Veritabanı Yönetim Sistemleri (335)

Dr. Öğr. Üyesi Ahmet Arif AYDIN

L19-

B+ Tree

GÜZ -2022

B+ Ağaç Yapısı

- Esnek ve dinamik bir yapısı bulunmaktadır.
- Her bir **düğüm** disk üzerinde bir **sayfadır**.
- İki adet düğüm çeşidi bulunmaktadır:
  - Yaprak düğümler (leaf nodes)
  - Yaprak olmayan düğümler (non-leaf nodes)

B+ Ağaç Yapısı: Yaprak Düğümler

```
struct YaprakNode {
    vector<Key> keys;
    vector<Value> values;
    PagePointer sonraki_sayfa;
}
```

B+ Ağaç Yapısı: Yaprak(leaf) Düğümler

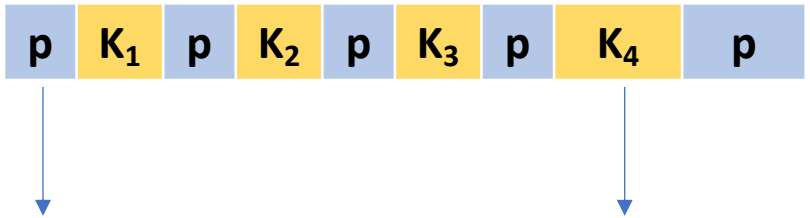
```
struct YaprakNode {
    vector<Key> keys;
    vector<Value> values;
    PagePointer sonraki_sayfa;
}
```

Verinin kaydedildiği sayfalar birbirine bağlı **linkedlist** ile tanımlanabilir.

Düğümlerde bulunan anahtarlar (key) ve yapraklarda (leaf) bulunan değerler sıralıdır.

B+ Ağaç Yapısı: Yaprak Olmayan(non-leaf) Düğümler

```
struct InteriorNode {
    vector<Key> keys;
     vector<PagePointer> pointers;
}
```



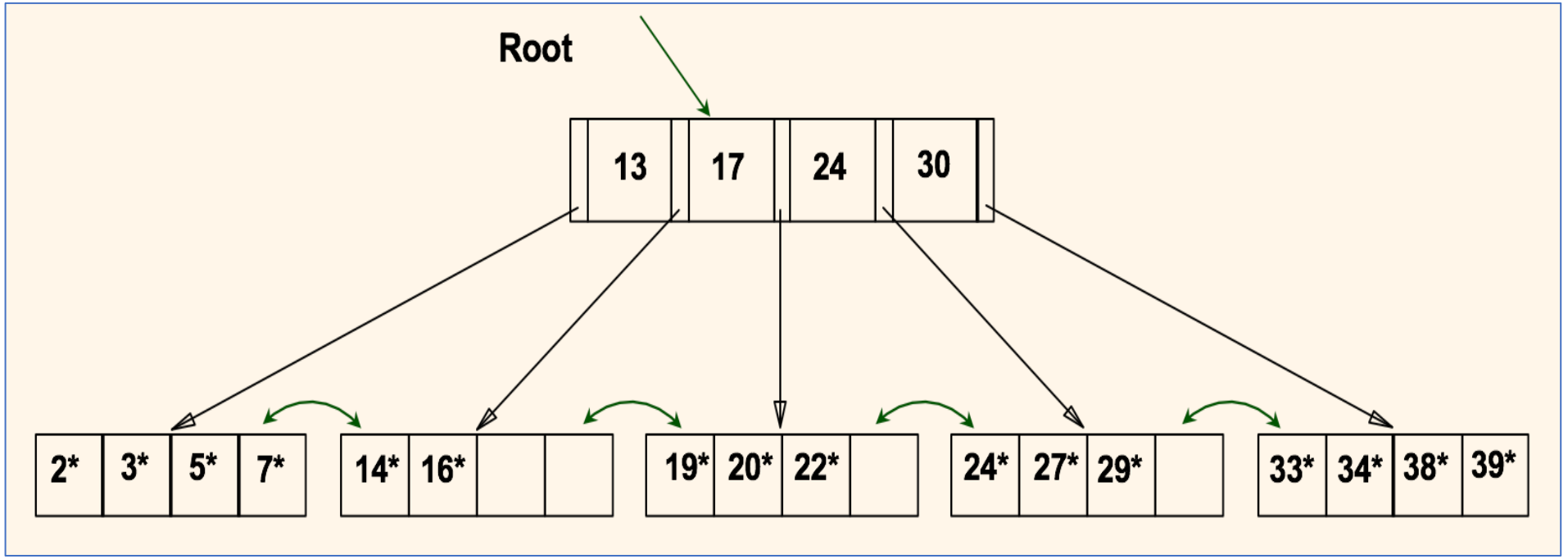
Pointer sayısı anahtar sayısından 1 fazla

