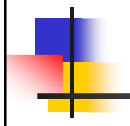
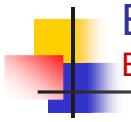


BM 307 Dosya Organizasyonu (File Organization)

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Konular

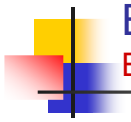
- B-Tree and Derivatives
  - B-Trees
  - B#-Trees
  - B+-Trees
  - Değerlendirme



B-Tree and Derivatives

B-Trees

- B-Tree sequential ve direct erişimde iyi performansa sahiptir.
- Binary tree'lerde branching factor ikiden büyük olamaz. B-Tree'lerde teorik olarak limit yoktur.
- B-Tree'ler multiway search tree'dir ve bir node'da çok sayıda kayıt tutulabilir
- B-Tree'ler diğer ağaçlardan farklı olarak bottom-up yapılandırılırlar.
- Herhangi bir döndürmeye gerek kalmadan (AVL tree ve IPR tree'deki gibi) otomatik olarak balance edilirler.



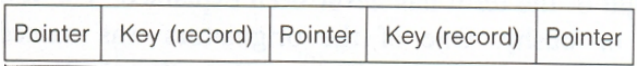
B-Tree and Derivatives

B-Trees

- B-tree aşağıdaki kriterlere göre oluşturulur:

1- $d \leq keys \leq 2d$ (root node için $1 \leq keys \leq 2d$)
$d+1 \leq keys \leq 2d+1$ (root node için $2 \leq pointers \leq 2d+1$)

2- Bütün yaprak node'lar aynı seviyededir.



B-Tree and Derivatives

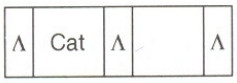
B-Trees

- Örnek

d = 1 (capacity order)

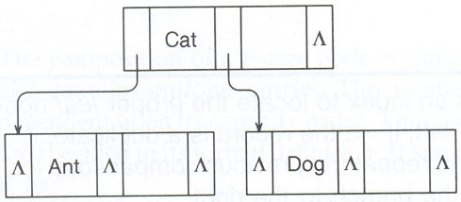
cat, ant, dog, cow, rat, pig, gnu

Cat insert edildi



Ant insert edildi

Dog insert edildi



B-Tree and Derivatives

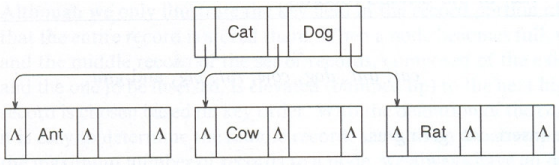
B-Trees

- Örnek

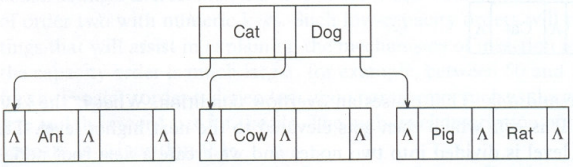
d = 1 (capacity order)

cat, ant, dog, cow, rat, pig, gnu

Cow ve Rat insert edildi



Pig insert edildi



B-Tree and Derivatives

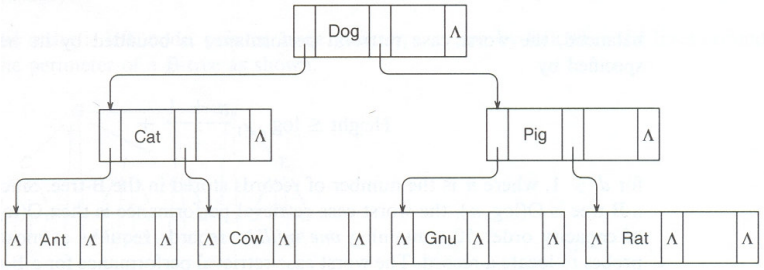
B-Trees

- Örnek

d = 1 (capacity order)

