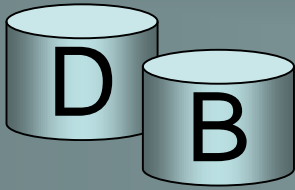


Harvat hakemistot

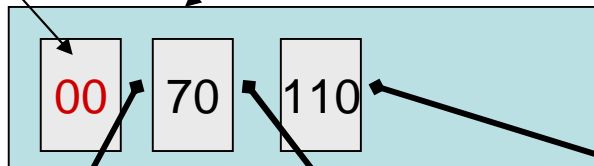
- Harvassa hakemistossa on ei ole hakemistomerkintöjä jokaista tietuetta kohden vaan yksi merkintä jotain isompaa kokonaisuutta esimerkiksi sivua tai sivujoukkoa (esim. saman uran sivut) kohti
- Harvan hakemiston käyttö edellyttää, että tietueet on järjestetty



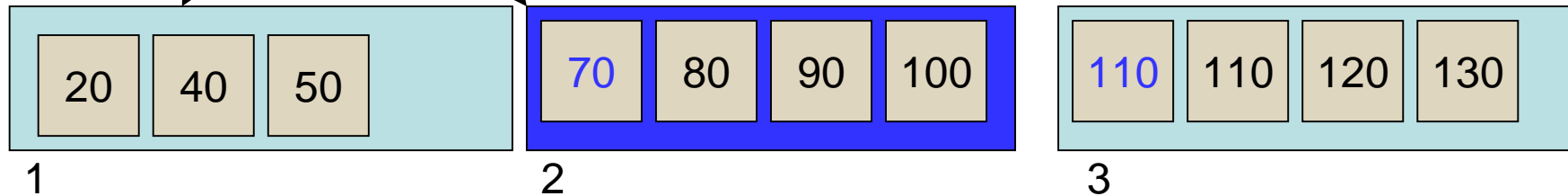
Harvat hakemistot

pienin mahdollinen

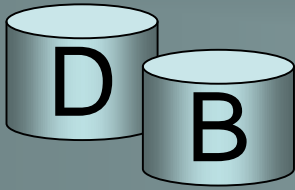
hakemistossa alarajat sivulle
kuuluville avaimille ja osoitteet sivuille



kaikki tietueet, joissa
 $70 \leq \text{avain} < 110$
kuuluvat sivulle 2



perustietueet järjestettynä peräkkäisrakenteena

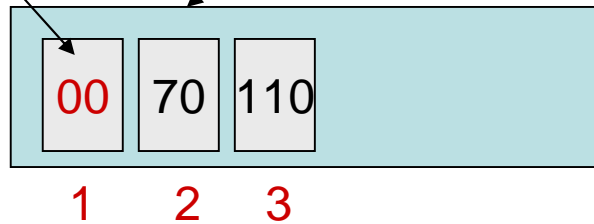


Harvat hakemistot

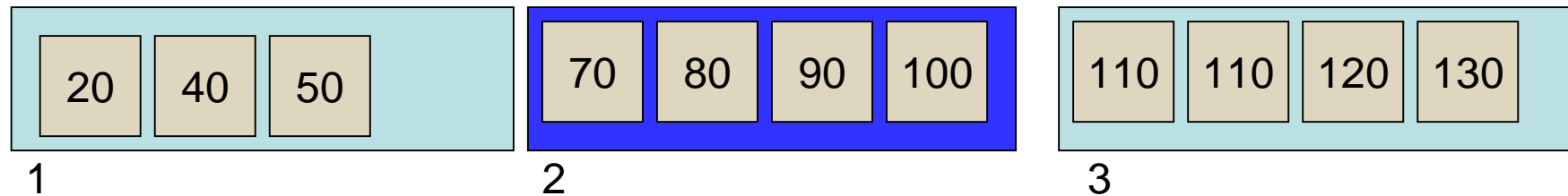
hakemistossa alarajat sivulle
kuuluville avaimille

(Voidaan toteuttaa myös ilman osoitteita,
hakemistomerkin indeksin vastaa sivunumeroa)

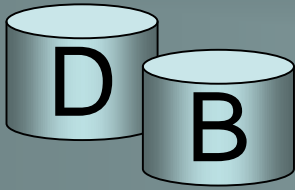
pienin
mahdollinen



kaikki tietueet, joissa
 $70 \leq \text{avain} < 110$
kuuluvat sivulle 2

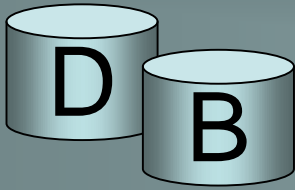


perustietueet järjestettynä peräkkäisrakenteena

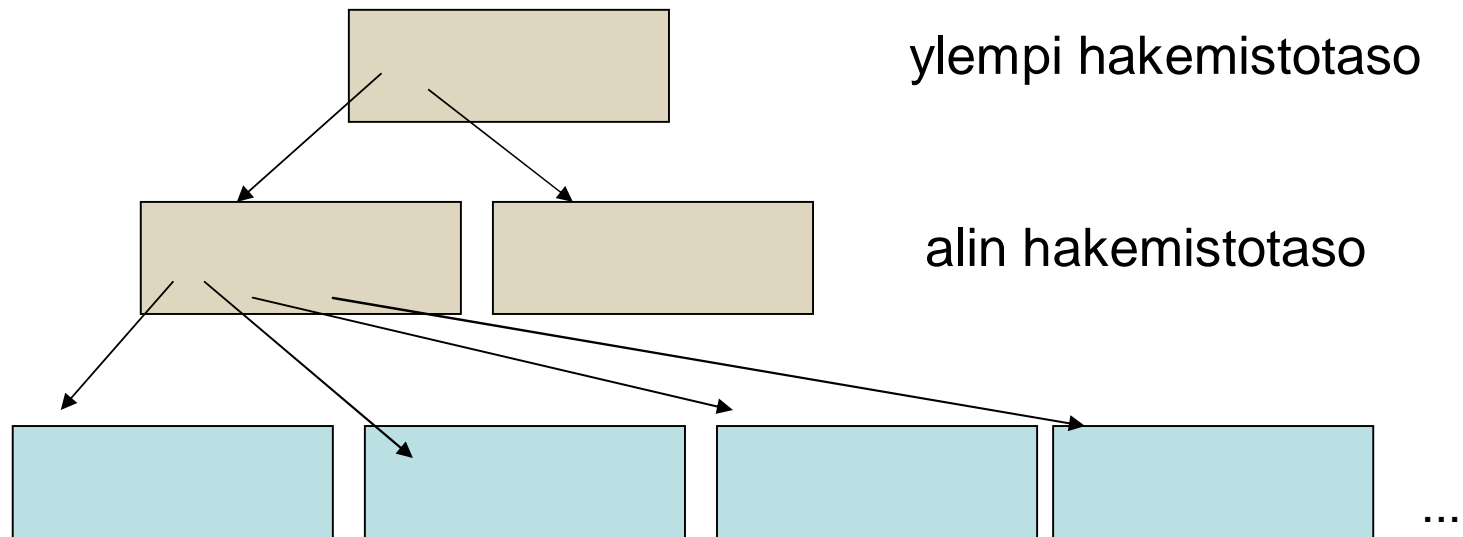


Harvat hakemistot

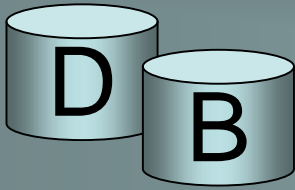
- perustietueet järjestettynä peräkkäistiedostona
- hakemistoon alarajat (tai ylärajat) sivulle kuuluvien tietueiden avaimille (+osoitteet sivuille)
- kaksi peräkkäistä hakemistoavainta määrää sivulle i kuuluvat avaimet
- **$h(i) \leq$ sivulle i kuuluva avain $< h(i+1)$**
- hakemisto voi olla monitasoinen
 - jos alin hakemistotaso tulee liian isoksi, tehdään sille samoin harva hakemisto



Harvat hakemistot

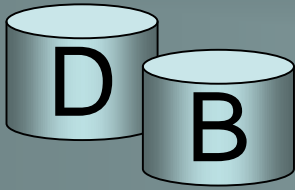


hakemistotasoja voisi olla useitakin, mutta yleensä 2 tai 3 riittää siihen, että ylimmän tason hakemisto on niin pieni, että se voidaan pitää jatkuvasti keskusmuistissa.



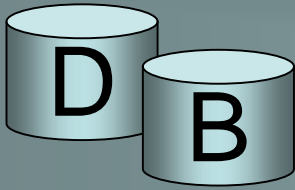
Harvat hakemistot

- Haku:
 - etsi hakemistosivulta suurin hakemistoavain, joka on pienempi tai yhtä suuri kuin etsittävän avain,
 - jos ollaan hakemistosivulla, siirry löydetyn alkion indeksin osoittamalle sivulle seuraavalla tasolla (jos hakemistomerkintöihin liittyy sivuosoite käytetään sitä indeksin tilalla)
 - tässä joudutaan suorittamaan levyhaku (ellei sivu ole puskurissa)
 - jos ollaan datasivulla, etsi tietue sivulta



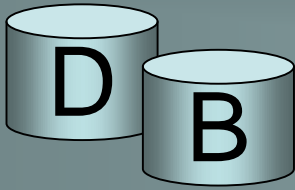
Harvat hakemistot

- Jos datasivuille mahtuu useita tietueita muodostuu harva hakemisto tiheää pienemmäksi, joten haku on nopeampaa
- Esimerkki
 - työntekijätietueen keskipituus oli 300, joten yhdelle datasivulle voitaisiin sijoittaa vaikkapa 10 tietuetta (jätetty hieman tilaa lisäyksille)
 - Tiedosto mahtuu 800 sivulle
 - Kunkin sivun pienin avain vieään hakemistoon, jolloin hakemistosivulle menee keskimäärin 400 avainta (tässä muunnelmassa ei käytetä osoitteita)
 - Tarvitaan 2 hakemistosivua alimmalla tasolla,
 - Koko hakemisto (yht. 8K) voidaan pitää keskusmuistissa, joten tietueen hakuun työntekijänumeron perusteella tarvitaan noin 1 levyhaku



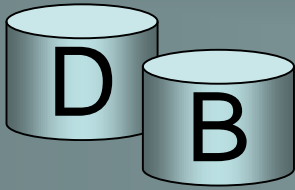
Harvat hakemistot

- Etuja
 - pieni hakemisto - tehokas haku
 - jos tietueita on N ja hakemistosivulle mahtuu X merkintää, tarvitaan hakemistoon $\text{roof}(\log_x N)$ tasoa
 - tukee myös
 - arvovälihakuja
 - hakuja avaimen alkuosalla
 - (osoite like 'Helsingin%')



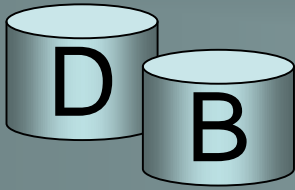
Harvat hakemistot

- Ongelmia
 - tiedostolla voi olla vain yksi harva hakemisto sillä perustietueet voidaan järjestää vain yhteen järjestykseen
 - lisäykset voivat aiheuttaa ongelmia
 - jos lisäykset hoidetaan sivukohtaisilla ketjutetuilla ylivuotosivuilla, voi ketjusta tulla pitkä, jos lisäykset kasautuvat
 - usein jätetään sivuille tilaa lisäyksille – täyttösuhde huononee eikä tämä välttämättä hidasta kovin paljon rakenteen degeneroitumista
 - tietueiden lisääminen satunnaisessa järjestyksessä on erityisen ongelmallista



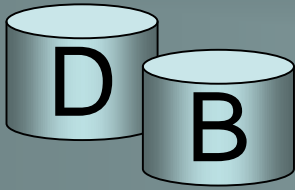
Harvat hakemistot - ISAM

- ISAM (**indexed sequential access method**)
- indeksoitu peräkkäisrakenne
- harva **monitasoinen staattinen** hakemisto
 - **staattinen**: hakemisto pysyy luontinsa jälkeen muuttumattomana kunnes tiedosto organisoidaan uudelleen ja sille luodaan uusi hakemisto
 - tiedosto jakautuu **perussivuihin** ja **ylivuotosivuihin**
 - ylivuotosivuille viedään lisäykset, jotka eivät mahdu perussivuille, ylivuotosivut ketjutetaan



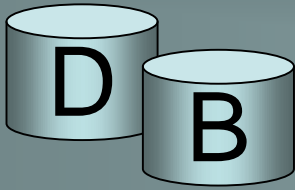
Harvat hakemistot - ISAM

- Tiedosto **luodaan järjestetyn aineiston pohjalta**
 - Tietueita lisätään tiedoston loppuun (järjestyksessä)
 - Sivuille jätetään tilaa lisäyksille, esim. 30 % - 50%.
 - Hakemistomerkinässä **avainarvo** ja **sivuosoite**
 - Viedään merkintä hakemistosivulle aina kun aloitetaan uusi datasivu
 - ensimmäisen hakemistosivun kohdalla viedään hakemisto-avaimeksi **low-value** = pienin mahdollinen arvo, muille sivuille sivun ensimmäisen tietueen avain
 - Hakemistosivut voidaan laittaa täyteen, sillä niitä ei päivitetä.



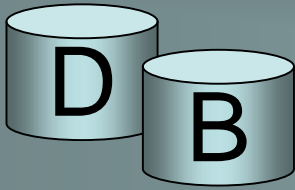
Harvat hakemistot - ISAM

- Rakenne sopii hyvin tiedostoille, joihin tulee vain vähän lisäyksiä. Jos lisäyksiä tulee runsaasti tarvitaan ajoittaisia uudelleenorganisointeja.
- Ylivuotoketjut hidastavat läpilukuaikaa samoin kuin järjestetyssä peräkkäisrakenteessa.
- Koska hakemistosivut eivät muutu, ei niitä myöskään tarvitse lukita rinnakkaisessa käsittelyssä
 - prosessin ei tarvitse jonottaa hakemistosivua



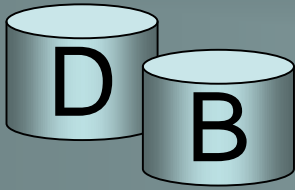
Harvat hakemistot - ISAM

- Poistot
 - kun ylivuotosivu tyhjenee poiston seurauksena sivu irrotetaan ketjusta ja voidaan ottaa uudelleen käyttöön
 - kun datasivu tyhjenee poiston seurauksena se jätetään tyhjäksi. Sivun tarjoaa tällöin lisäyspaikan, jos sen avainalueelle sattuisi tulemaan lisäyksiä.