page (sayfa) pointeri

Anahtar

B+ Ağaç Yapısı

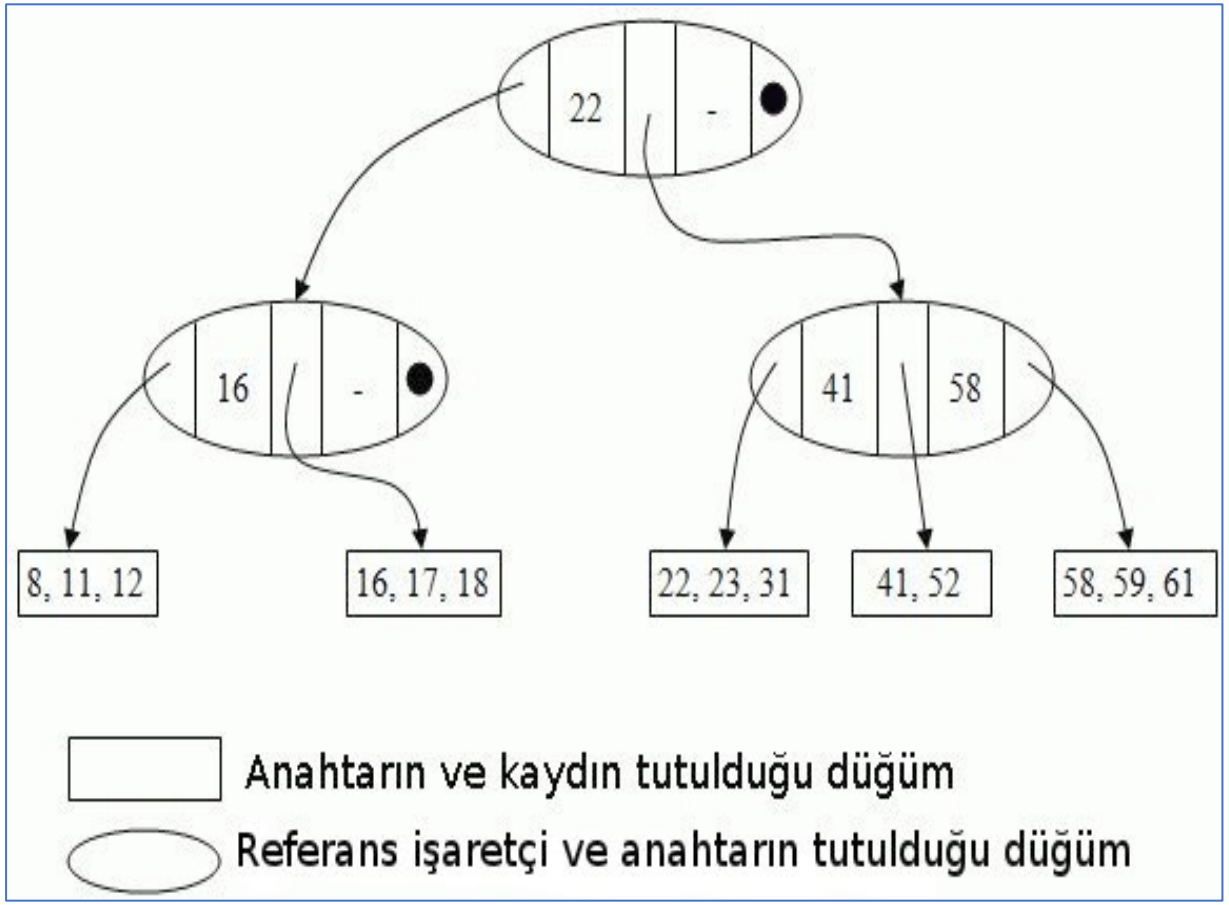
- B+ ağacının yapısında bulunan işaretçiler (pointer) yardımıyla aramalar yönlendirilir.
- Leaf nodes (yaprak düğümleri) kaydedilen verileri içermektedir.
- B+ ağacının yapısı eklenen veya silinen anahtara göre dinamik olarak değişir.



B+ Ağaç Yapısı

- B+ Tree dinamik olarak büyüyüp (grow) küçülür (shrink) ve *ağacın yüksekliği de dinamik olarak değişir.*

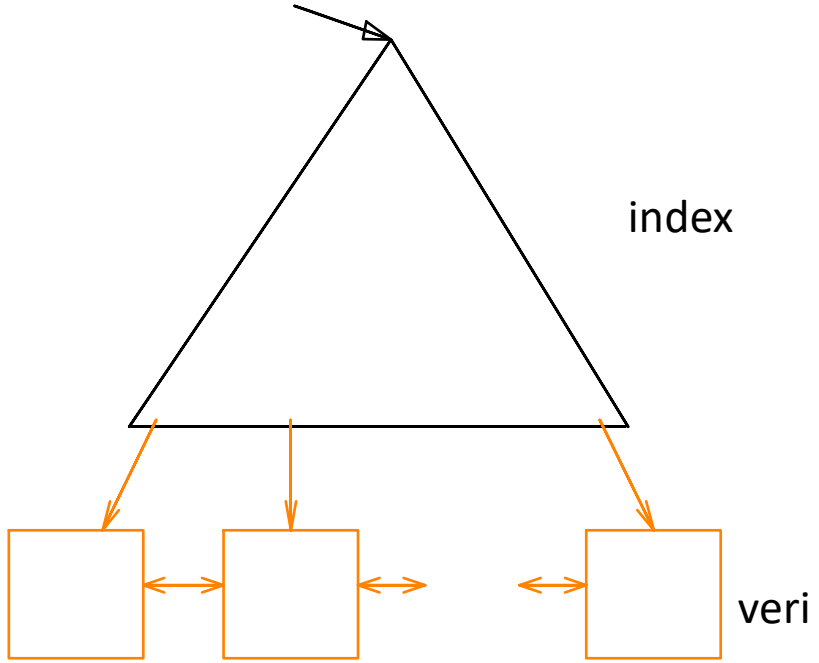
- Anahtarlar ve işaretçileri (pointer) *yaprak olmayan düğümlerde* (non leaf node) tutulur.