cat, ant, dog, cow, rat, pig, gnu

Gnu insert edildi



Packing factor = depolanan kayıt sayısı / kullanılan yer sayısı
= 7 / 14 = 50 %

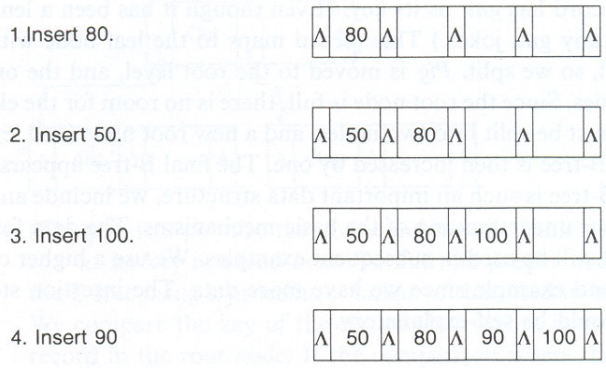
B-Tree and Derivatives

B-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



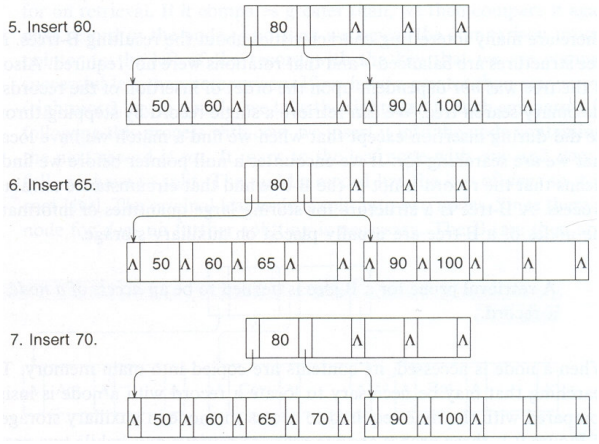
B-Tree and Derivatives

B-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



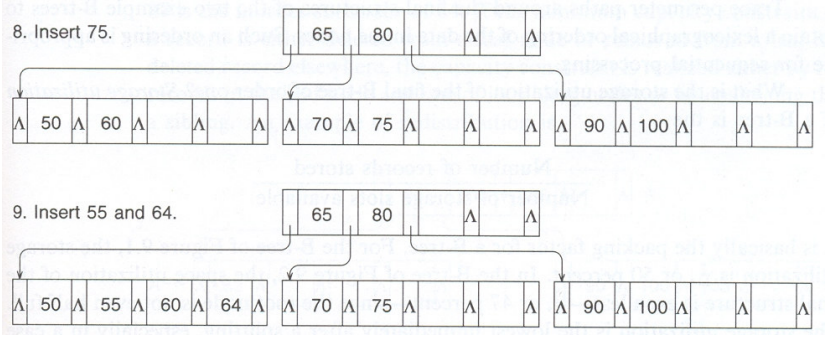
B-Tree and Derivatives

B-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



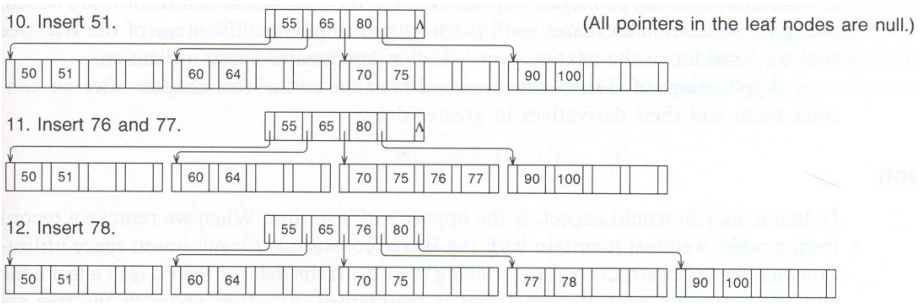
B-Tree and Derivatives

B-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



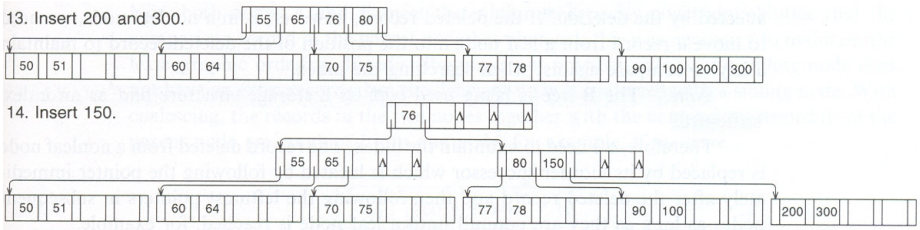
B-Tree and Derivatives

B-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



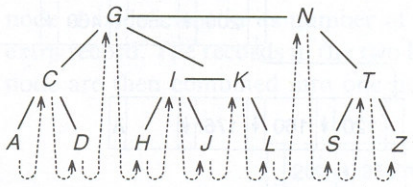