Harvat hakemistot - ISAM

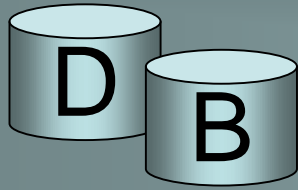
- Alkuperä:
 - IBM:n ISAM rakenteessa oli alunperin kaksi hakemistotasoa alimpana urahakemisto (uran suurin avain), ja seuraavalla tasolla sylinterihakemisto (sylinterin suurin avain)
- Muunnelmia
 - ISAM rakennetta voidaan käyttää hakemistorakenteena, siten että 'datatietueina' eivät olekaan oikeat datatietueet vaan tiheän hakemiston hakemistomerkinnot.



Harvat hakemistot - ISAM

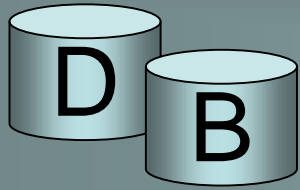
- ISAM on keskimäärin hyvin tehokas rakenne hakemistoavaimen avulla tehtävissä hakuoperaatioissa (yhtäsuuruus , alkuosa, arvoväli)
- Rakenne ei kykene takaamaan kasvavalle tiedostolle kovin pientä ylärajaa levyhakujen määrälle pahimmassa tapauksessa, jolloin levyhakuja olisi luokkaa:

lisäysten määrä/sivukoko

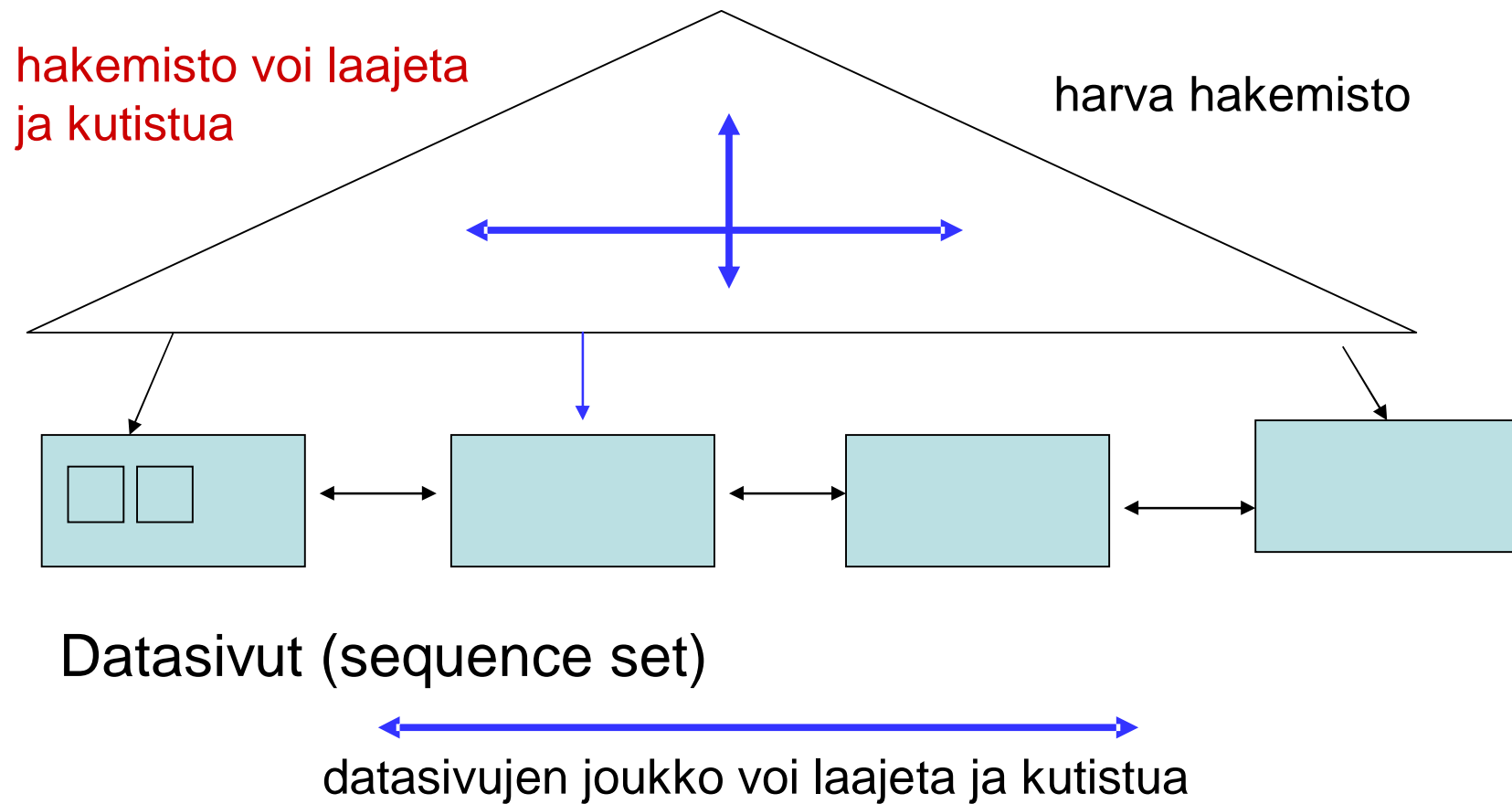


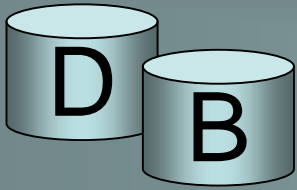
Dynaamiset hakemistorakenteet B+ -puu

- **B+ -puut** ovat laajalti tietokantojen yhteydessä käytetty tiedostorakenne
- Rakenne on oikeastaan ISAM rakenteen dynaaminen muunnos
 - Rakenteen alimmalla tasolla ovat datatietueet järjestettynä rakenteena kuten ISAM:ssa,
 - Datasivut eivät kuitenkaan ole välttämättä fyysisesti peräkkäisiä vaan ne on usein kytketty kaksisuuntaiseksi linkitetyksi listaksi. Erillisiä ylivuotosivuja ei ole, vaan listarakennetta ylläpidetään dynaamisesti.
 - Datasivujen 'päällä' on monitasoinen harva hakemisto, joka ISAM:ista poiketen on dynaaminen



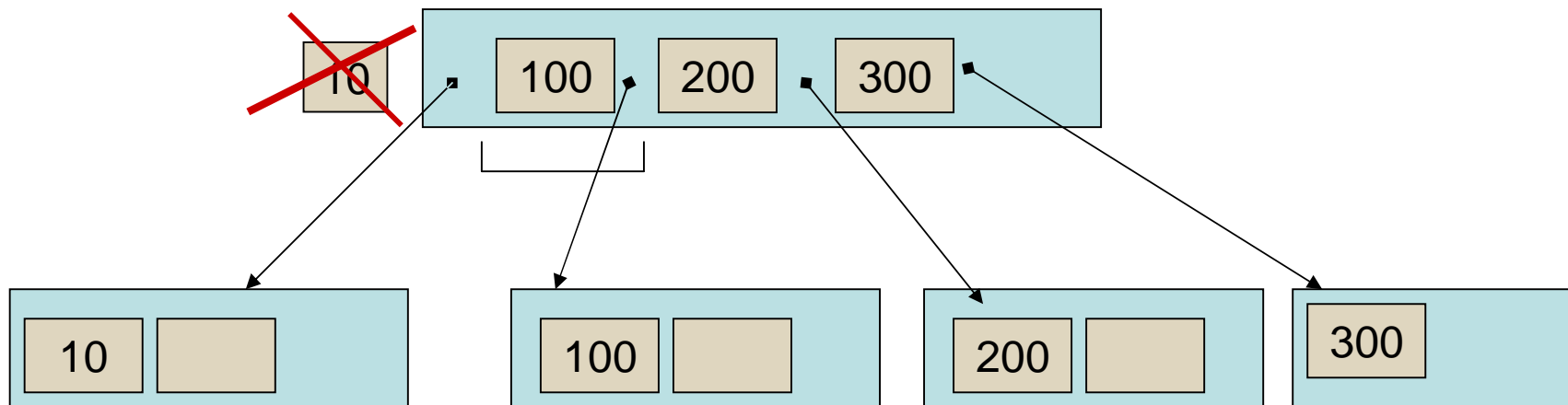
Dynaamiset hakemistorakenteet B+ -puu

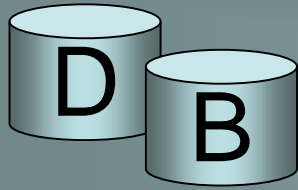




Dynaamiset hakemistorakenteet B+ -puu

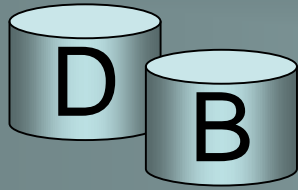
- B+ -puun hakemisto on rakenteeltaan ISAM:in kaltainen, mutta
 - hakemistosivujen täyttöaste ylintä sivua lukuunottamatta on välillä **50-100%**
 - hakemiston alaisuuteen kuuluvan ensimmäisen sivun pienin avainarvo jää pois





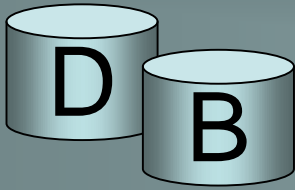
Dynaamiset hakemistorakenteet B+ -puu

- hakemistosivut **eivät ole** staattisia, niihin voi tulla uusia hakemistomerkinlöjä, merkinlöjä voidaan poistaa, sivuja voidaan jakaa ja yhdistää.
- B+ -puun tasapainovaatimuksena on, että jokainen **hakupolku on yhtä pitkä**, eli jokaisen tietueen hakemiseen tarvitaan yhtä monta levyhakua (elleivät sivut ole puskuureissa)
- lisäys- ja poisto-operaatiot säilyttävät hakurakenteen tasapainon



Dynaamiset hakemistorakenteet B+ -puu

- Tarkastellaan **kertaluokan** (order) d B+ -puuta
- Tällaisen puun jokaisessa **hakemistosolmussa** paitsi juuressa (ylin taso) on $d \leq m \leq 2d$ hakemistomerkinettä ja lisäksi yksi osoite
- hakemistosivulla ovat siis osoitteet p_0, \dots, p_m ja avaimet K_1, \dots, K_m
- Kuten ISAM:ssa tietue kuuluu osoitteessa p_i ($i=1, \dots, m$) olevan sivun alaisuuteen, jos sen avaimelle pätee $K_i \leq \text{avain} < K_{i+1}$
- Tietue kuuluu osoitteessa p_0 olevan sivun alaisuuteen, jos sen $\text{avain} < K_1$

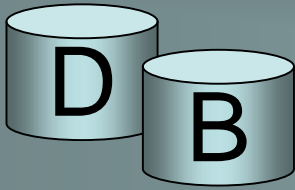


Dynaamiset hakemistorakenteet B+ -puu

- Haku (oletetaan, että avaimet yksikäsitteisiä)

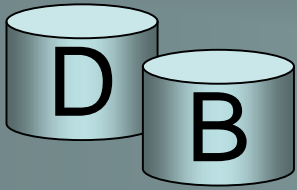
```
function search(node, key) {  
    if (node is leaf) return node;  
    else  
        if (key < K[1]) return search(p[0], key);  
        else  
            if (key >= K[m]) return search(p[m], key);  
            else {  
                find i such that K[i] <= key < K[i+1];  
                return search(p[i], key);  
            }  
}
```