http://e-bergi.com/y/B+-Agaclari

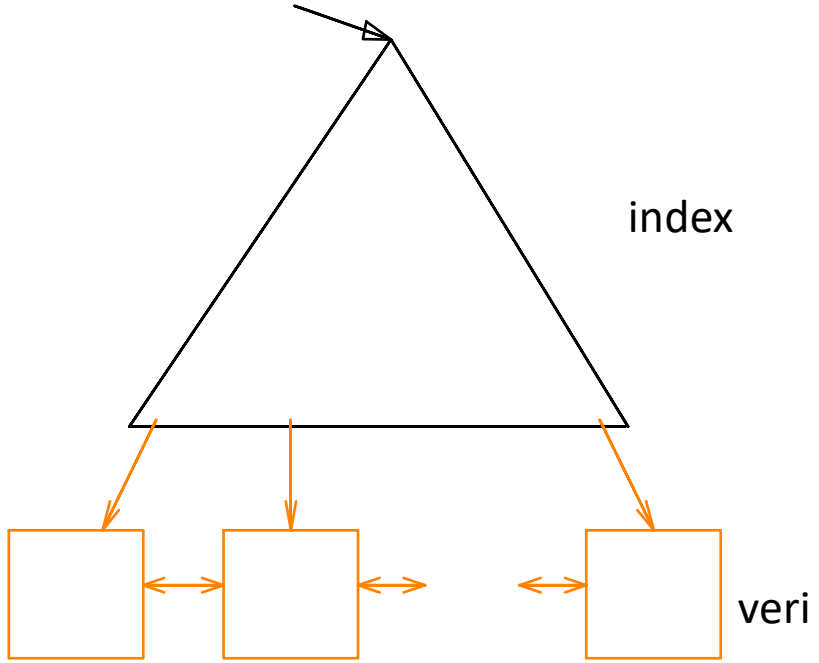
B+ Ağaç Yapısının Temel Karakteristikleri

- Bir tane root (ana düğüm) bulunmak zorundadır.
- Her bir düğüm için (root hariç) ortama %50 doluluk oranı istenmektedir
  - d <= düğümde bulunabilecek anahtar sayısı <= 2d
  - d parametresi (order of tree): bir düğümde bulunabilecek minimum kayıt sayısını belirler
  - d bir sayfaya kaydedilen kayıtların kapladığı alana göre belirlenir
- B+ ağacının yüksekliği: $h = \log_k N$
  - k= *düğümün kapasitesi*, N= *kayıt sayısı*



B+ Ağacı : Pratikdeki değerler

- Tipik order 100 ve doluluk oranı % 67
  - Düğümlerde bulunan ortalama alt dügüm sayısı (fanout) 133
- Kapasite
  - Yükseklik 4: 133^4 = 312,900,700 kayıt
  - Yükseklik 3: 133^3 = 2,352,637 kayıt
- Ara bellekte tutulan degerler :
  - Seviye 1 = 1 page = 8 Kbytes
  - Seviye 2 = 133 pages = 1 Mbyte
  - Seviye 3 = 17,689 pages = 133 MBytes



B+ Ağaç Yapısı: Arama (search) işlemi

Find record with search-key value *V*.

1. *C*=*root*
2. While C is not a leaf node {
   1. Let *i* be least value s.t. $V \leq K_i$.
   2. If no such exists, set *C* = *last non-null pointer in C*
   3. Else { if (V= K_i) Set C = P_{i+1} else set *C* = P_i}
   }
3. Let *i* be least value s.t. $K_i = V$
4. If there is such a value *i*, follow pointer P_i to the desired record.
5. Else no record with search-key value *k* exists.

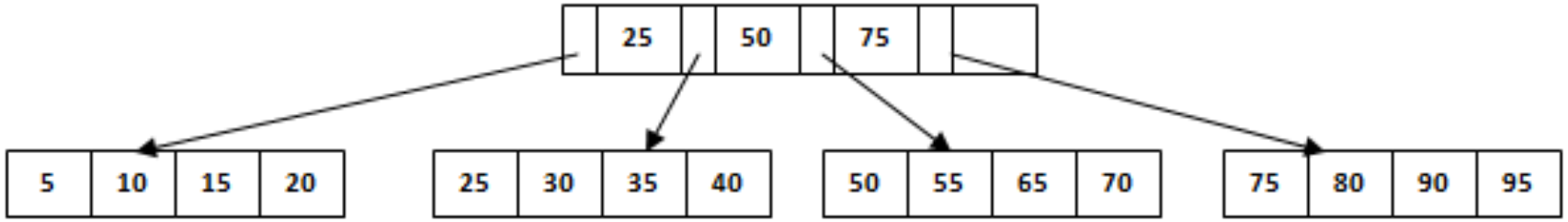
B+ Ağaç Yapısı: Arama (search) işlemi

1. Aramaya root (kök düğüm) ile başlanır
2. Arama yapılan düğüm (node) bir yaprak değilse;
   - Anahtar değerine göre düğümde arama yapılır
   - Aranılan **anahtar değerden küçük olan en yüksek değerli anahtardan sonraki** işaretleyicinin gösterdiği düğüme gidilir.
   - Arama yapılan düğüm güncellenir
   - 2.adıma geri dönülür.
3. Arama yapılan node (düğüm) bir yaprak ise **aranılan değer bu düğümdedir veya yoktur.**

B+ Ağaç Yapısı: Arama (search)