Packing factor = depolanan kayıt sayısı / kullanılan yer sayısı
= 17 / 36 = 47 %

B-Tree and Derivatives
B-Trees

- Değerlendirme
  - B-Tree'lerde retrieval probe kayda ulaşma değil node'a ulaşmadır
  - Ağacın yüksekliği aşağıdaki gibi ifade edilir

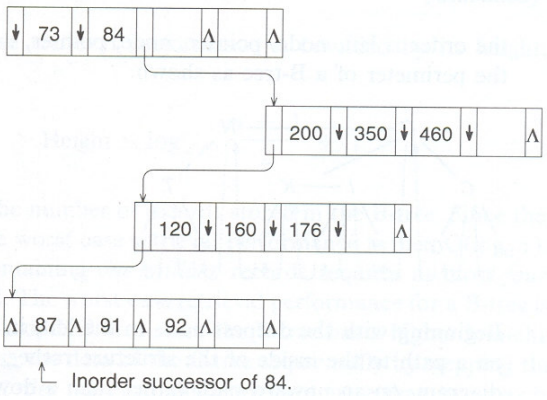
$$\text{Height} \leq \log_{d+1} \frac{n+1}{2} + 1$$

  - Worst case retrieval performansı $O(\log_d n)$ 'dir. n, B-tree'deki kayıt sayısıdır
  - Capacity order'ı 50 olan ve 1 milyon kayıt bulunduran B-tree'de bir kayda ulaşmak için en fazla 4 retrieval probe gerekir.
  - B-tree'deki kayıtlara sıralı ulaşmak için inorder dolaşma yapılır.



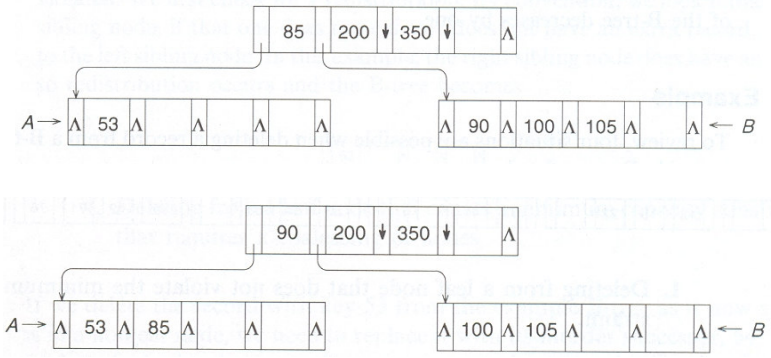
B-Tree and Derivatives
B-Trees

- Silme
  - Bir nonleaf node üzerinden kayıt silindiğinde inorder takipçisi yerine yazılır