Palauttaa datarivun osoitteen



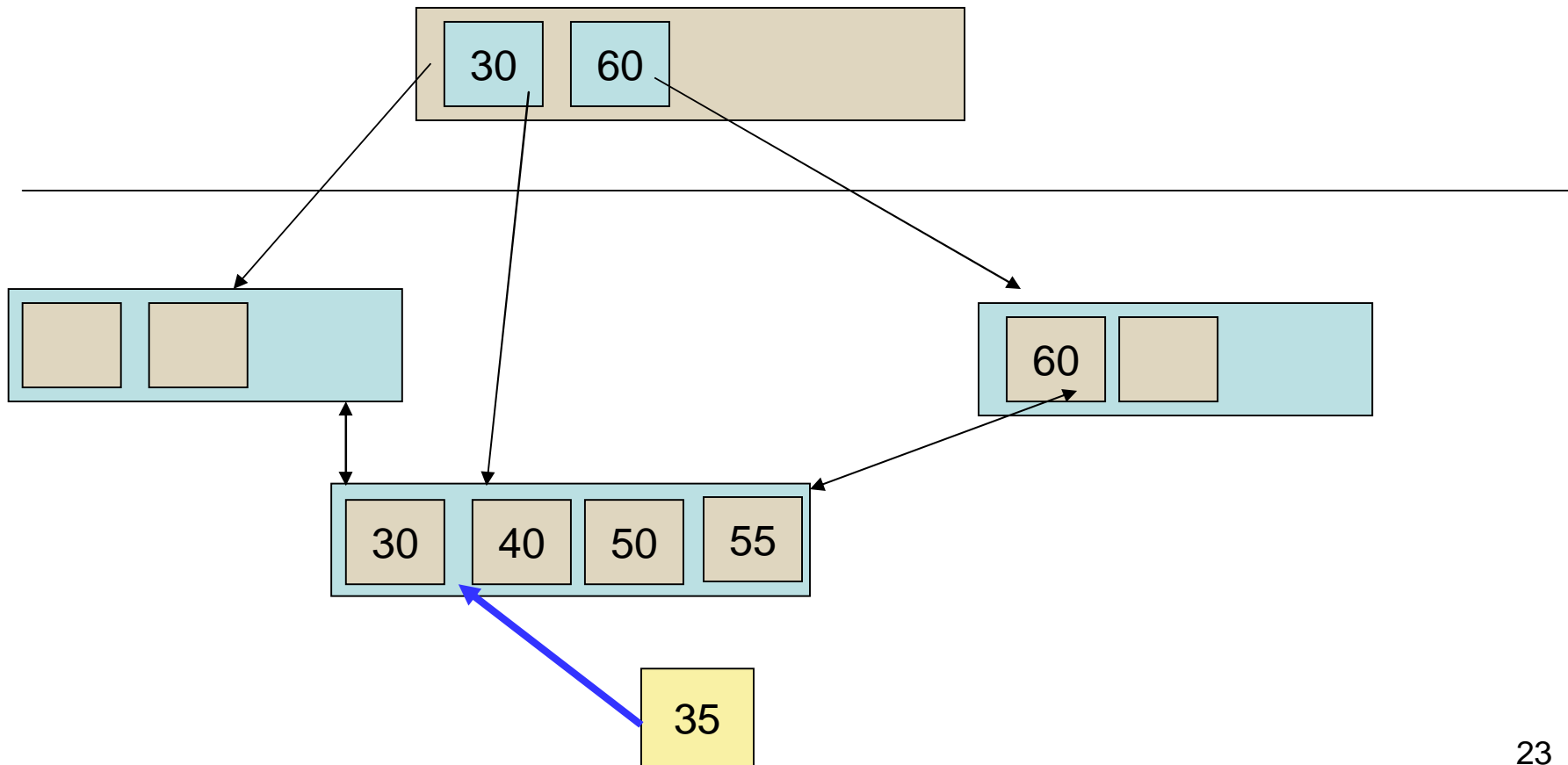
Dynaamiset hakemistorakenteet B+ -puu

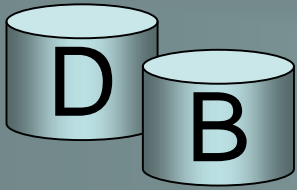
- Lisäykset:
 - Lisättäessä tietuetta B+ -puuhun sille haetaan sijoituspaikka edellä kuvatulla hakuoperaatiolla
 - Jos sivulla on tilaa, tietue lisätään järjestyksessä oikealle paikalleen sivulla
 - Jos tietue ei mahdu sivulle se sijoitetaan loogisesti oikealle paikalleen ja jaetaan ylivuotanut sivu:
 - Otetaan käyttöön uusi sivu ja kytketään se linkitettyyn listaan
 - Siirretään puolet alkuperäisen sivun tietueista uudelle sivulle
 - Muodostetaan hakemistomerkinä (K_{new}, P_{new}) uuden sivun ensimmäisen tietueen avaimesta ja sivun osoitteesta ja lisätään tämä siihen hakemistotietueeseen, jonka alaisuuteen jaettu sivu kuului