65 değerini arayalım

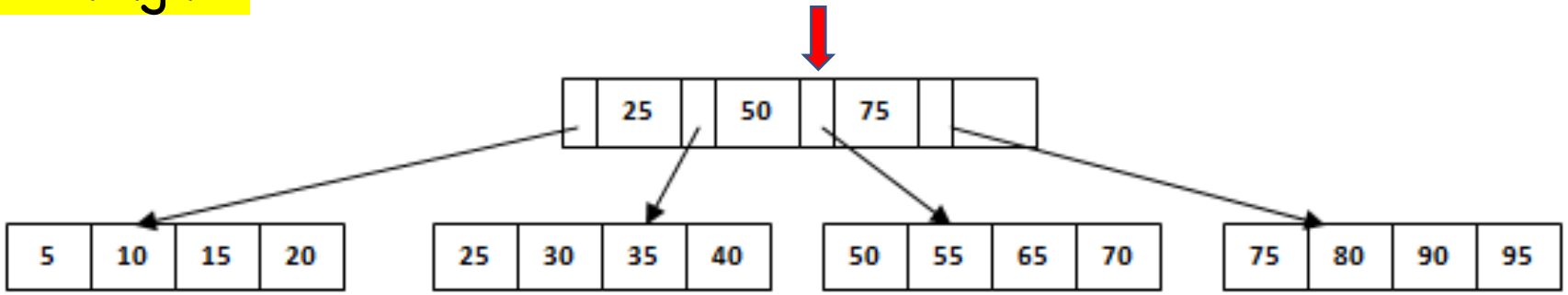
https://www.tutorialcup.com/dbms/b-tree.htm



B+ Ağaç Yapısı: Arama (search)

65 değerini arayalım

https://www.tutorialcup.com/dbms/b-tree.htm

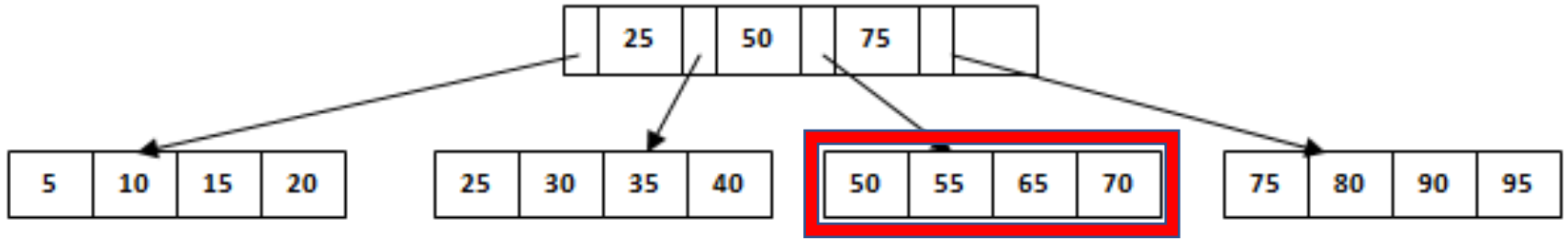


1. 65 değerinden küçük olan en büyük anahtardan sonraki pointera gidilir

B+ Ağaç Yapısı: Arama (search)

65 değerini arayalım

https://www.tutorialcup.com/dbms/b-tree.htm

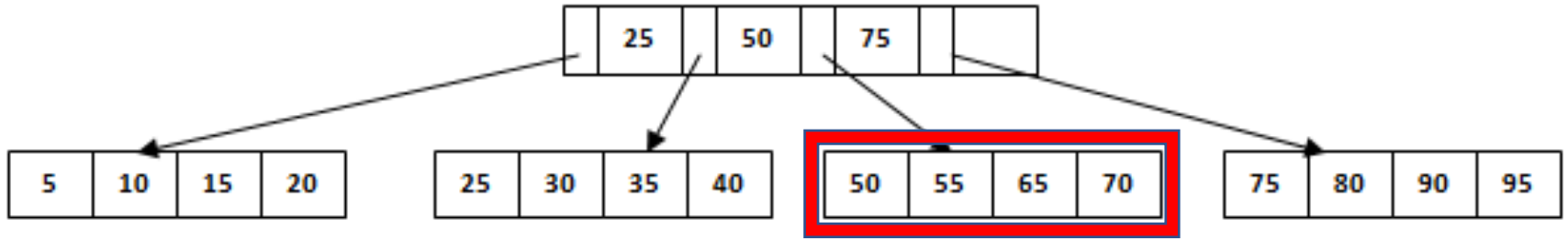


1. 65 değerinden küçük olan en büyük anahtardan sonraki pointera gidilir
2. Bulunan düğüm yaprak (leaf) oldugu için bu node içerisinde bulunan kayıtlar seri olarak (sequential) olarak taranır.

B+ Ağaç Yapısı: Arama (search)

65 değerini arayalım

https://www.tutorialcup.com/dbms/b-tree.htm



1. 65 değerinden küçük olan en büyük anahtardan sonraki pointera gidilir
2. Bulunan düğüm yaprak (leaf) oldugu için bu node içerisinde bulunan kayıtlar seri olarak (sequential) olarak taranır.