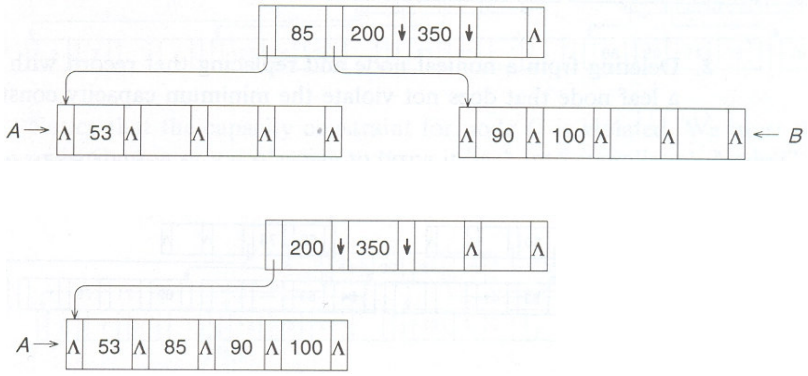
B-Tree and Derivatives
B-Trees

- Silme
  - Bir node'daki kayıt sayısı minimum capasite'den aşağı düşerse ve kardeş node'u fazla kayda sahipse, parent ve kardeş node ile yeniden düzenleme yapılır



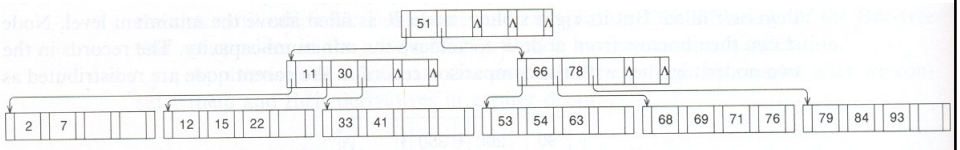
B-Tree and Derivatives
B-Trees

- Silme
  - İki kardeş node minimum kapasitenin altına düşerse ikisi ve parent node'daki kayıt birleştirilir

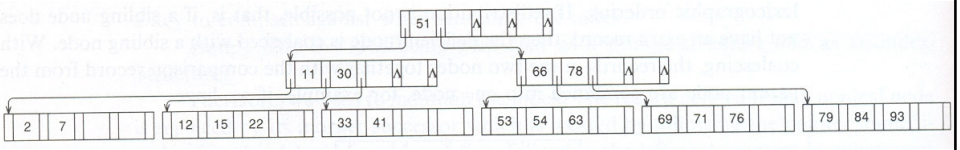


B-Tree and Derivatives
B-Trees

- Silme - Örnek
  - Minimum kapasitenin üstündeki yapraktan kayıt silinmesi

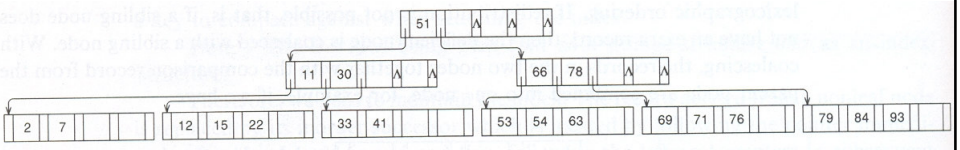


68 silindi

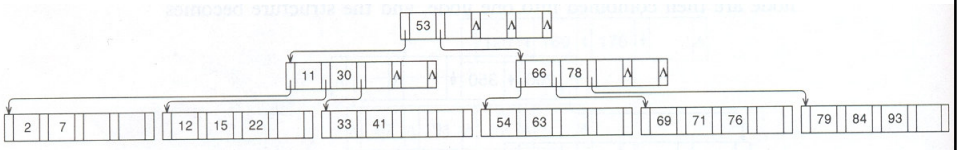


B-Tree and Derivatives
B-Trees

- Silme - Örnek
  - Bir nonleaf node'da kayıt silinmesi ve minimum kapasitenin üzerindeki nir node'dan kayıt aktarılması

