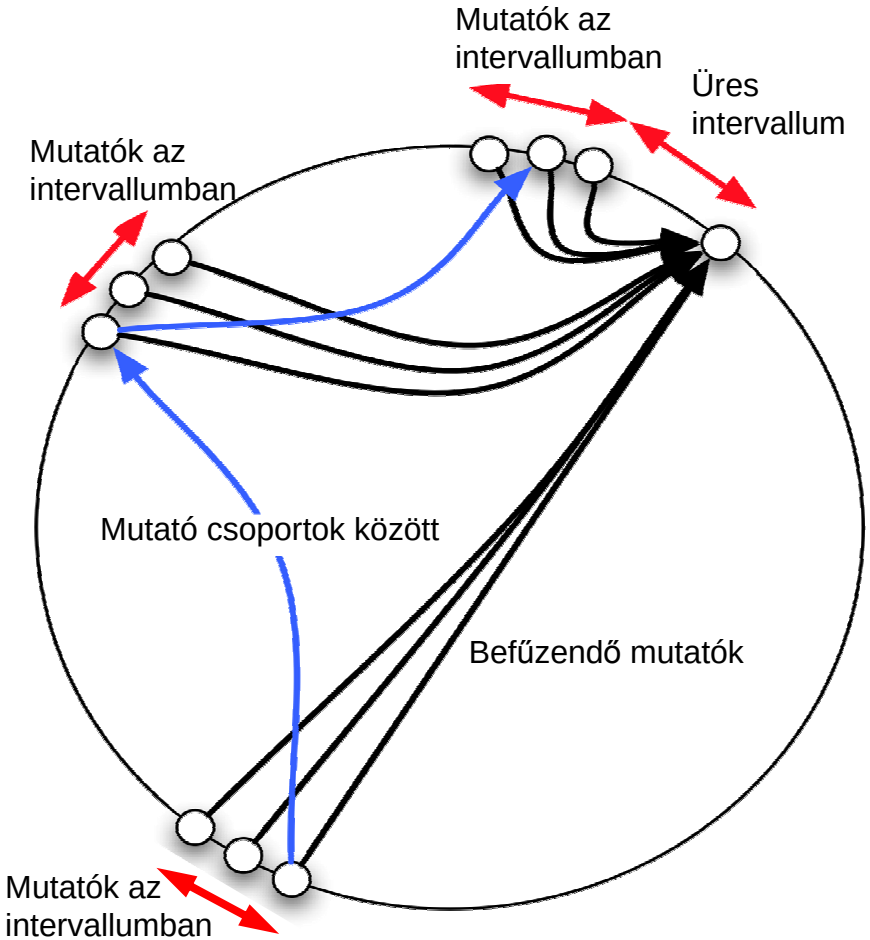
Bizonyítási ötlet:

- A cél-intervallumot megkeressük O(log n) lépésben
- A mutató (finger) információkat átvesszi az új Peer az őt megelőző és az őt követő csomóponttól és ezeket aktualizálja
  - Minden ilyen mutatót a gyűrűn legfeljebb O(log n) lépésben aktualizálhatunk
- Az új Peer p be-foka nagy valószínűséggel $O(\log^2 n)$
  - A keresés költsége minden alkalommal O(log n)
  - A Peer-ek, ahol a finger[i] mutatót aktualizálni kell p-re, maximum O(log n) sokan vannak és ezek egymással szomszédosak a gyűrűn
  - Így O(log n) keresésre van szükség, melyek mindegyikének költsége O(log n)
  - Az aktualizálás költsége minden esetben konstans

Peer-ek hozzáadása



- Először megkeressük a célintervallumot O(log n) lépésben
- Az új Peer a kimenő mutatókat (finger) átveszi az őt megelőző és az őt követő csomóponttól és ezeket aktualizálja
  - Minden ilyen mutatót a gyűrűn legfeljebb O(log n) lépésben aktualizálhatunk
- Az új Peer be-foka nagy valószínűséggel $O(\log^2 n)$
  - A keresés költsége minden alkalommal O(log n)
  - A Peer-ek, ahol a finger[i] mutatót aktualizálni kell p-re, maximum O(log n) sokan vannak és ezek egymással szomszédosak a gyűrűn.
  - Így O(log n) keresésre van szükség, melyek mindegyikének költsége O(log n)
  - Az aktualizálás költsége minden esetben konstans



Belépés és kilépés várható költsége

Tétel 7

Egy Peer belépéséhez a CHORD struktúrába szükséges üzenetek számának várható értéke O(log n)

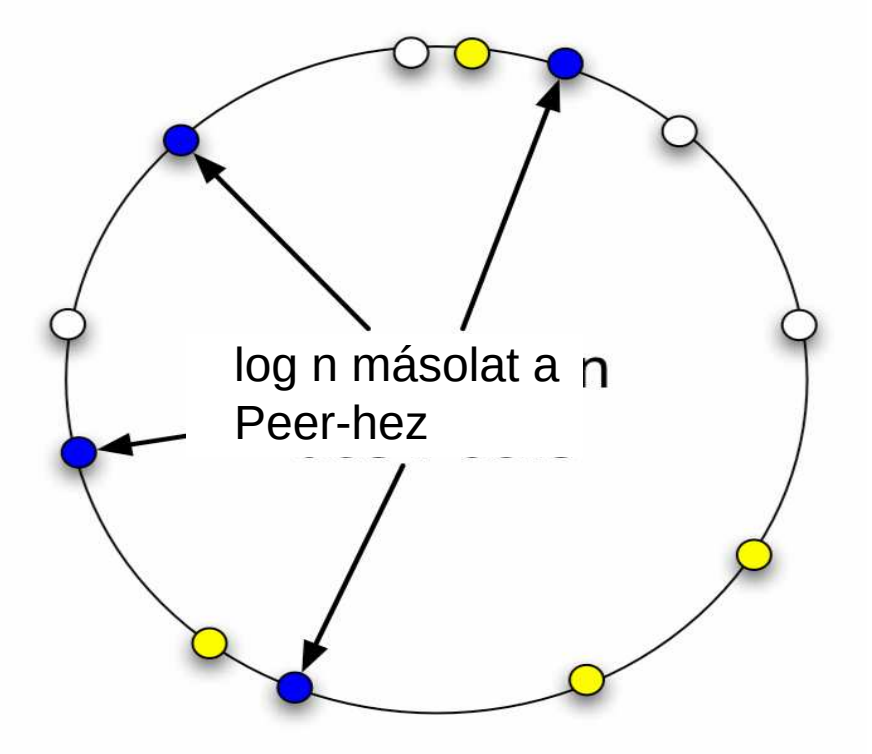
Bizonyítási ötlet:

- Minden Peer ki-foka nagy valószínűséggel O(log n)
- Ezáltal a be-fok várható értéke ugyancsak O(log n)
- Az ugrások számának a várható értéke a belépés-operációban O(1)

- Egy Peer kilépésének (törlésének) költsége ugyanakkora

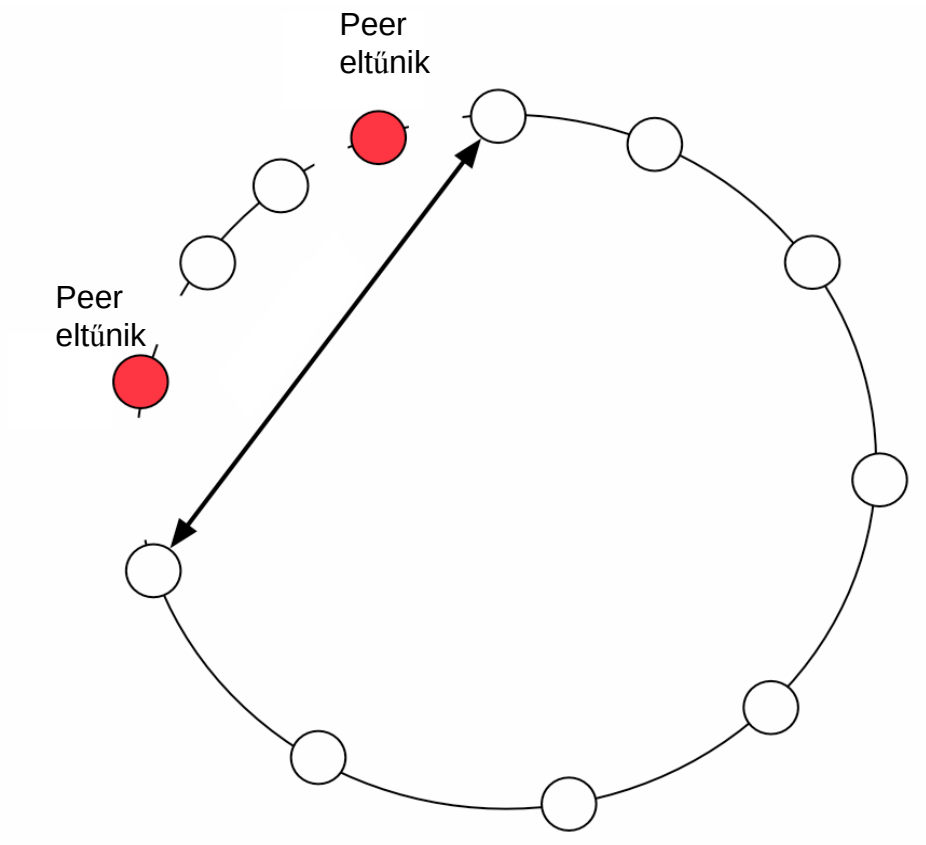
Egyensúly virtuális Peer-ek segítségével

- Minden Peer tart O(log n) virtuális Peer-t a gyűrűn
  - Ezáltal a ki-fok megnő $O(\log^2 n)$-re
- Előnyök:
  - A be-fok továbbra is $O(\log^2 n)$ nagy valószínűséggel
  - Belépés/Törlés-Operáció végrehajtható $O(\log^2 n)$ üzenettel nagy valószínűséggel
  - Ha az adatok száma $k = \Omega(n \log n)$, akkor minden Peer nagy valószínűséggel $O(k/n)$ adatot tárol



Stabilizálás a CHORD-ban

- Csomópontok belépése és törlése párhuzamosan történhet
- Ha csomópontok magukat törlik (eltünnek), nem értesítik a szomszédjaikat
  - A csomópontoknak rendszeresen tesztelni kell, hogy a szomszédai jelenvannak-e
  - Ha a szomszéd eltünt, a finger-információ segítségével keresünk új szomszédot
  - Ha két csomópont egyidejűleg tűnik el, a gyűrű széteshet két részre és ez inkonzisztenciához vezethet
- Egyidejű belépés szintén inkonzisztenciához vezethet az adatstruktúrában
- Megoldás: speciális stabilizálás-operáció segítségével
  - Ötlet: egy kiesett csomópont szomszédainak a finger-információi „hasonlók“ a kieset Peer-éhez (nem kerül pontosabb bemutatásra)



Irodalom

- I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan: **Chord: A scalable peer-to-peer lookup service for Internet applications.** In *Proc. ACM SIGCOMM*, 149-160, 2001.