Arama (search) işlemi gerçekleştirilirken
insert, delete ve update
İşlemlerine izin verilmemektedir

B+ Ağaç Yapısı: Ekleme (insert) işlemi

1- Eklenmesi gereken kayıt için (düğüm) uygun düğüm aranır

2-Eğer düğüm maximum kaydedebileceği değerden az kayıt bulunduruyorsa kayıt sırası dikkate alınarak ekleme yapılır (işlem tamamlanır)

3- Eklenecek değer için düğümde yer yoksa

- Düğüm ikiye ayrılır, yeni bir yaprak (leaf) oluşturulur
- Elemanların yarısı sırası degişmeden oluşturulan yeni yaprağa eklenir.
- *Yeni oluşturulan düğümün anahtarı bir üst seviyedeki düğüme eklenir*
- *Üst seviyedeki düğüm dolu ise ikiye ayrılır ve ortadaki anahtar ust seviyeye alınır.*
- Bölünme gerektirmeyen düğüm oluşuncaya kadar devam eder.

4- Eğer root (kök) düğümünün bölünmesi gerekiyorsa tek bir degeri ve iki pointeri bulunan yeni bir root oluşturulur.

B+ Ağaç Yapısı: Ekleme (insert) işlemi

Root (kök) düğüm istisnadır.
Kökte olabilecek anahtar sayısı

1<= anahtar sayısı <= 2d

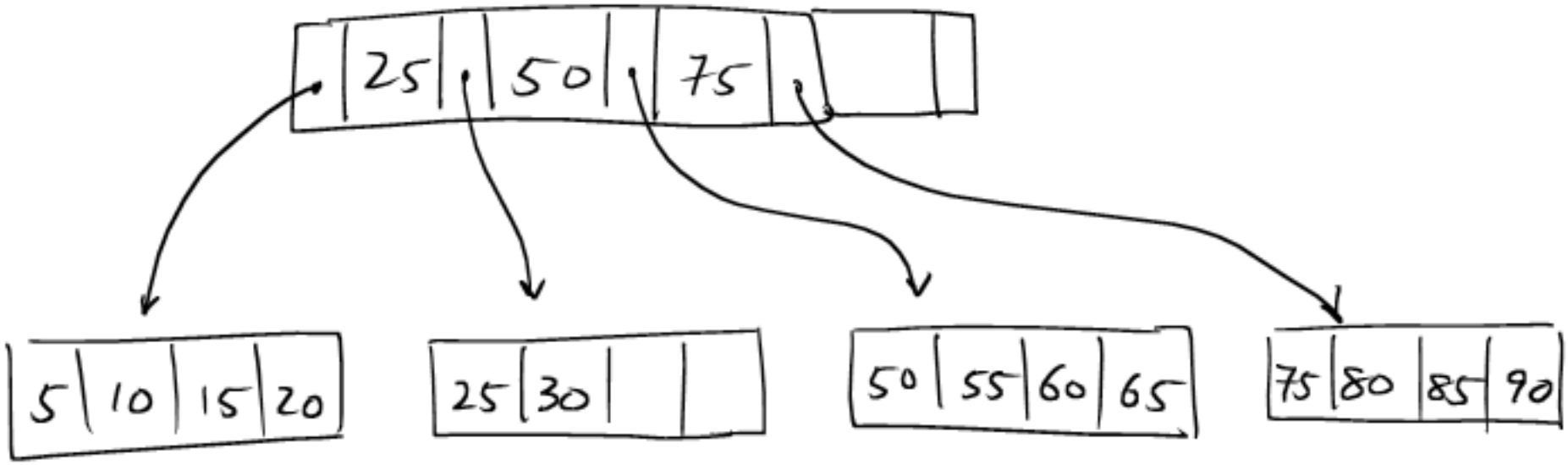
B+ ağaçının kök olmayan düğümlerin de bulunan anahtar sayısı

d<= anahtar sayısı <= 2d

B+ Ağaç Yapısı: Ekleme (insert) işlemi

28 değerini ekleyelim

Ekleyeceğiniz değerden küçük olan en büyük değerden sonraki pointeri bul

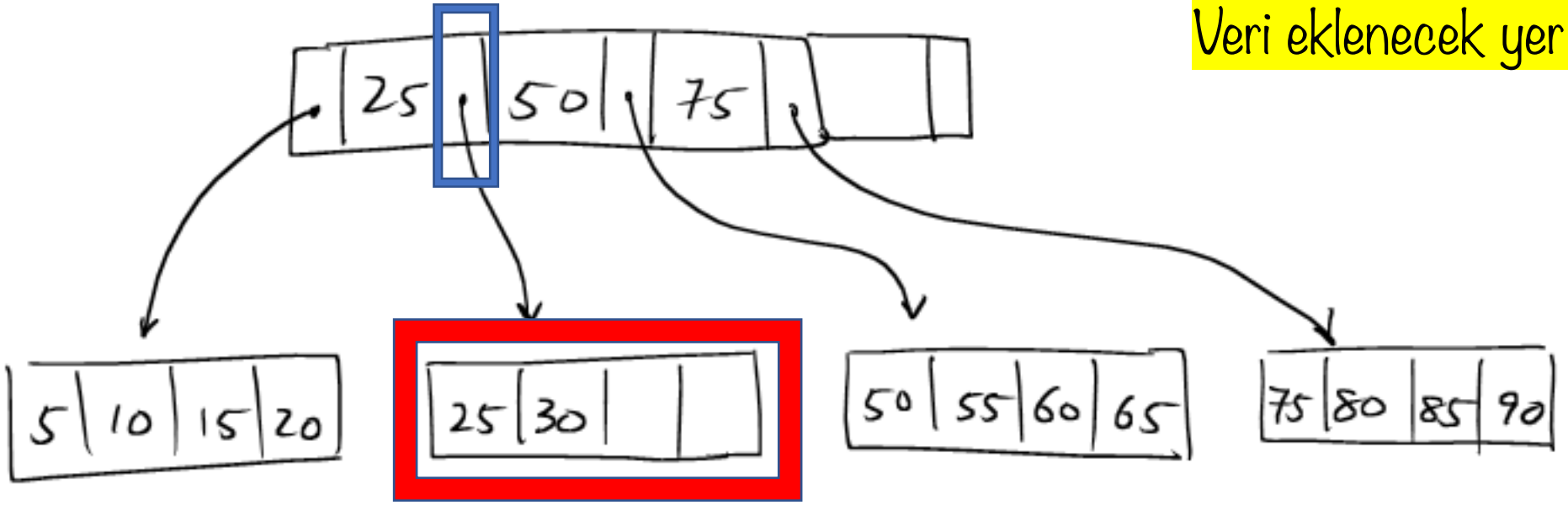


http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

28 değerini ekleyelim

Search (root, 28)



