



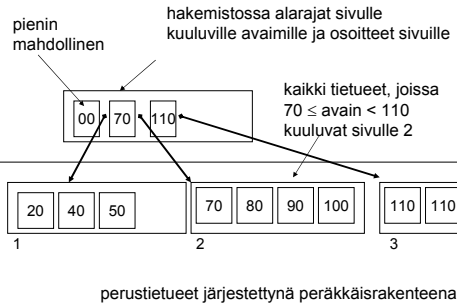
## Harvat hakemistot

- 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

1



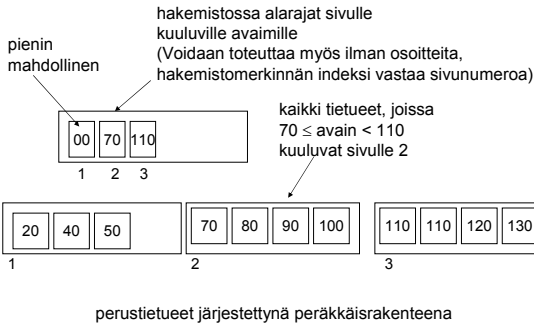
## Harvat hakemistot



2



## Harvat hakemistot



3



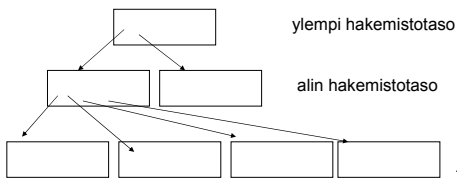
## Harvat hakemistot

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

4



## Harvat hakemistot



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

5



## 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 datasisivulla, etsi tietue sivulta

6



## Harvat hakemistot

- Jos datasisivuille mahtuu useita tietueita muodostuu harva hakemisto tiheää pienemmäksi, joten haku on nopeampaa
- Esimerkki
  - työntekijätietueen keskipituus oli 300, joten yhdelle datasisivulle voitaisiin sijoittaa vaikkapa 10 tietuetta (jätetty hieman tilaa lisäyksille)
  - Tiedosto mahtuu 800 sivulle
  - Kunkin sivun pienin avain viedään 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

7



## 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%')

8



## 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

9



## 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

10



## Harvat hakemistot - ISAM

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

11



## 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

12



## Harvat hakemistot - ISAM

### • Poistot

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

13



## 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.

14



## 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, pahin tapaus levyhakuja luokkaa:  
lisäysten määrä/sivukoko

15



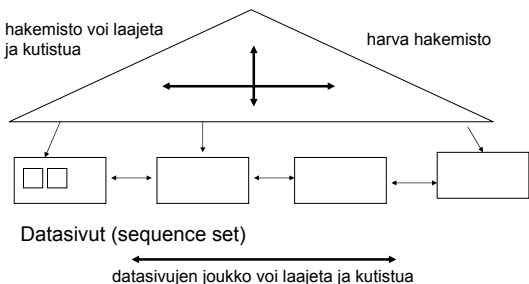
## Dynaamiset hakemistorakenteet B+ -puu

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

16



## Dynaamiset hakemistorakenteet B+ -puu

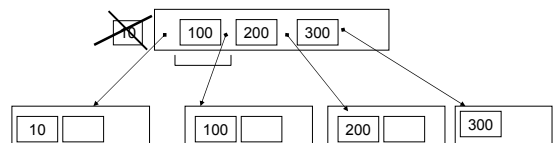


17



## Dynaamiset hakemistorakenteet B+ -puu

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



18



## Dynaamiset hakemistorakenteet B+ -puu

– hakemistosivut eivät ole staattisia, niihin voi tulla uusia hakemistomerkinthöjä, merkinthöjä 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 puskureissa)
- lisäys- ja poisto-operaatiot säilyttävät hakurakenteen tasapainon

19



## Dynaamiset hakemistorakenteet B+ -puu

- Tarkastellaan kertaluokan (order) d B+ -puuta
- Tällaisen puun jokaisessa hakemistosolmussa paitsi juuressa (ylin taso) on  $d \leq m \leq 2d$  hakemistomerkinthöjä 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 avain  $< K_1$

20



## 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 datasisivun osoitteen

21



## Dynaamiset hakemistorakenteet B+ -puu

- Lisäykset:

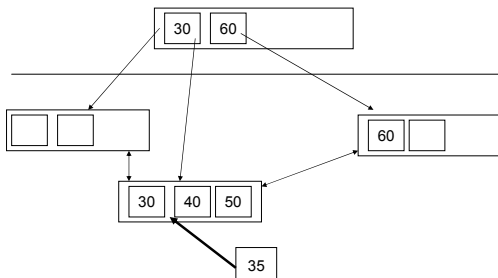
- Lisättäessä tietuetta B+ -puuhun sille haetaan sijoituspaikka edellä kuvatulla hakuoperaatiolla
- Jos sivulla on tilaa, tietue lisätään järjestyksessä oikealle paikalleen sivulla
- Jos tietue ei mahdu sivulle se sijoitetaan loogisesti oikealle paikalleen ja jaetaan ylivuotanut sivu:
  - Otetaan käyttöön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

22



## Tasapainotus lisäyksessä

lisättävä ei mahdu sivulle

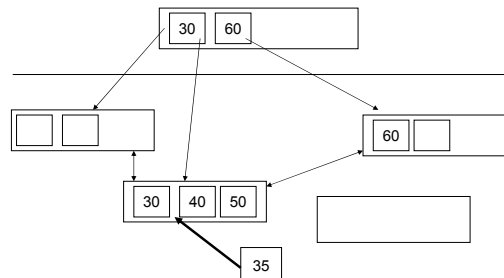


23



## Tasapainotus lisäyksessä

otetaan käyttöön uusi datasisivu

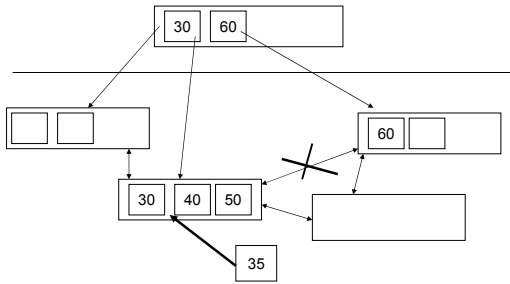


24



## Tasapainotus lisäyksessä

kytketään uusi datasisivutietuun

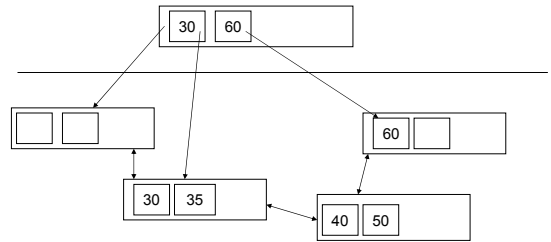


25



## Tasapainotus lisäyksessä

Siirretään puolet ylivoimavan sivun sisällöstä uudelle sivulle

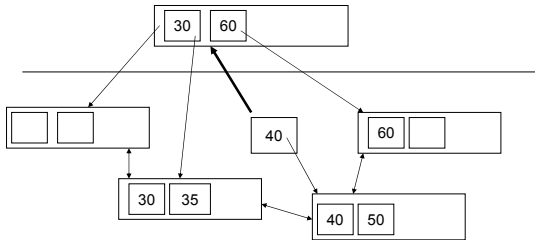


26



## Tasapainotus lisäyksessä

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

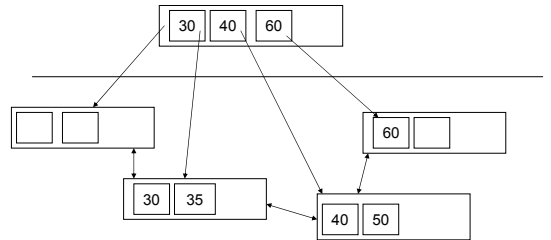


27



## Tasapainotus lisäyksessä

Hakemistotietue mahtui sivulle – lisäys on valmis



28



## Tasapainotus lisäyksessä

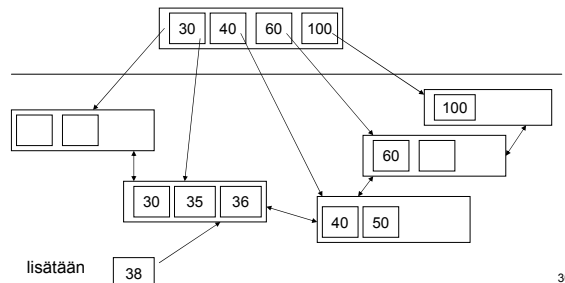
- Hakemistomerkin lisääminen hakemistosivulle voi aiheuttaa ylivoimavan hakemistosivulla. Lisätään aluksi hakemistomerkin oikealle kohdalleen.
- Sivulla olisi nyt  $2d+1$  avainta ja  $2d+2$  osoitetta
- Otetaan käyttöön uusi hakemistosivu X. Siirretään sinne avaimet  $K_{d+2} \dots K_{2d+1}$  ja osoitteet  $p_{d+1} \dots p_{2d+2}$
- Muodostetaan 'nouseva' hakemistotietue keskimmaisesta 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.