Tasapainotus lisäyksessä

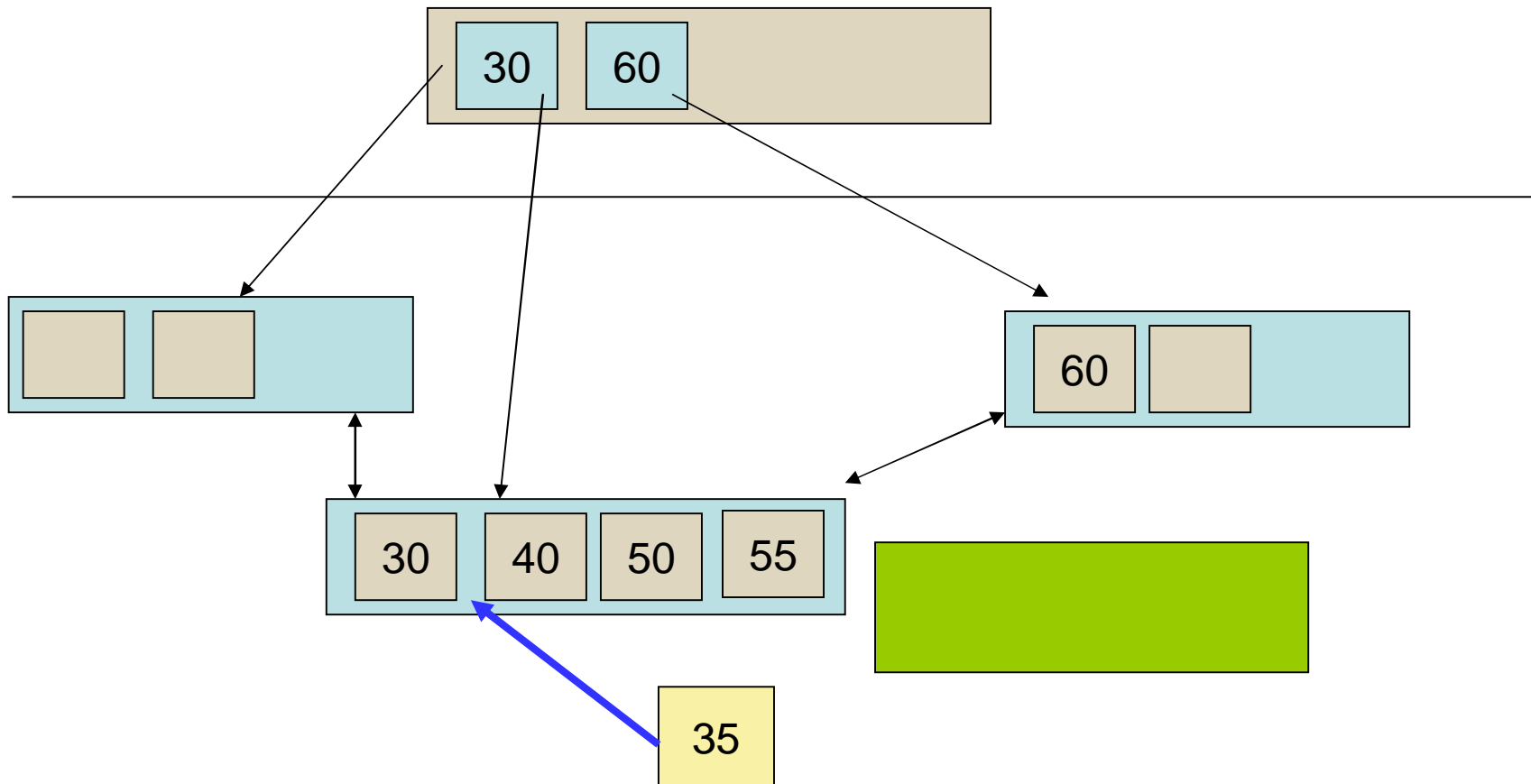
lisättävä ei mahdu sivulle

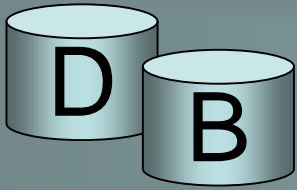




Tasapainotus lisäyksessä

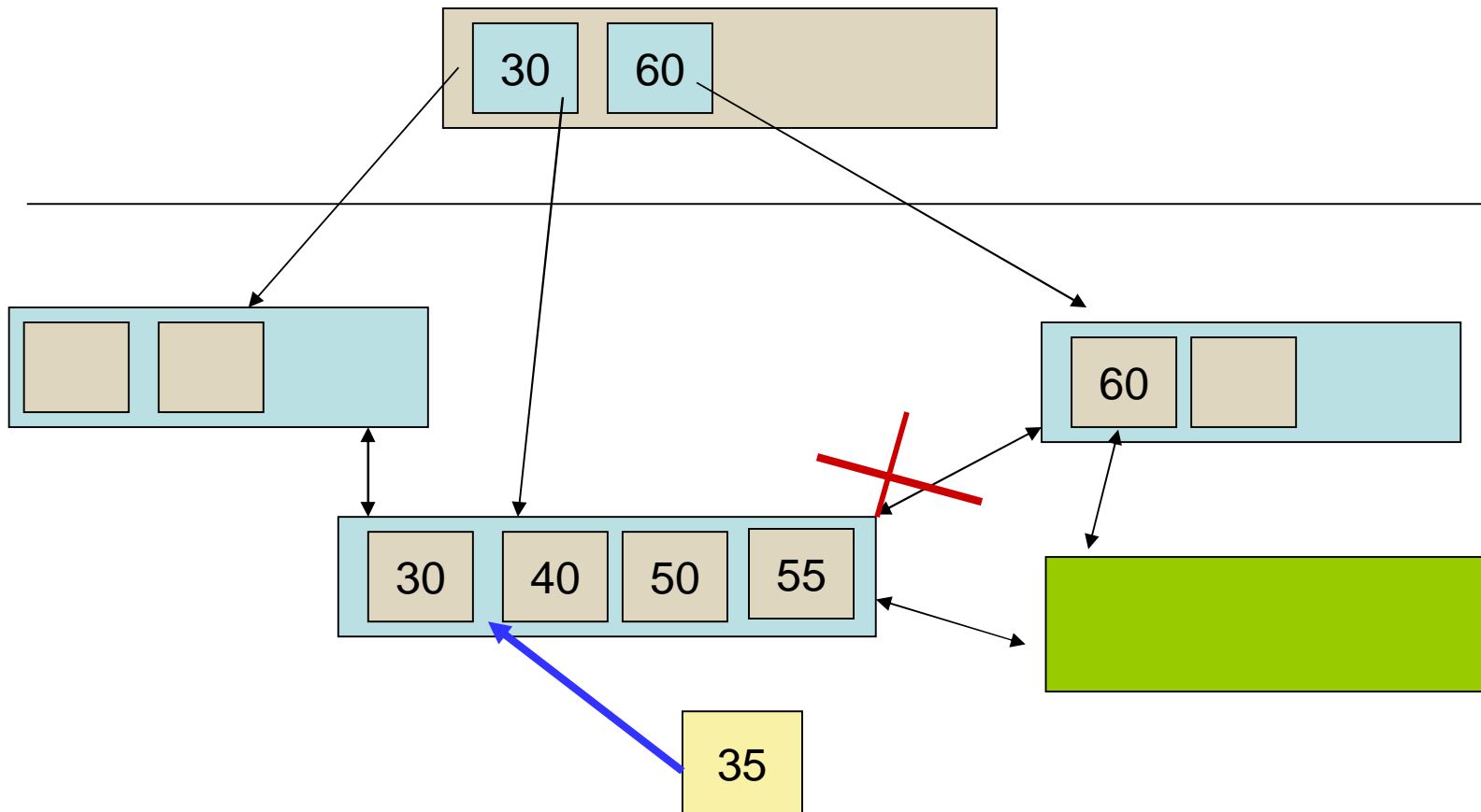
otetaan käyttöön uusi datasiivu

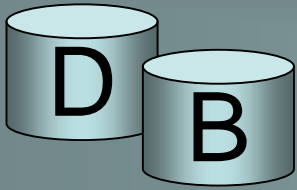




Tasapainotus lisäyksessä

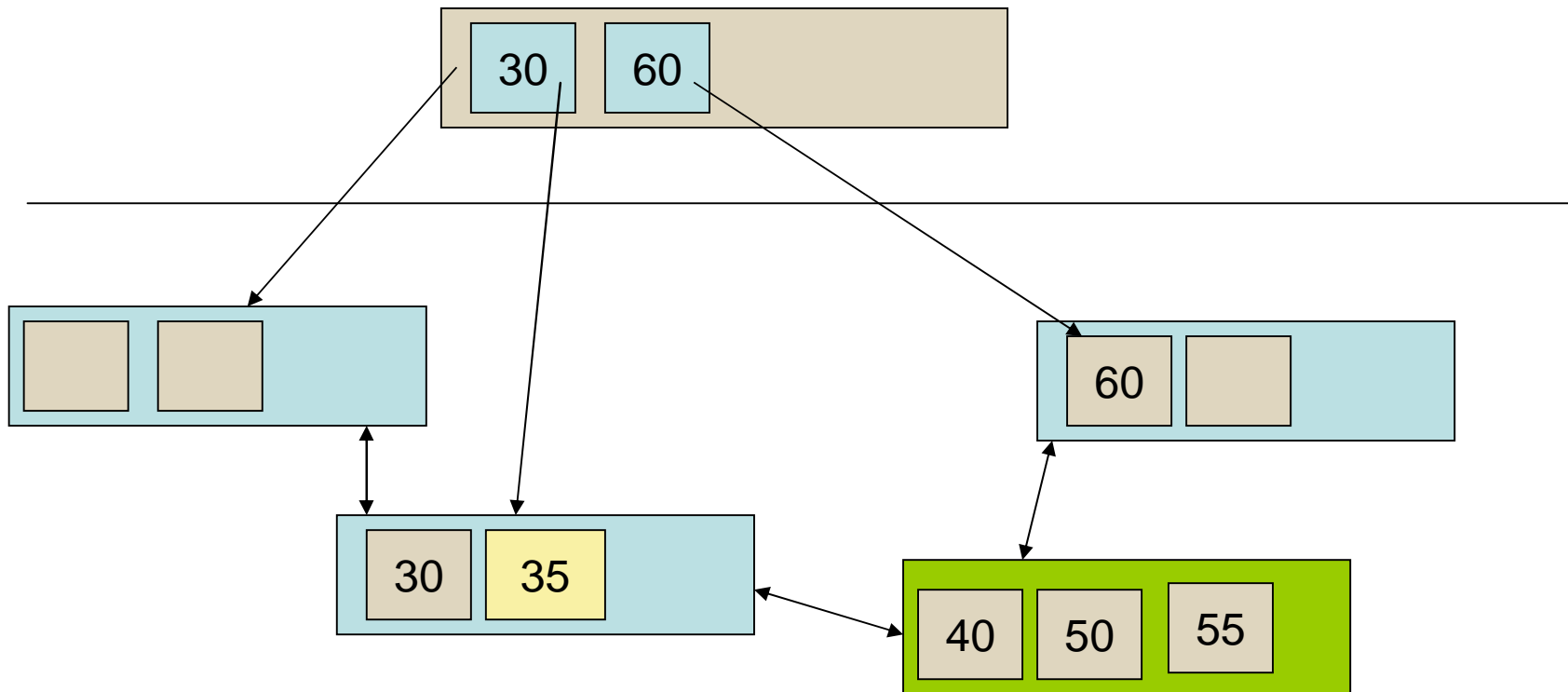
kytketään uusi datasivu sivuketjuun

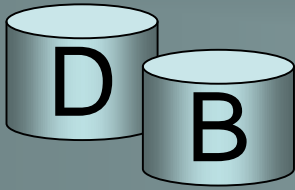




Tasapainotus lisäyksessä

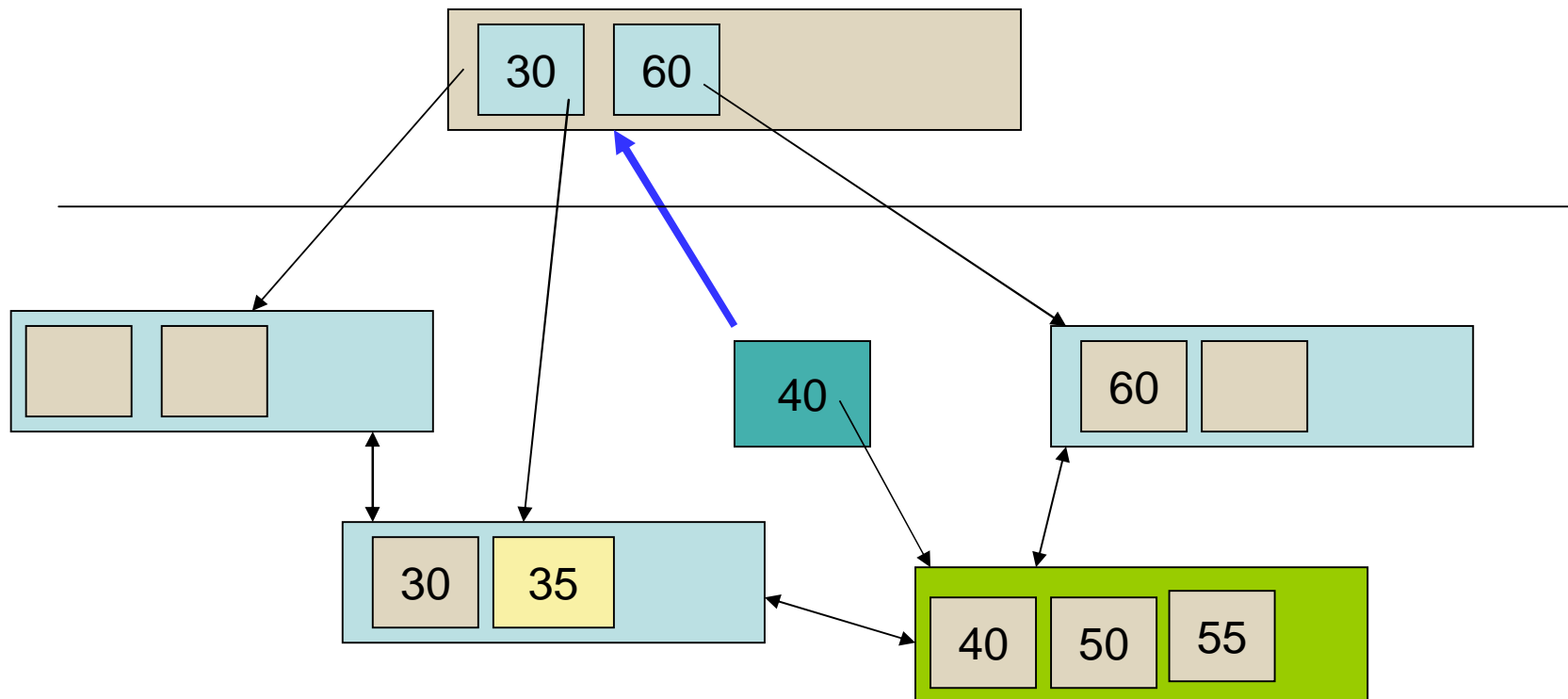
Siirretään puolet ylivuotavan sivun sisällöstä uudelle sivulle

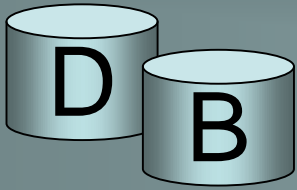




Tasapainotus lisäyksessä

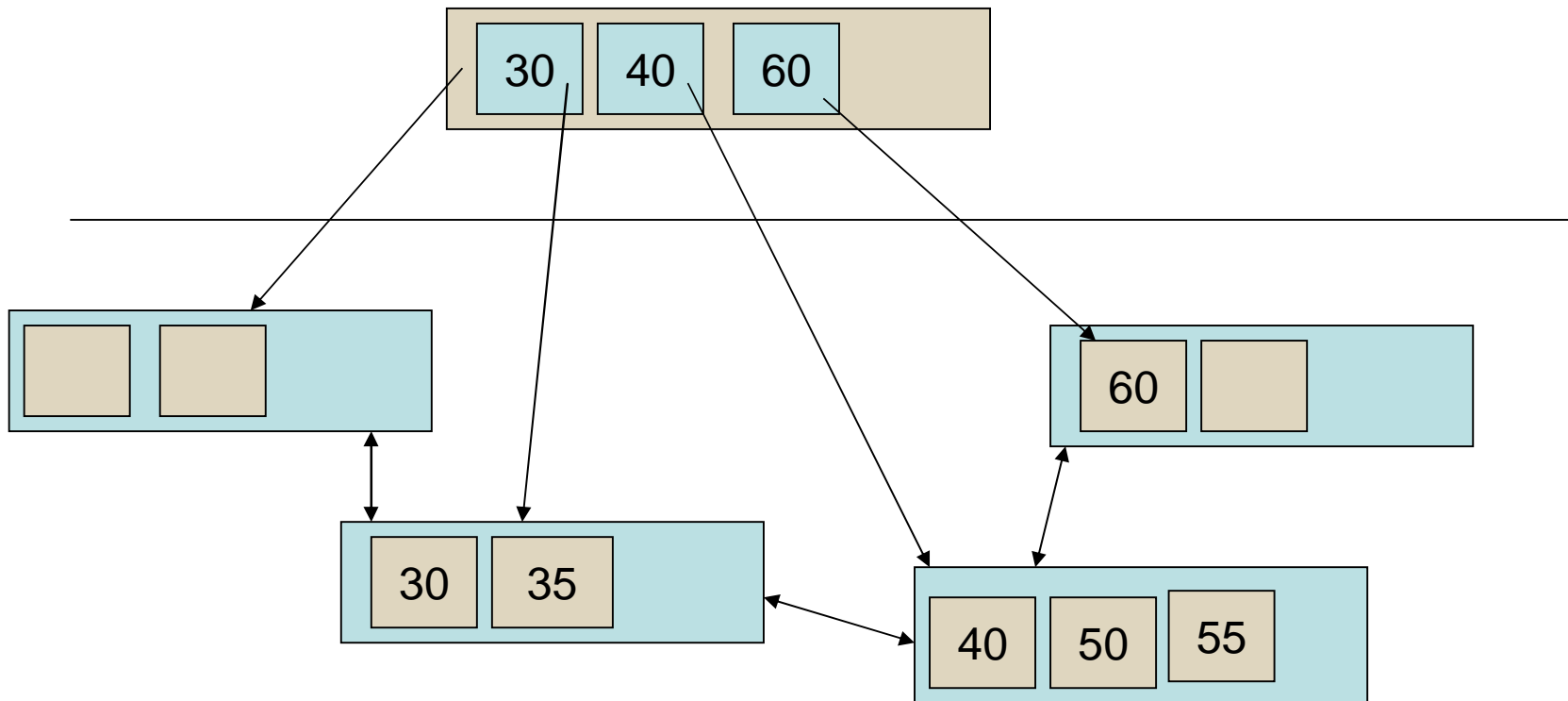
tehdään uudelle sivulle hakemistotietue ja lisätään se hakemistosivulle, jonka alaisuuteen tietue lisättiin

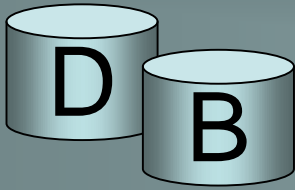




Tasapainotus lisäyksessä

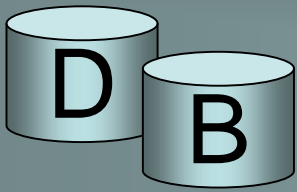
Hakemistotietue mahtui sivulle – lisäys on valmis





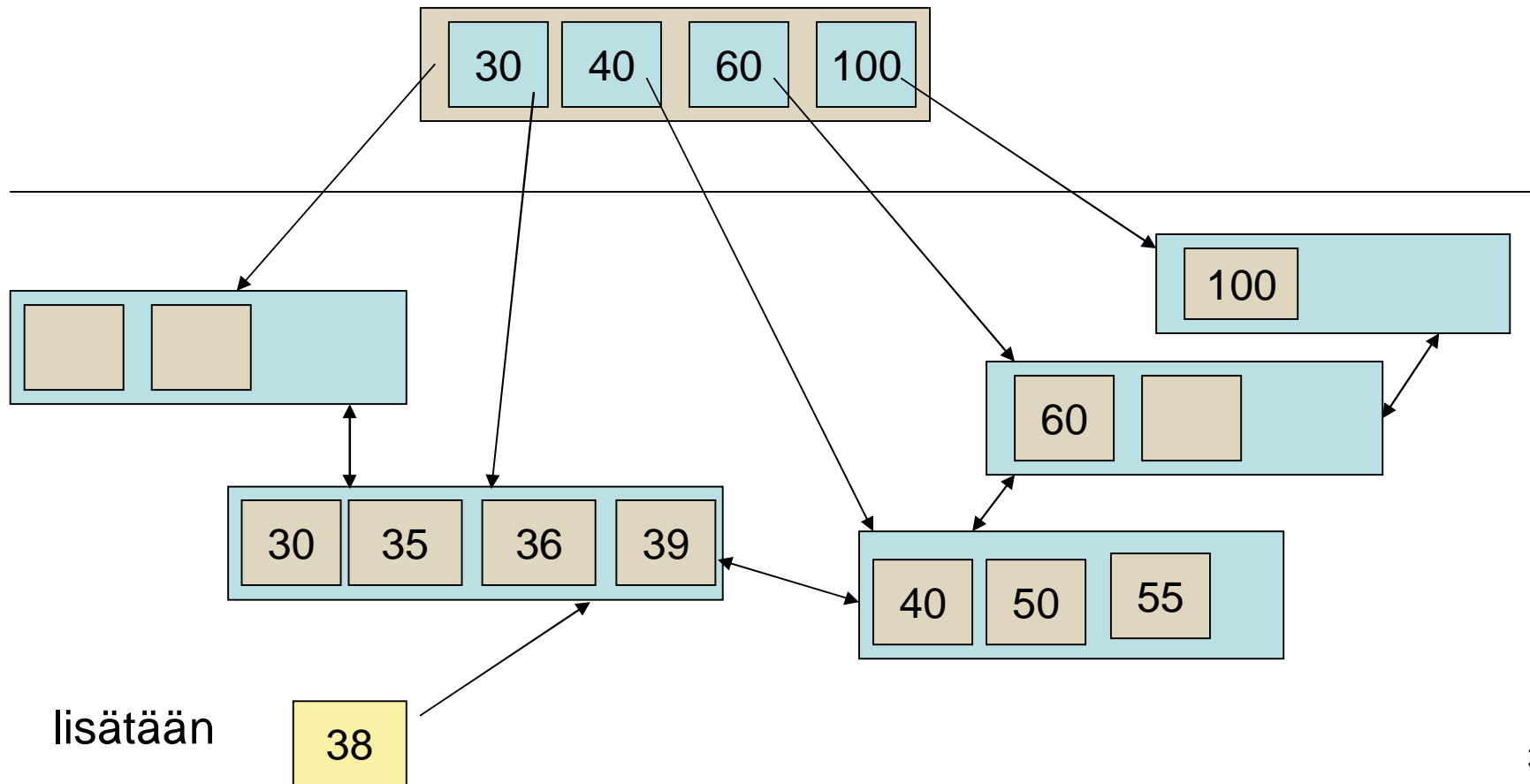
Tasapainotus lisäyksessä

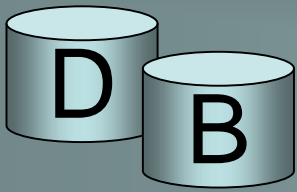
- Hakemistomerkinnän lisääminen hakemistosivulle voi aiheuttaa ylivuodon hakemistosivulla. Lisätään aluksi hakemistomerkintä oikealle kohdalleen.
- Sivulla olisi nyt $2d+1$ avainta ja $2d+2$ osoitetta
- Otetaan käyttöön uusi hakemistosivu X . Siirretään sinne avaimet $K_{d+2} \dots K_{2d+1}$ ja osoitteet $p_{d+1} \dots p_{2d+2}$
- Muodostetaan '*nouseva*' hakemistotietue keskimmäisestä avaimesta K_{d+1} ja uuden sivun osoitteesta X . Tämä lisätään siihen hakemistosivuun, jonka alaisuuteen jaettu hakemistosivu kuului. Jos jaettu sivu oli juuri, lisätään sen osoitteeksi p_0 edellisen juuren osoite.