Pointer'in gösterdiği yaprak dügüm mü?

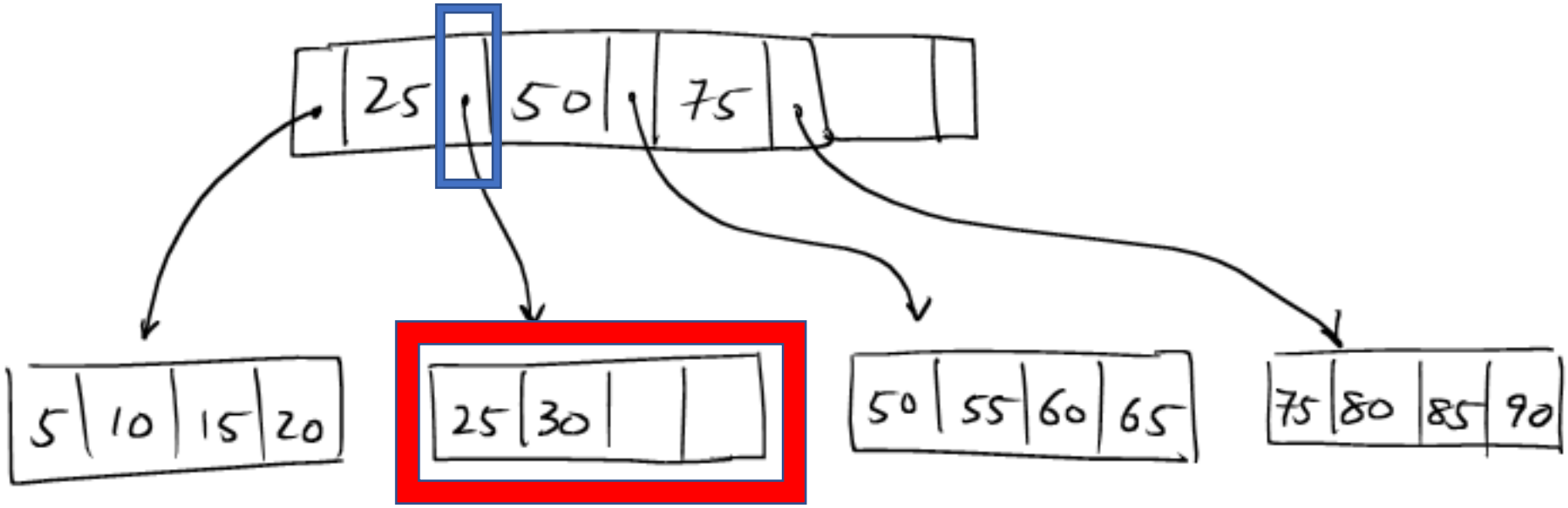
Veri eklenecek yer var mı?

http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

28 değerini ekleyelim

```
Search (root, 28)
```



Yaprak düğüm ve veri eklenecek alan var !

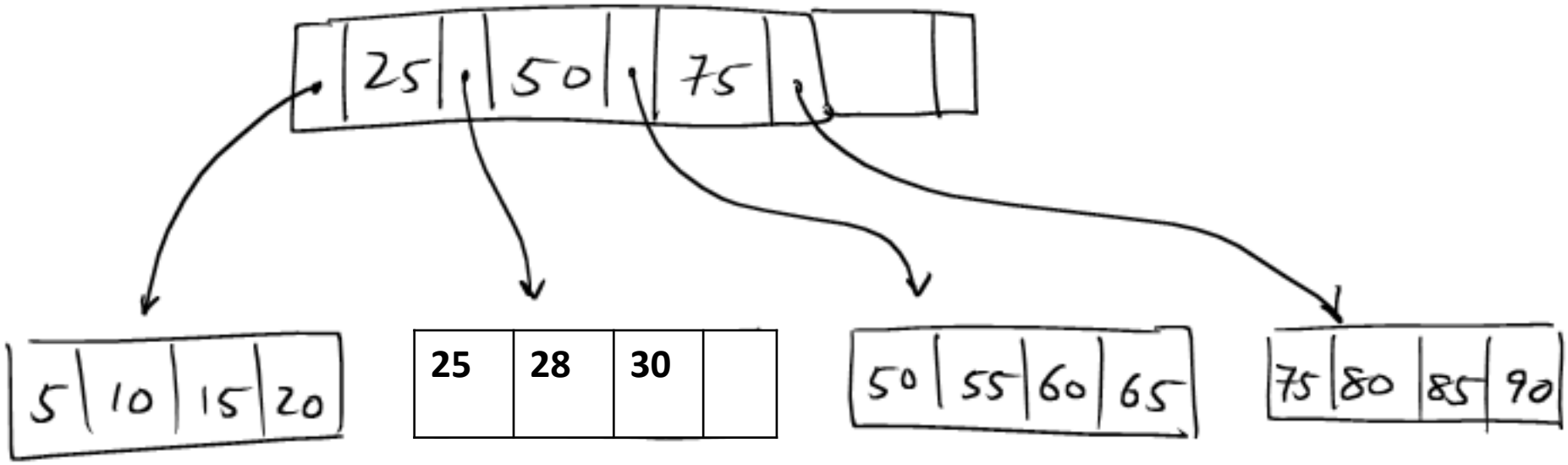
http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