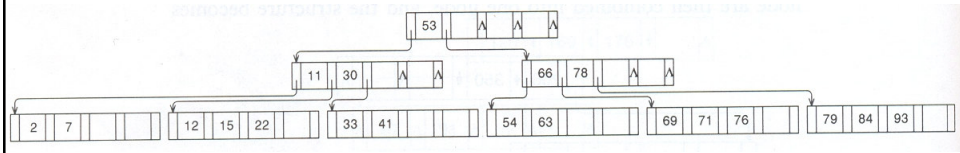


51 silindi

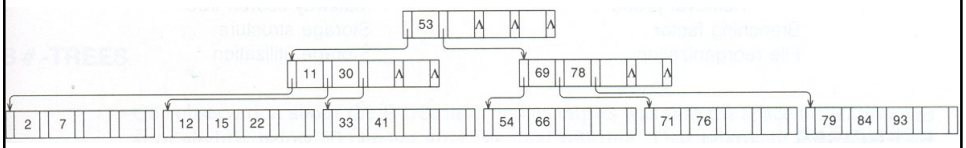


B-Tree and Derivatives
B-Trees

- Silme - Örnek
  - Bir leaf node'da kayıt silinmesi ve minimum kapasitenin altına düşülmesi

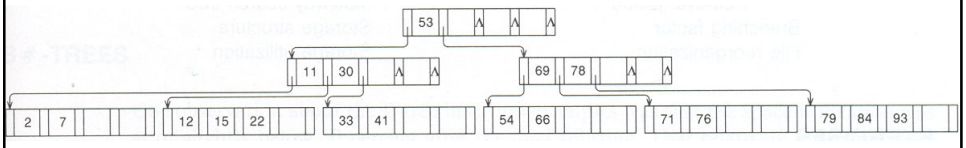


63 silindi

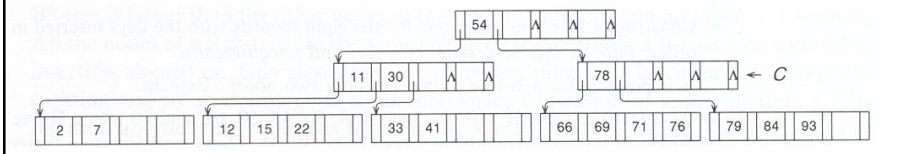


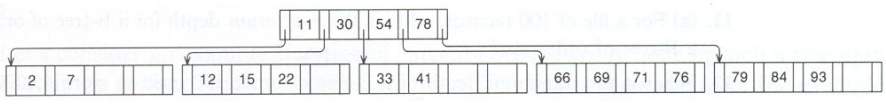
B-Tree and Derivatives
B-Trees

- Silme - Örnek
  - Bir leaf node'da kayıt silinmesi ve minimum kapasitenin altına düşülmesi ve node'ların birleştirilmesi



53 silindi





B-Tree and Derivatives

B#-Trees

- B-tree'lerde bir node dolunca bölme işlemi yapılmaktadır
- Bölme sonucunda oluşan iki node'da yarı yarıya doludur
- B#-tree'lerde bölme işlemi geciktirilerek node'ların doluluk oranı artırılır
- Ortalama Insert süresi uzar ve ağacın yüksekliği daha azdır.
- Tüm ağacın packing faktor değeri B-tree'lere göre daha yüksektir
- B# - tree'lerde retrieval performansı daha yüksektir
- Literatürde B*-tree şeklinde variantları vardır