Tasapainotus lisäyksessä

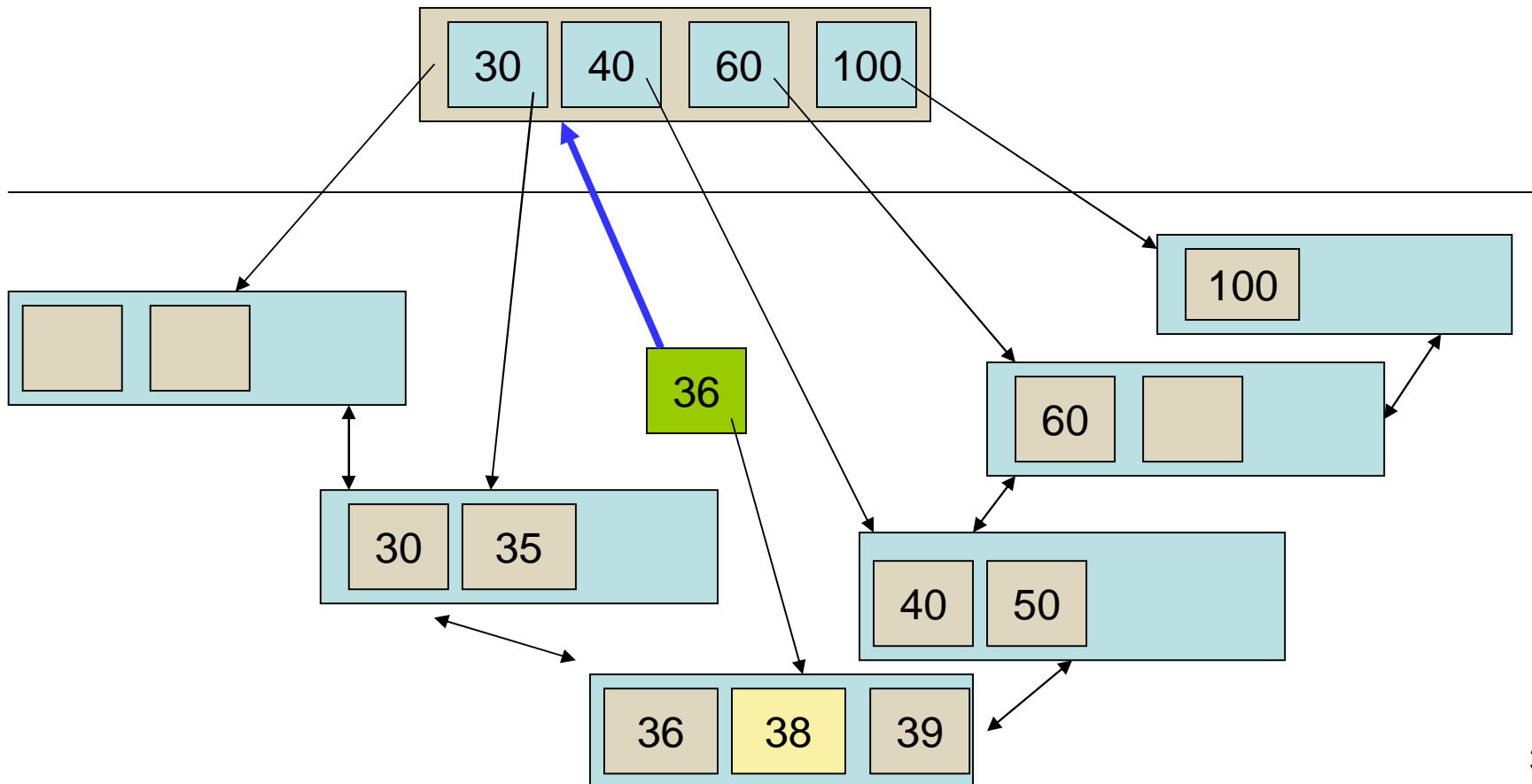
Rakenne on nyt hieman kasvanut,
kyseessä kertaluvun 2 B+- puu

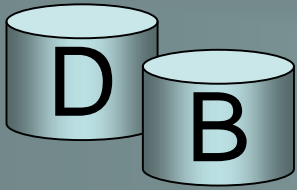




Tasapainotus lisäyksessä

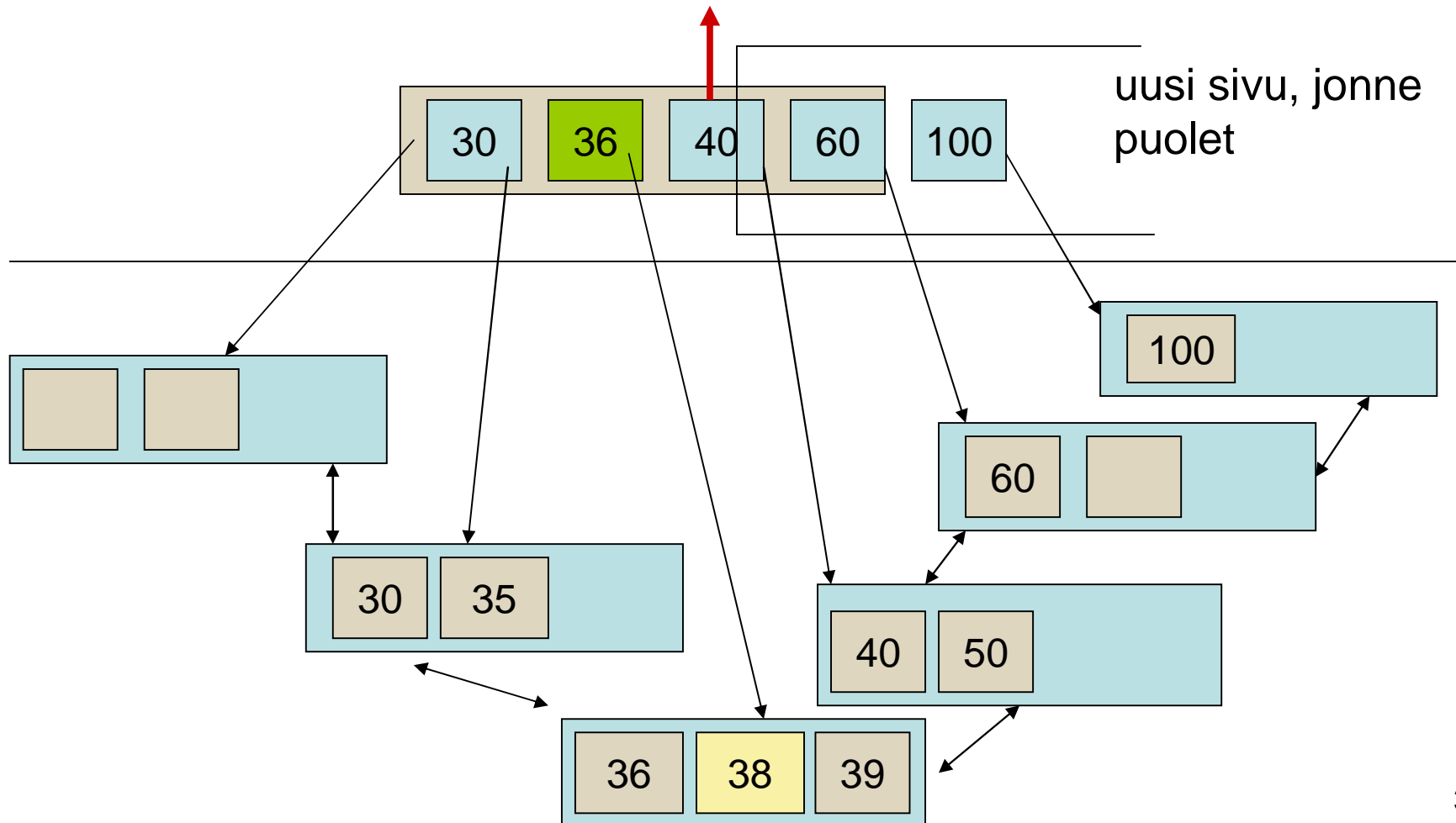
'nouseva' hakemistotietue ei mahdu hakemistosivulle

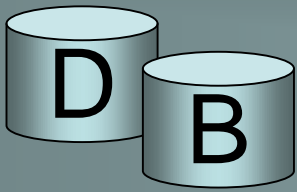




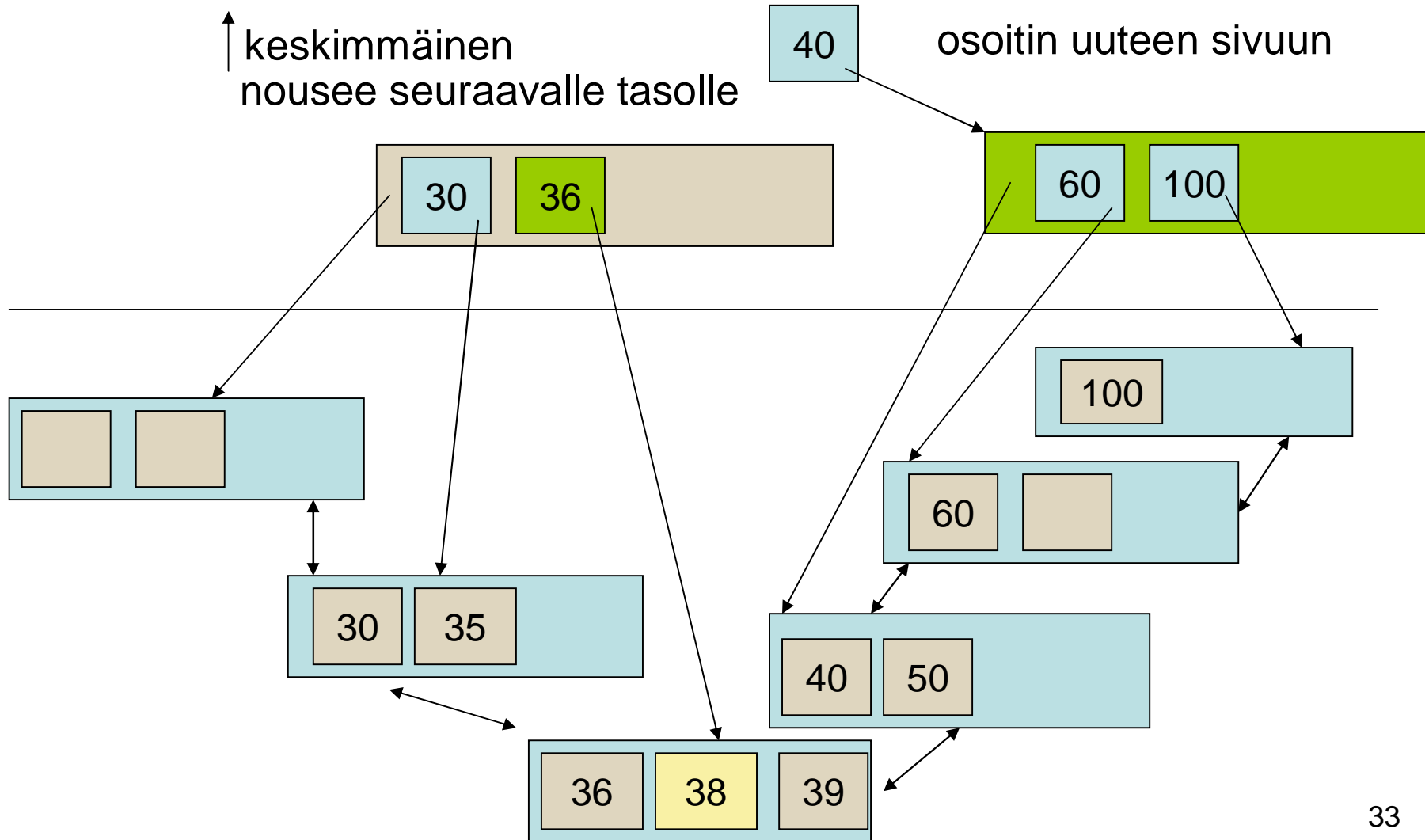
Tasapainotus lisäyksessä

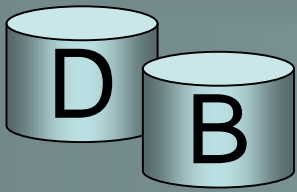
keskimmäinen avain nousee ylöspäin





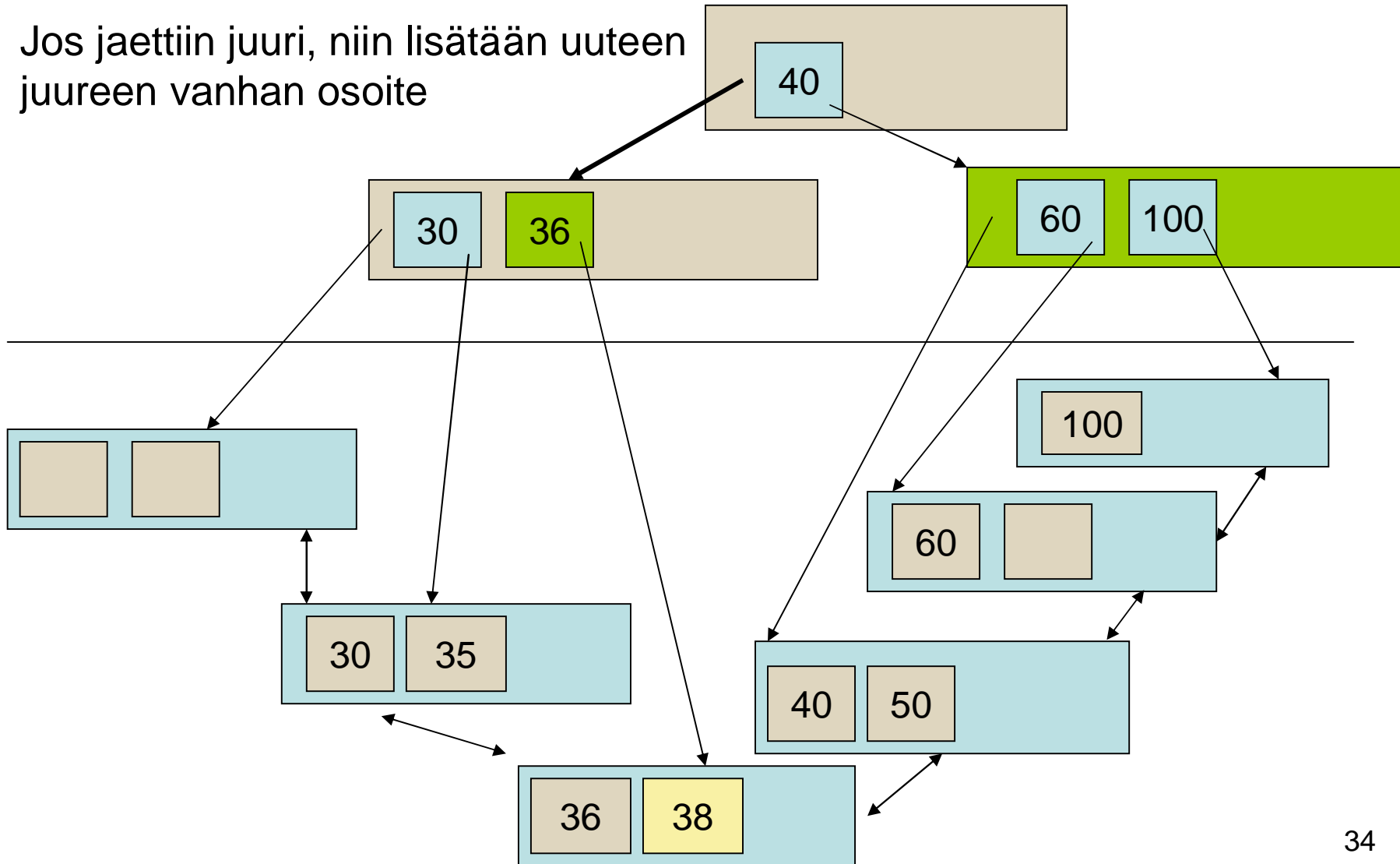
Tasapainotus lisäyksessä

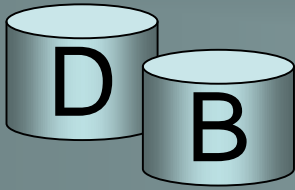




Tasapainotus lisäyksessä

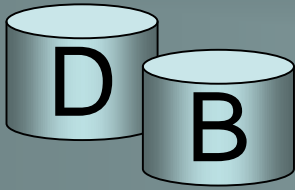
Jos jaettiin juuri, niin lisätään uuteen juureen vanhan osoite