29



## Tasapainotus lisäyksessä

Rakenne on nyt hieman kasvanut, kyseessä kertaluvun  $2B+1$  -puu

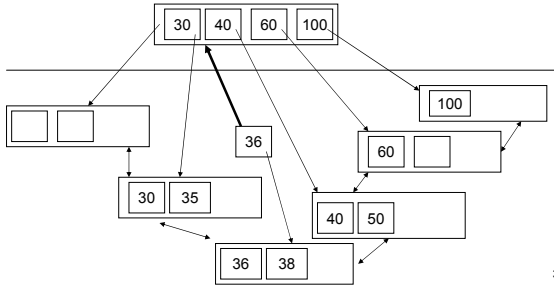


30



## Tasapainotus lisäyksessä

'nouseva' hakemistotietue ei mahdu hakemistosivulle

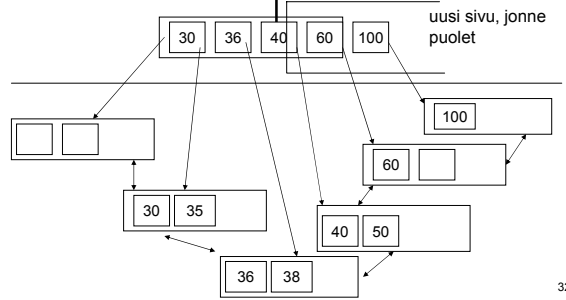


31



## Tasapainotus lisäyksessä

keskimmäinen avain nousee ylöspäin

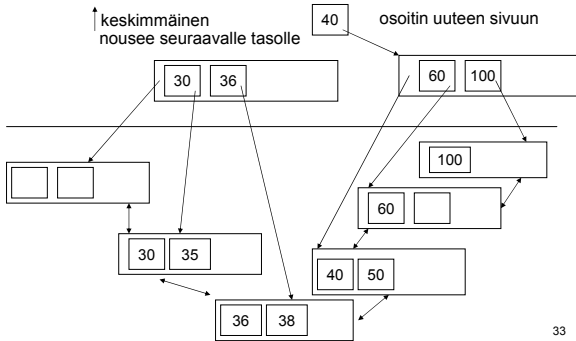


32



## Tasapainotus lisäyksessä

↑ keskimmäinen nousee seuraavalle tasolle

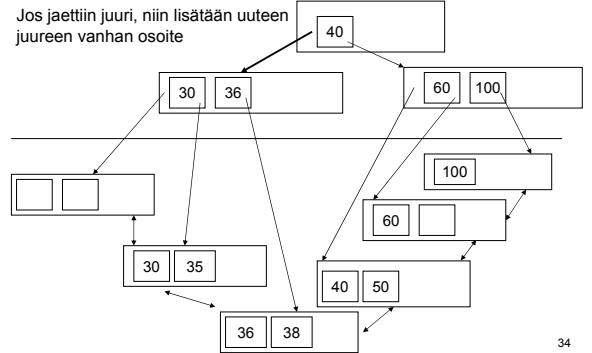


33



## Tasapainotus lisäyksessä

Jos jaettiin juuri, niin lisätään uuteen juureen vanhan osoite



34



## Tasapainotus lisäyksessä

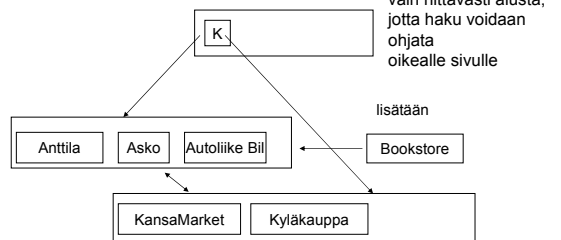
- Edellä kuvatulla tasapainotuksella hakemistosivuilla on aina vähintään d hakemistotietuetta.
- Käytännössä avaimet voivat olla vaihtuvapituisia ja jakokriteeri voisi olla puolet käytettävissä olevasta tilasta avainten lukumäärän sijasta.
- Mitä leveämpi hakupuu on, sitä matalampi se on ja sitä vähemmän 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

35



## Erottimet lyhyempiä kuin avaimet

Alkuosa erottelu - Prefix key compression



36



# Erottimet lyhyempiä kuin avaimet

Alkuosa erottelu - Prefix key compression