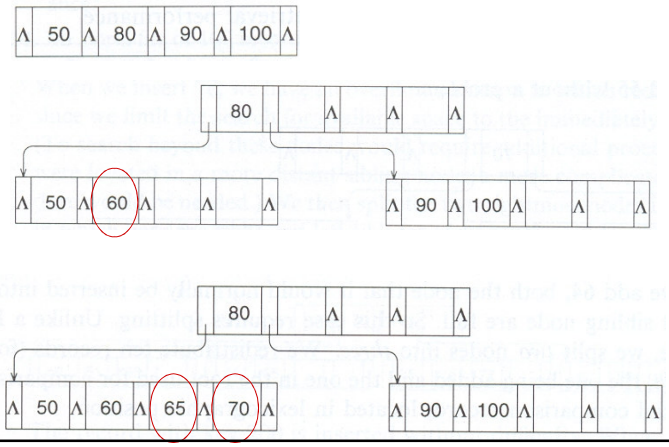
B-Tree and Derivatives

B#-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



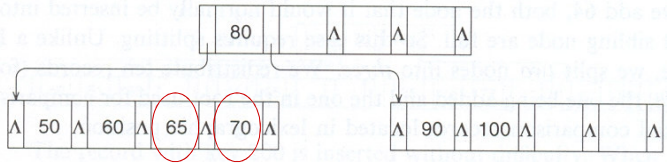
B-Tree and Derivatives

B#-Trees

- Örnek

d = 2 (capacity order)

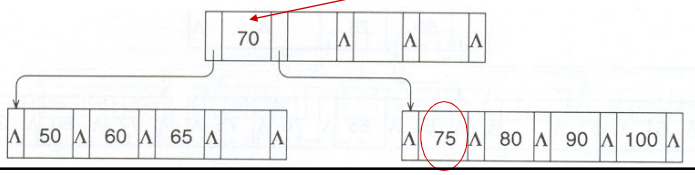
80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



$$\text{comparison.record} = \lfloor (r + 1) / n \rfloor$$

$$\lfloor 9/2 \rfloor = 4$$

r, dağıtılacak toplam kayıt sayısı



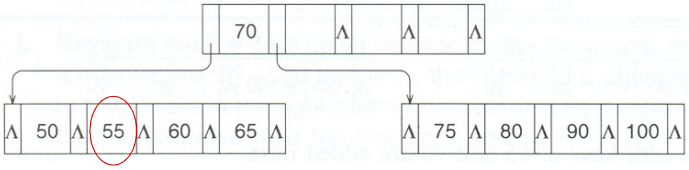
B-Tree and Derivatives

B#-Trees

- Örnek

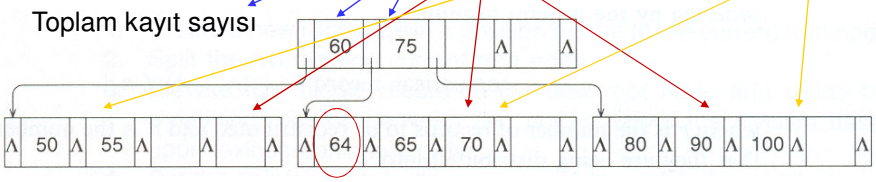
d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150



$\lfloor (10 - 2) / 3 \rfloor = 2$ $\quad \lfloor 6 / 2 \rfloor = 3$

Toplam kayıt sayısı



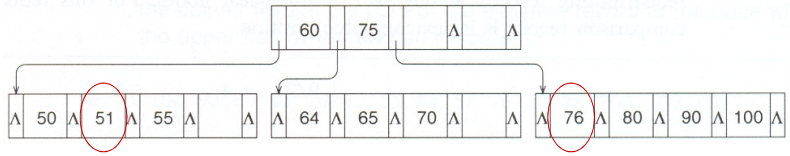
B-Tree and Derivatives

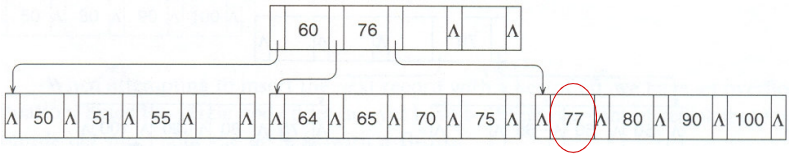
B#-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150