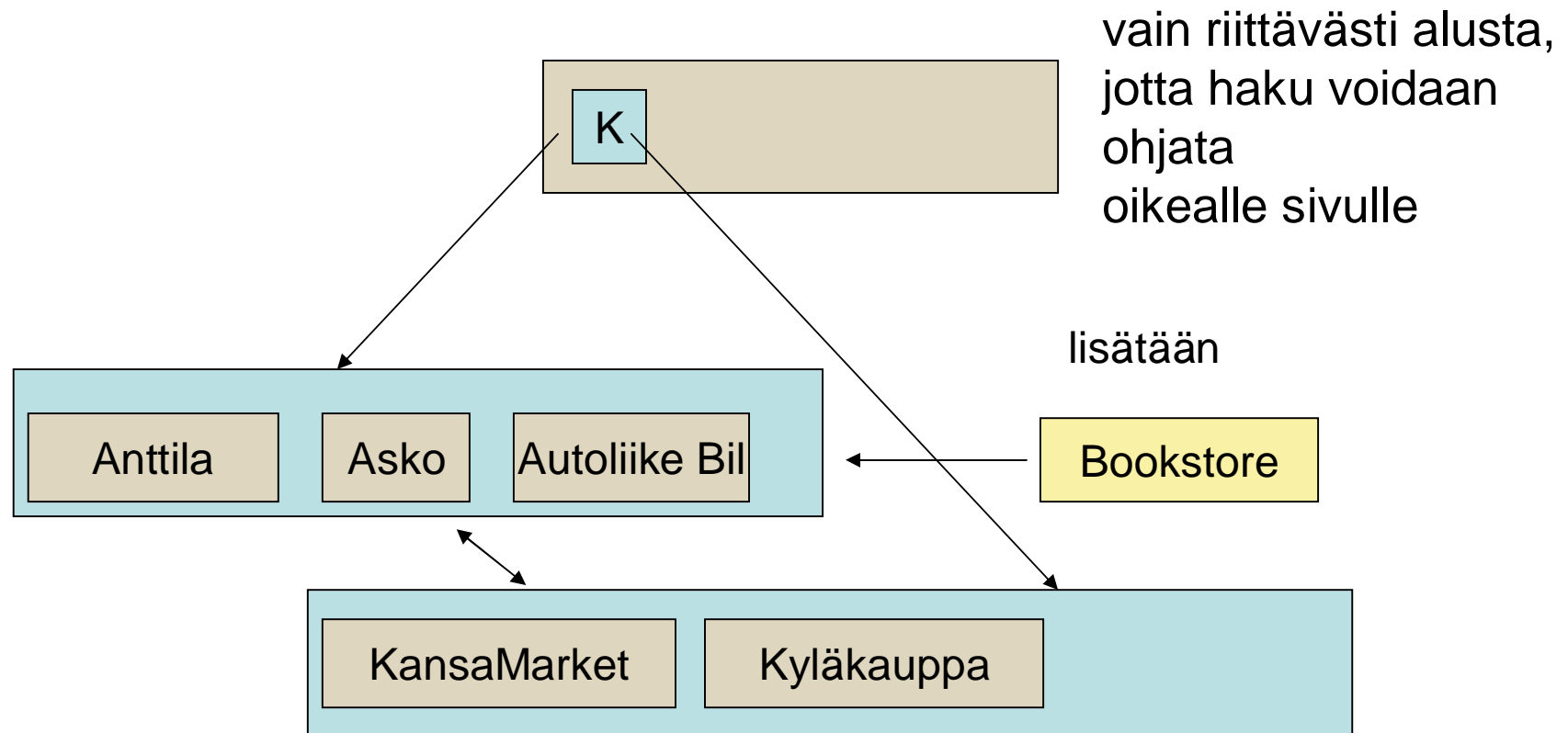
Tasapainotus lisäyksessä

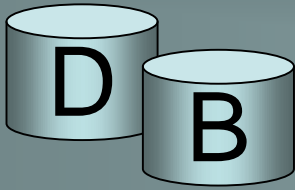
- Edellä kuvatulla tasapainotuksella hakemistosivuilla on aina vähintään d hakemistotietuetta (ei juuressa).
- Käytännössä avaimet voivat olla vaihtuvapituisia ja jakokriteeri voisi olla puolet käytettävissä olevasta tilasta avainten lukumäärän sijasta.
- Mitä leveämpi hakupuu on, sitä matalampi se on ja sitä vähemmän levyhakuja tarvitaan sivun löytämiseksi. Puun leveyteen vaikuttaa se montako hakemistomerkinää mahtuu hakemistosivulle. Useissa järjestelmissä ei tallennetakaan hakemistoon täydellisiä avaimia vaan pelkästään riittävän pitkiä osia avaimen alusta ohjaamaan hakuja



Erottimet lyhyempiä kuin avaimet

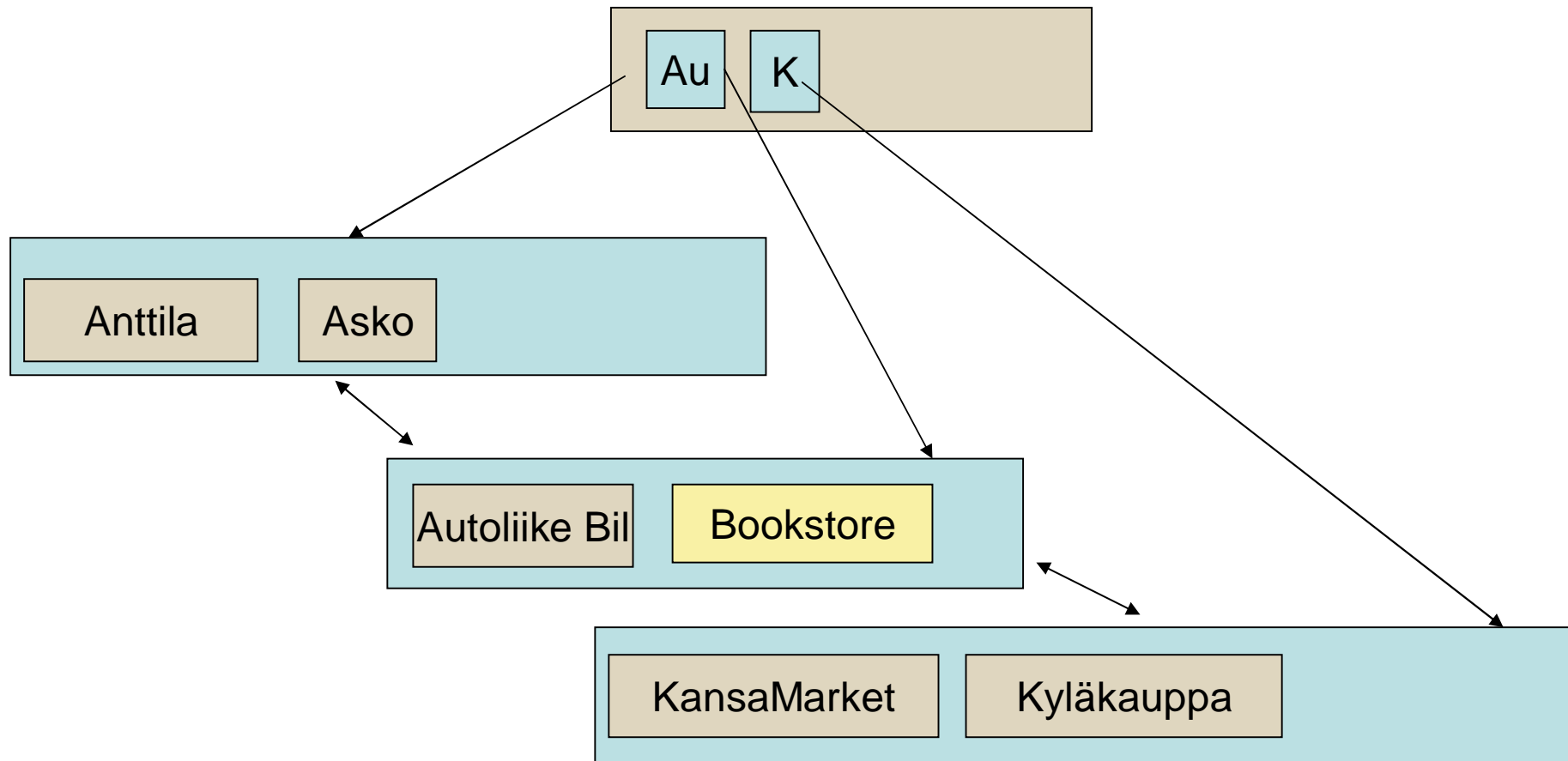
Alkuosa erottelu - Prefix key compression

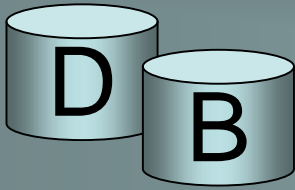




Erottimet lyhyempiä kuin avaimet

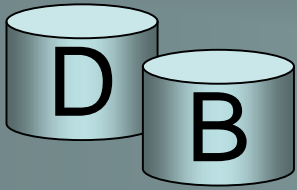
Alkuosa erottelu - Prefix key compression



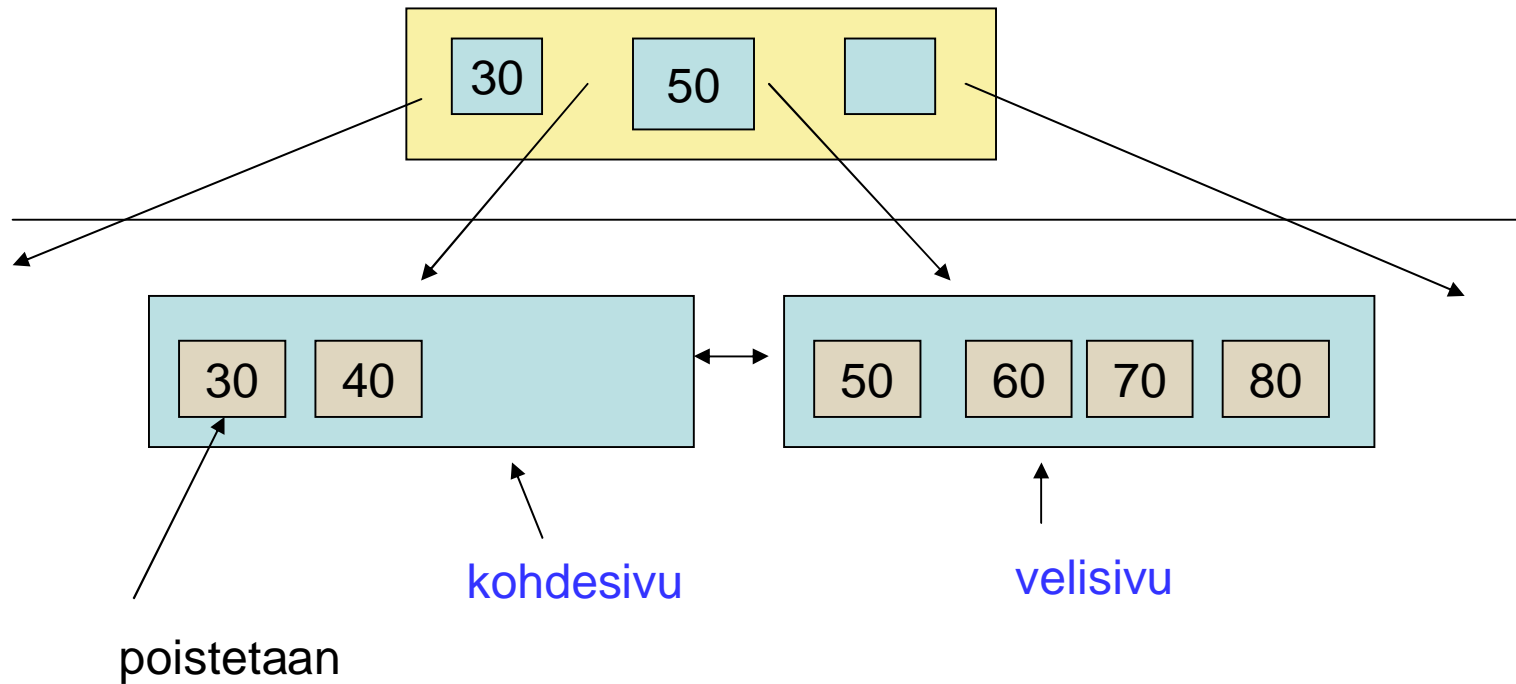


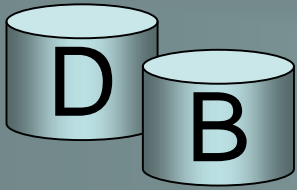
B+ -puun tasapainotus poistossa

- Poistot
 - Alkuperäisen B+ -puun idean mukaisesti tasapainotusta tehdään myös poistossa
 - Jos datasisivun täyttösuhde laskee alle puoleen ja sivun ja sen velisivun (**sibling**, saman isäsivun alla oleva vierussivu) yhteenlaskettu tietomäärä ylittää ylittää sivukoon, järjestetään sivuparin tietueet uudelleen siirtämällä täydemmältä sivulta tietue vajaalle sivulle.