28 değerini ekleyelim

Ekleyeceğiniz değerden küçük olan en büyük değerden sonraki pointeri bul

```
Search (root, 28)
```

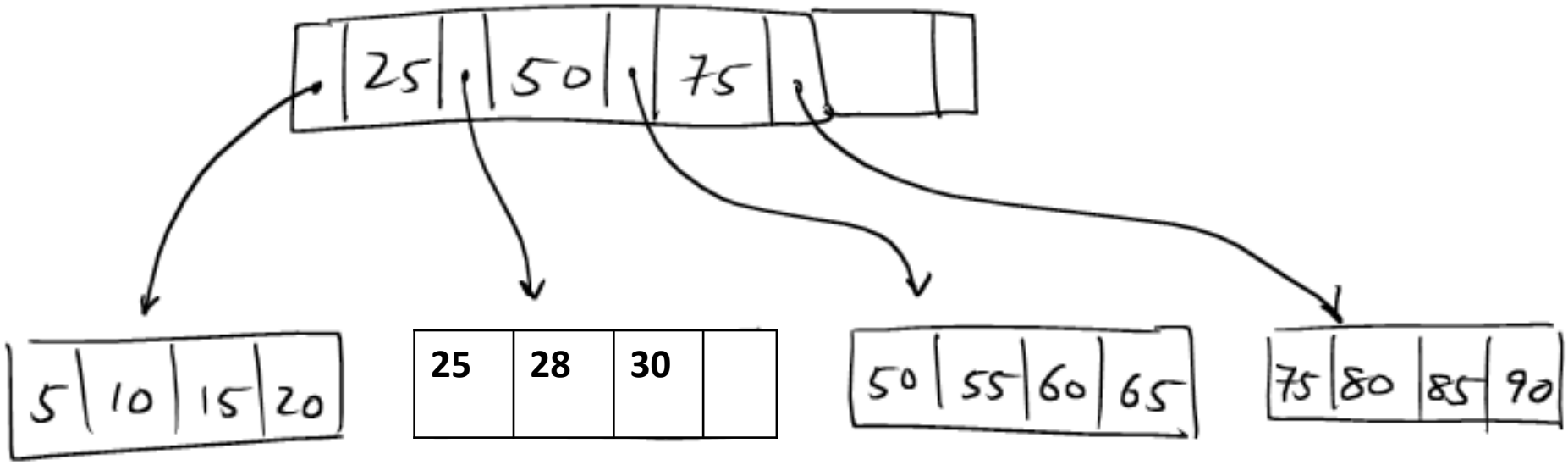


Yaprak düğümde yer olduğundan sayı eklendi ve yaprakda bulunan değerler kendi içinde sıralandı