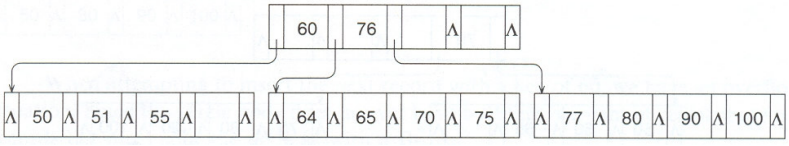
B-Tree and Derivatives

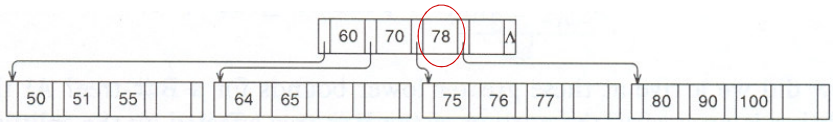
B#-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150





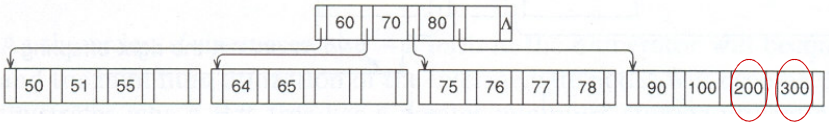
B-Tree and Derivatives

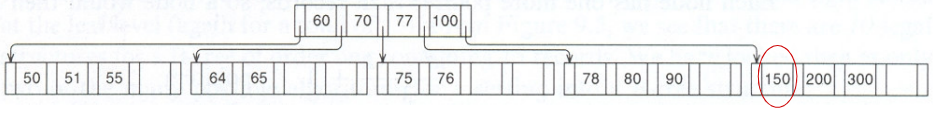
B#-Trees

- Örnek

d = 2 (capacity order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150





B-Tree and Derivatives

B#-Trees

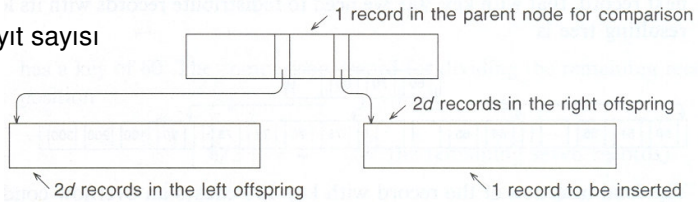
- Tanım

$$\frac{4d - 2}{3} \leq \text{keys} \leq 2d$$

$$\frac{4d + 1}{3} \leq \text{pointers} \leq 2d + 1$$

Toplam kayıt sayısı

$4d + 2$



Node'lara dağıtılacak en az kayıt sayısı

$$\left\lfloor \frac{4d}{3} \right\rfloor$$

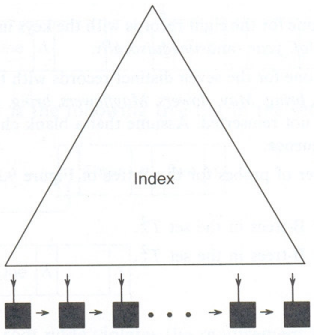
Minimum doluluk oranı

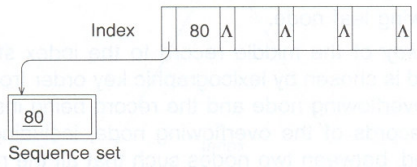
$$\frac{(4d - 2)/3}{2d} \rightarrow \frac{4d - 2}{6d} \rightarrow 2/3$$

B-Tree and Derivatives

B+ -Trees

- B+ -tree'lerde sıralı okuma için parent'a ulaşmaya gerek yoktur
- Bilgiler yapraklarda bulunur. Nonleaf node'lar sadece indeks için kullanılır
- Tüm yapraklar tek bağlı listeyle bağlanır
- B+ - tree'lerde indeks kısmı B-tree ile aynıdır. İndeks kısmındaki tüm kısıtlamalar ve işlemler B-tree ile aynıdır