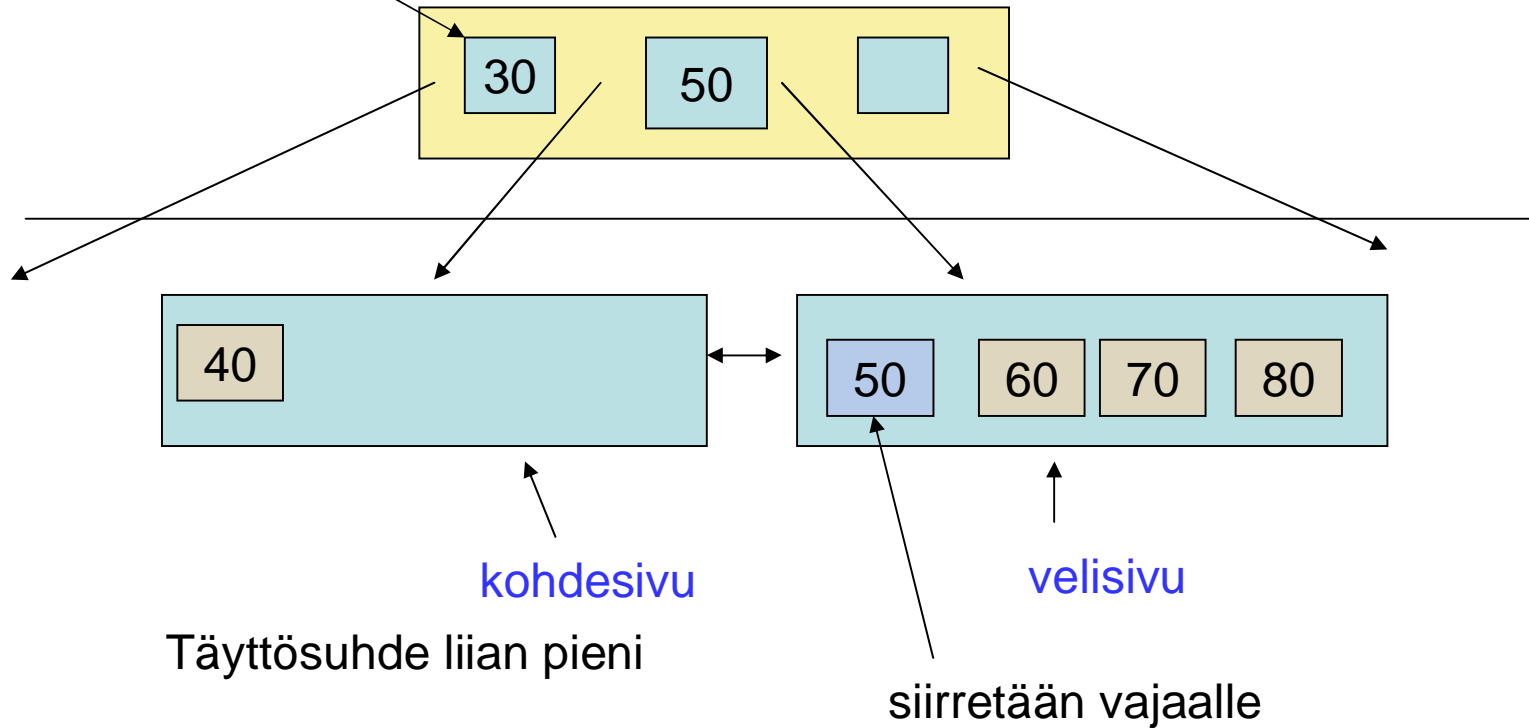
B+ -puun tasapainotus poistossa

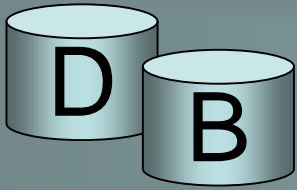




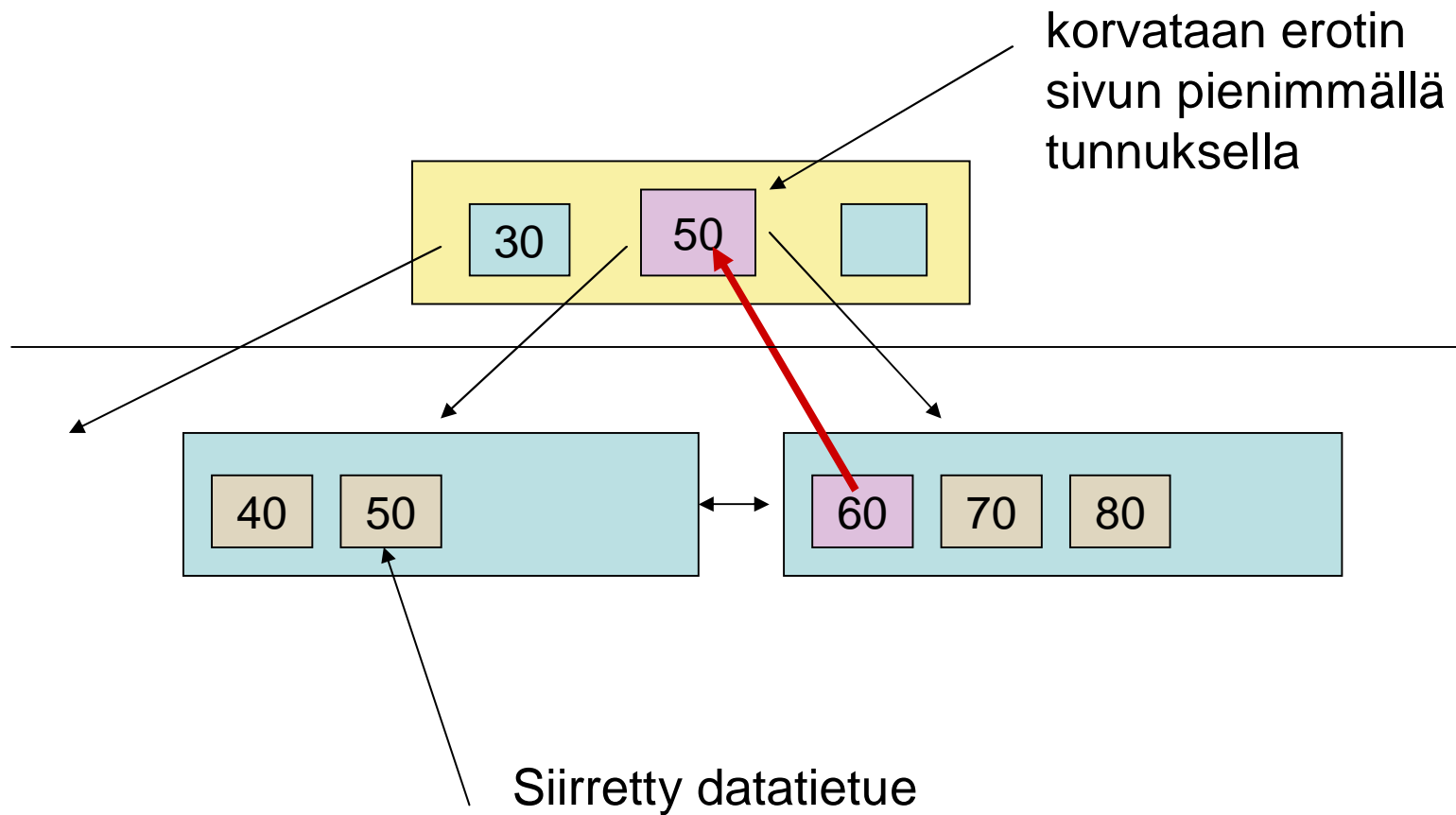
B+ -puun tasapainotus poistossa

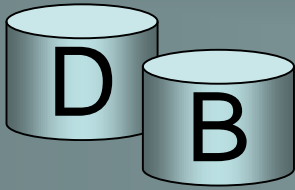
tätä ei tarvitse
vaihtaa vaikka
tietue katosikin



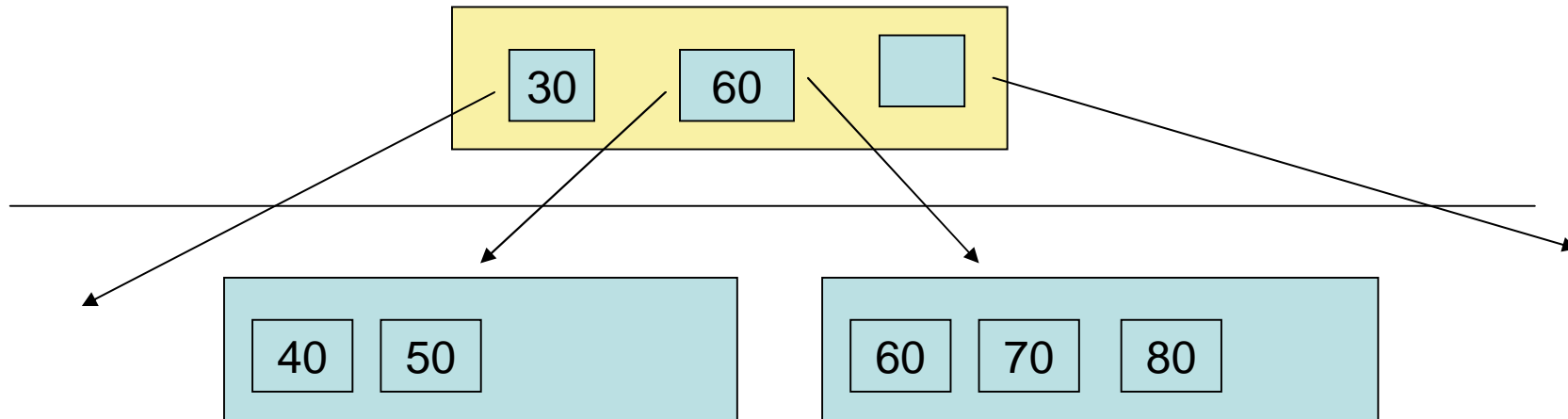


B+ -puun tasapainotus poistossa

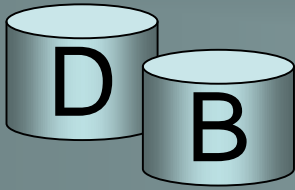




B+ -puun tasapainotus poistossa

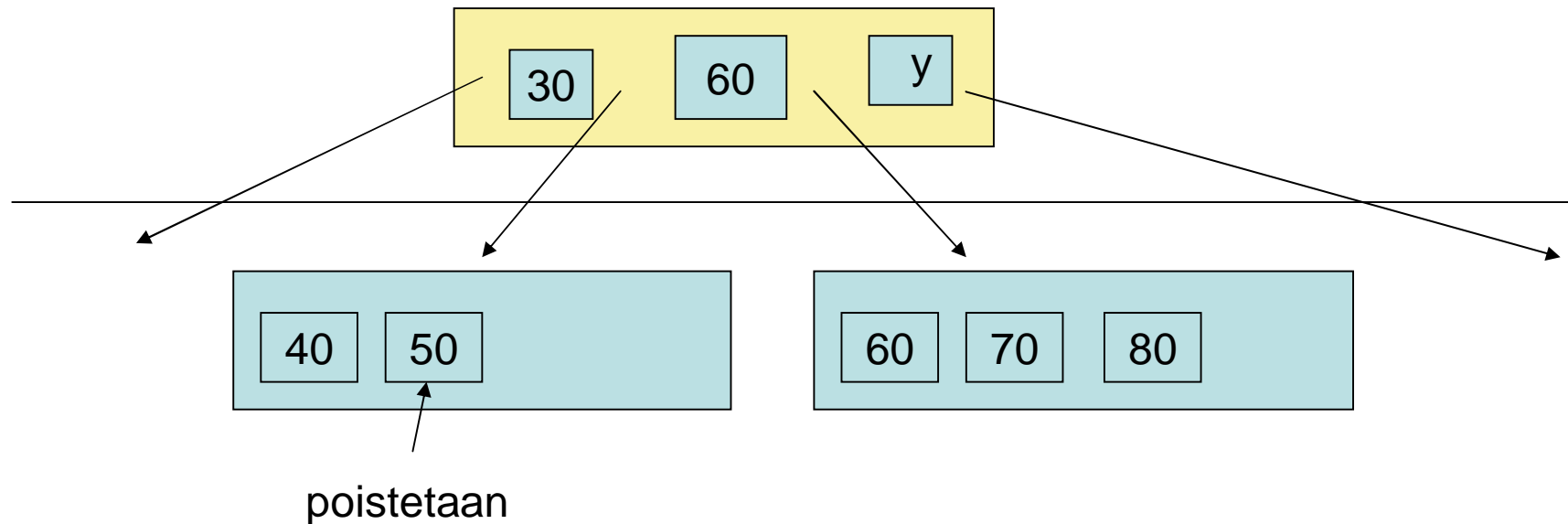


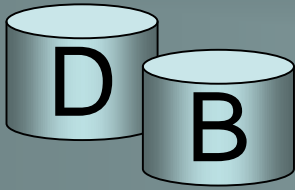
rakenne tasauksen jälkeen



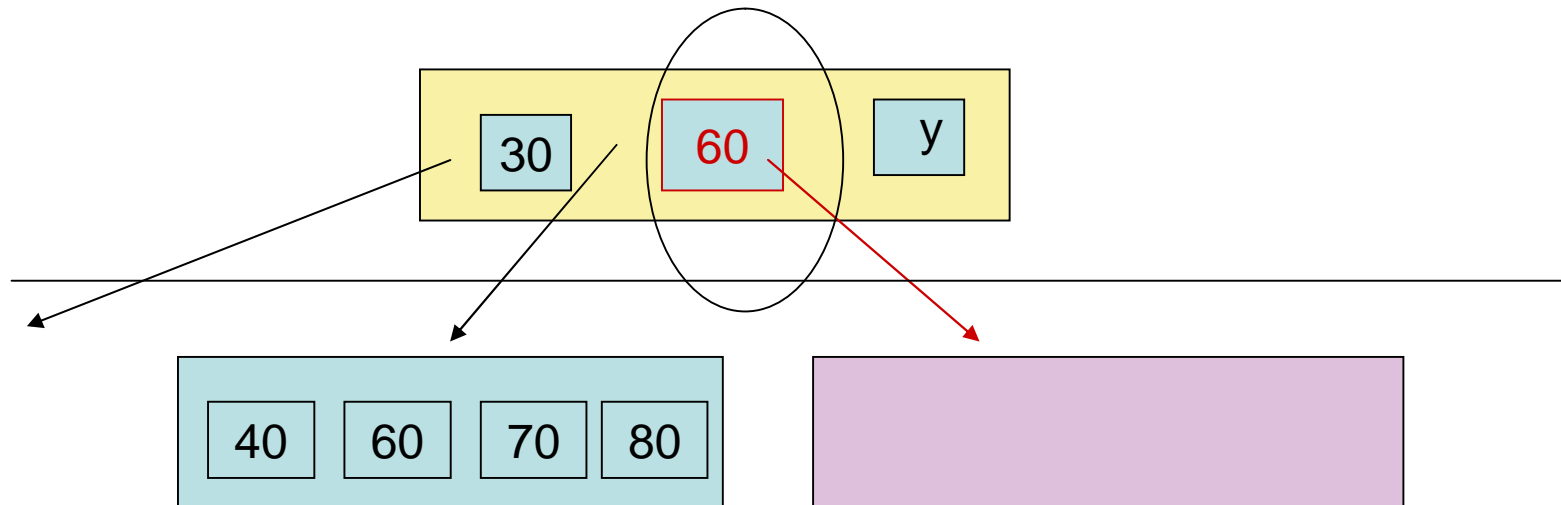
B+ -puun tasapainotus poistossa

- Jos poiston kohteena olevan sivun ja sen velisivun yhteenlaskettu tietomäärä jää alle sivun kapasiteetin yhdistetään sivut ja poistetaan niiden välinen erotin hakemistosta.

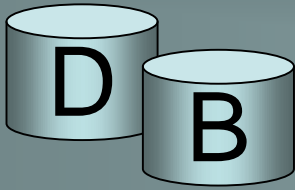




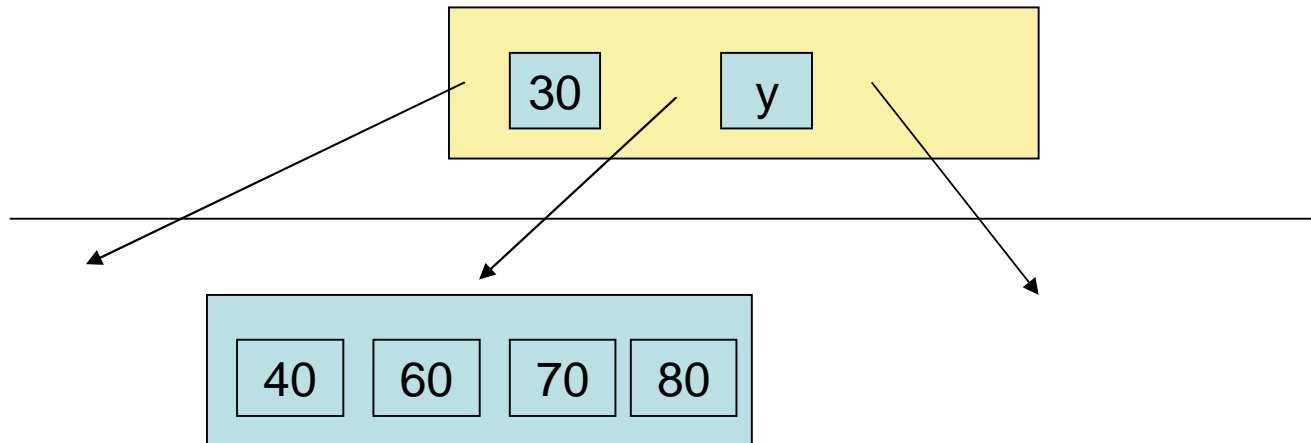
B+ -puun tasapainotus poistossa

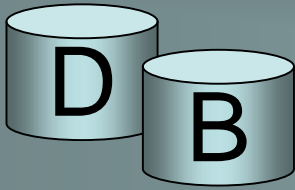


siirretään kaikki velisivun tietueet kohdesivulle,
vapautetaan tyhjentyneet sivu,
poistetaan erotin hakemistosivulta



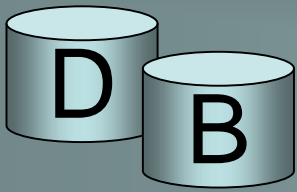
B+ -puun tasapainotus poistossa



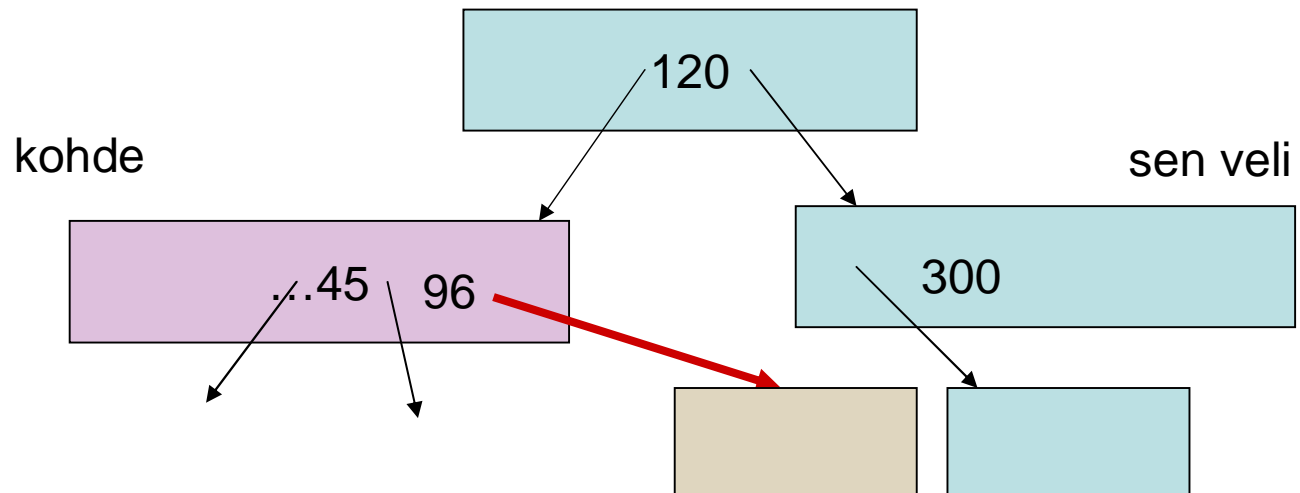
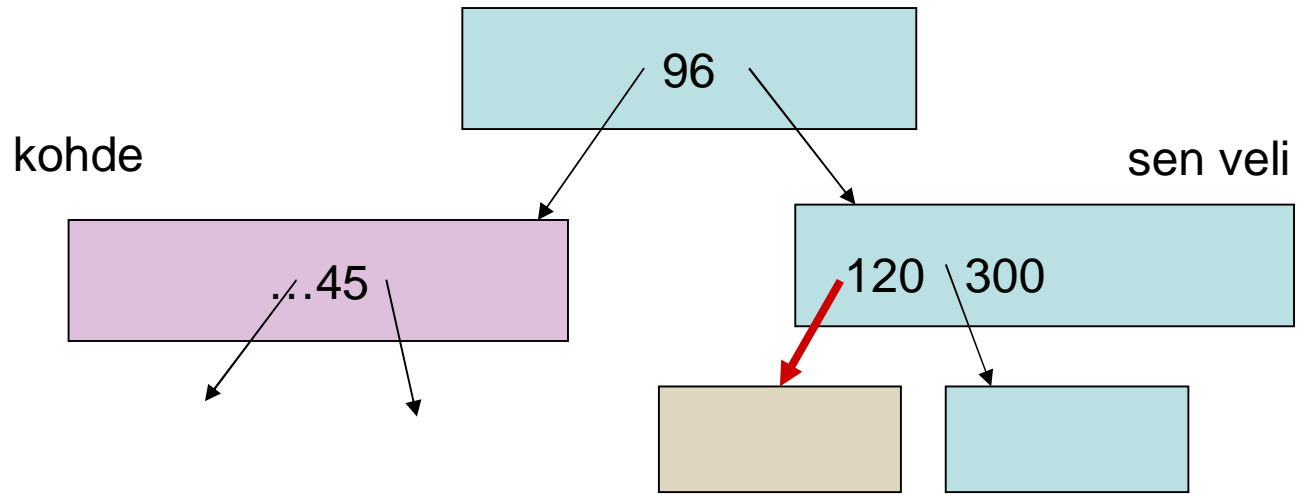


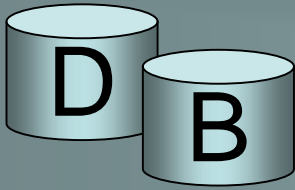
B+ -puun tasapainotus poistossa

- Hakemistomerkin poiston takia hakemistosolmu voi jäädä vajaatäyttöiseksi
- Sivun täyttösuhdetta korjataan samalla periaatteella kuin datasivujenkin eli siirtämällä velisolmusta täydennystietue, **siirto tehdään isäsolmun kautta vyöryttämällä**
- **Vyörytys:** Olkoon kyseessä oikeanpuoleinen veli eli velisolmussa on isompia avaimia. Vyörytys tapahtuu tällöin seuraavasti
 - Lisätään isäsolmusta kopioitu kohdesolmun ja velisolmun välinen avain kohdesolmun avainlistan loppuun
 - Lisätään velisolmun ensimmäinen osoitin p_0 kohdesolmun osoitinlistan loppuun
 - Korvataan isäsolmussa ollut kohdesolmun ja velisolmun välinen erotin velisolmun erotinlistan pienimmällä avaimella
 - Poistetaan velisolmun avain- ja osoitinlistojen alkupäästä ensimmäinen avain ja osoitin
 - Sivuja yhdistettäessä vyörytetään kaikki velisolmun tietueet kohdesolmuun ja poistetaan isäsolmusta erotin sekä osoitin velisolmuun
- Puun korkeus alenee, jos poistettavana on juurisolmun ainoa erotin.



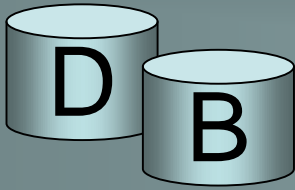
Vyörytys





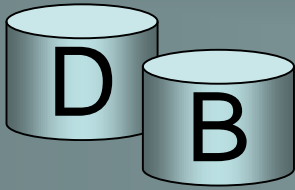
B+ -puun tasapainotus poistossa

- Kaikki toteutukset eivät käytä tasapainotusta poistojen yhteydessä, vaan antavat täyttöasteen laskea.
- B+ puun hakemistosivujen keskimääräiseksi täyttöasteeksi muodostuu lisäys- ja poistotasapainotusta käytettäessä noin 67%



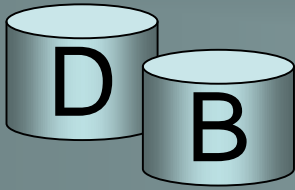
B+ -puun ominaisuuksia

- Olkoon B+ -puun korkeus h
 - haettaessa indeksointiavaimen perusteella joudutaan tutkimaan h sivua ($\leq h$ levyhakua)
 - lisättäessä joudutaan lukemaan h sivua ja kirjoittamaan vähintään 1 ja enintään $h+2$ sivua
 - poistettaessa joudutaan lukemaan vähintään h ja enintään $2h-1$ ja kirjoittamaan vähintään 1 ja enintään $2h-1$ sivua
 - Indeksointiavaimen arvon muuttaminen täytyy B+ -puun yhteydessä hoitaa poisto ja lisäys operaatioina.



B+ -puut tietokantatoteutuksessa

- Useimmat tietokannan hallintajärjestelmät mahdollistavat B+ -puun käytön joko
 - **taulun toteutusrakenteena**
 - datasivuilla on taulun rivejä vastaavia tietueita
 - ratkaisun huono puoli on se, että rivit voivat vaihtaa sivua, mikä tekee oheishakemistojen käytön työlääksi,
 - *kaikki osoitteet sivua vaihtaneeseen tietueeseen on vaihdon yhteydessä päivitettävä*
 - *Oraclessa tämä on ratkaistu siten, että B+ -puu rakenteisen tiedoston oheishakemisto ei käytäkään osoittamiseen tietueen osoitetta vaan tietueen pääavainta (haku hidastuu)*
 - **tiheän hakemiston toteutusrakenteena**
 - datasivuilla on tiheän hakemiston hakemistomerkitöjä
 - tämä on yleisin B+ -puun käyttötilanne.
 - taulun rivit ovat tällöin yleensä kasararakenteena



B+ -puu

- Edellä käsitellyt B+ -puun haku- ja lisäys-toiminnot perustuvat indeksointiavaimen yksikäsitteisyyteen
 - tietyllä avaimella varustettu tietue on tietyllä sivulla.
- Rakennetta voi kuitenkin käyttää myös toistuvien avainarvojen yhteydessä (indeksointi vaikkapa henkilön nimen perusteella)
 - vaihtoehto: datasivukohtaiset ylivuotolistat
 - yleisesti käytetty vaihtoehto (esim. Oracle, DB2):
Otetaan rivitunnus (rid) mukaan indeksointiavaimeseen
tällöin avaimet ovat aina yksikäsitteiset