B-Tree and Derivatives

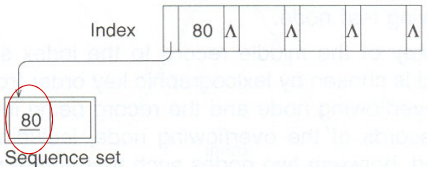
B+ -Trees

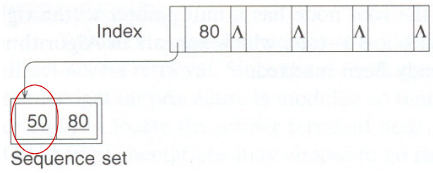
- Örnek

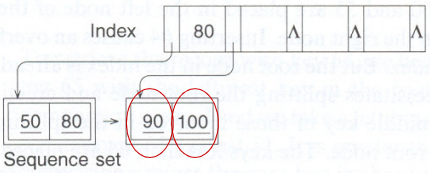
d = 2 (capacity order)

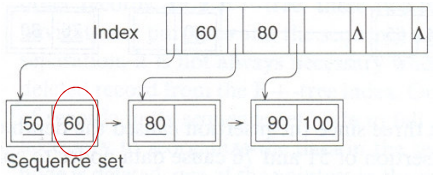
s = 1 (sequence set order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150









B-Tree and Derivatives

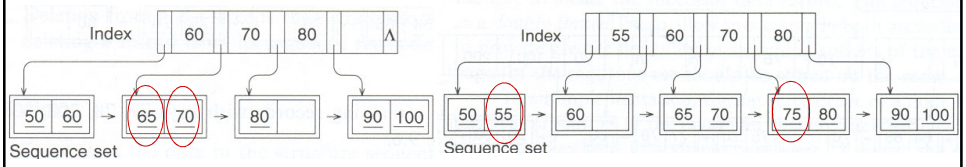
B+ -Trees

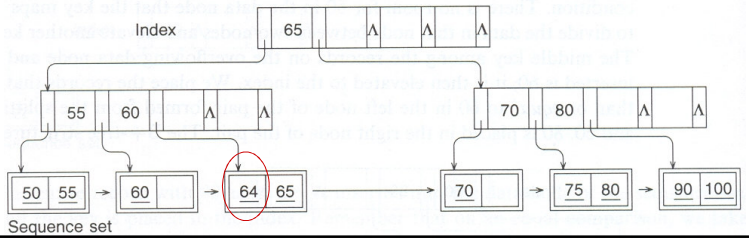
- Örnek

d = 2 (capacity order)

s = 1 (sequence set order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150





B-Tree and Derivatives

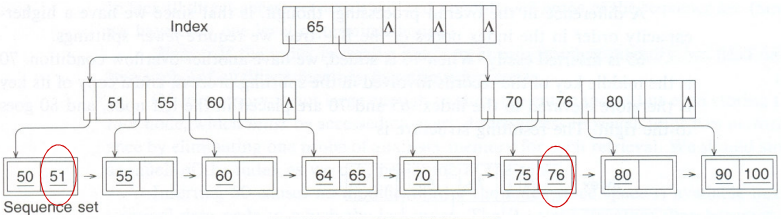
B+ -Trees

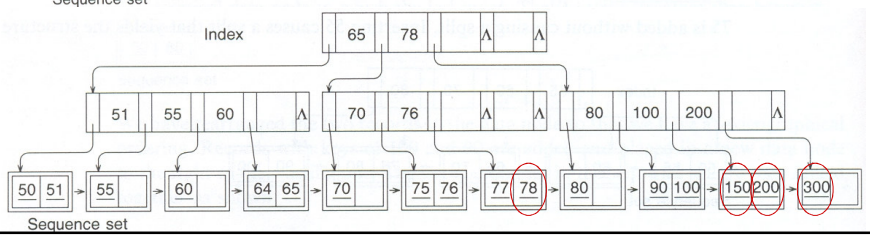
- Örnek

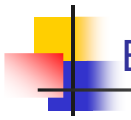
d = 2 (capacity order)

s = 1 (sequence set order)

80,50,100,90,60,65,70,75,55,64,51,76,77,78,200,300,150







Binary Tree Structures

Haftalık Ödev

B+ - tree'lerde ve B# - tree'lerde silme işleminin nasıl yapıldığını araştırınız ve bir rapor hazırlayınız .