http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

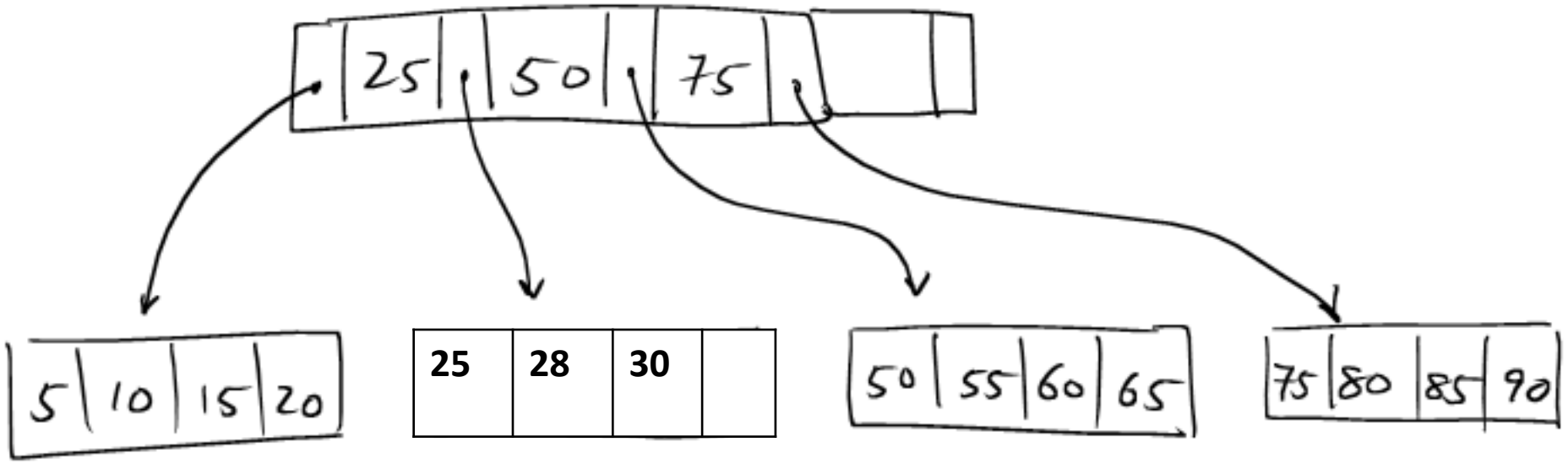


http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

Ekleyeceğiniz değerden küçük olan en büyük değerden sonraki pointeri bul

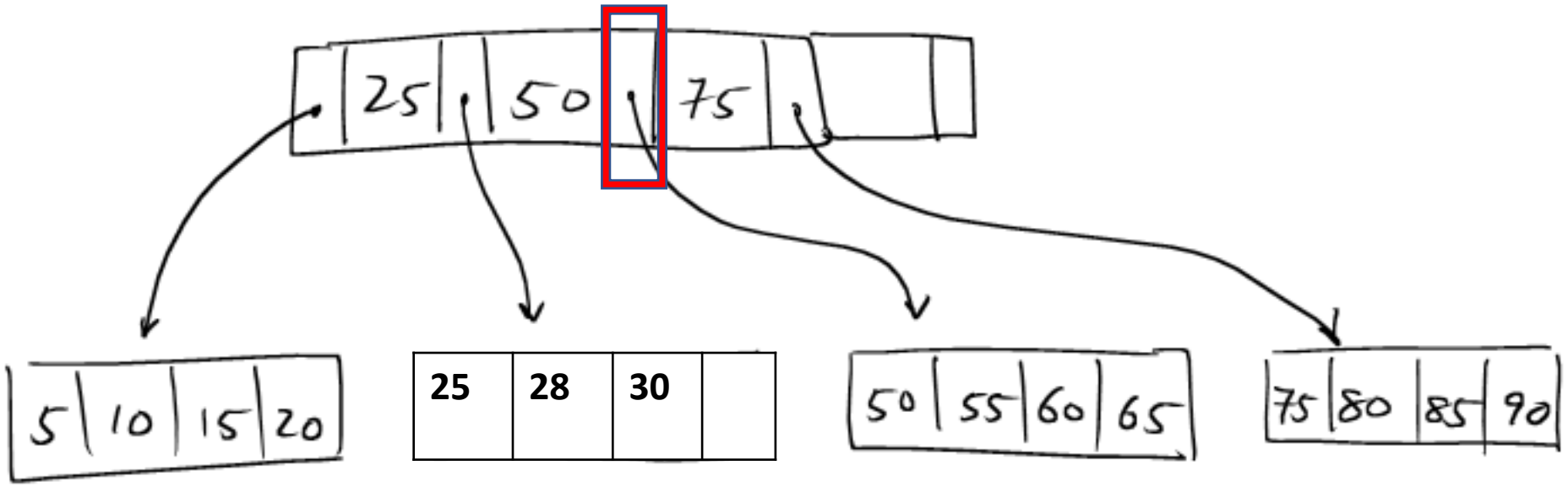


http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

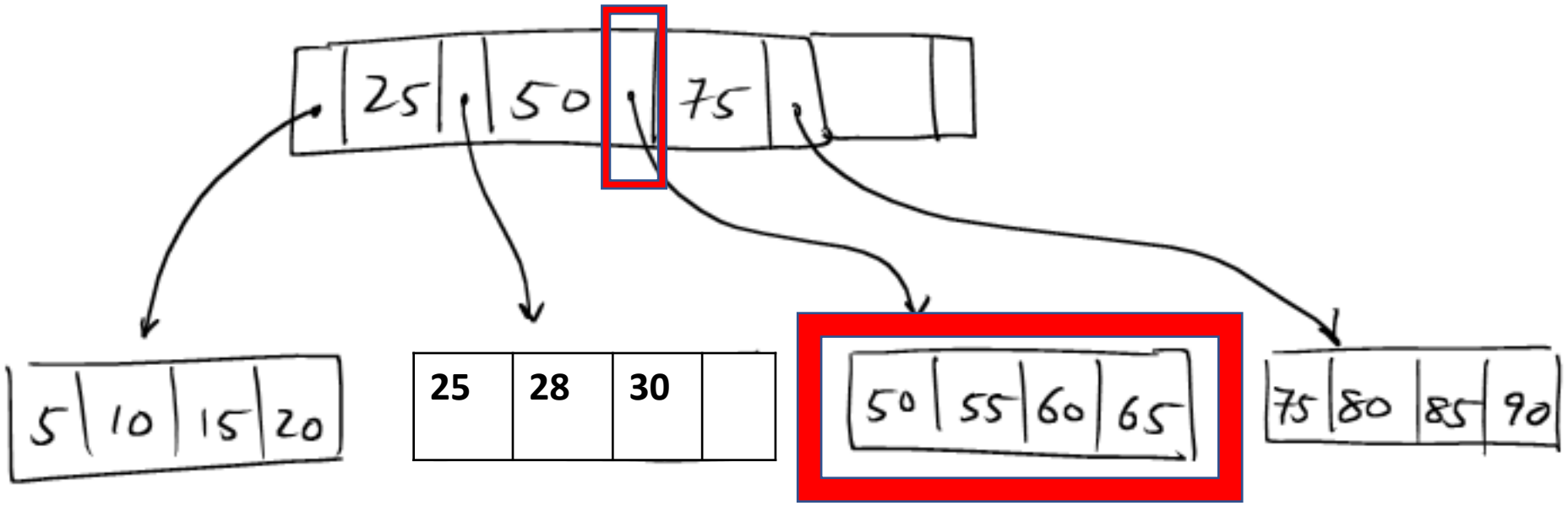
Ekleyeceğiniz değerden küçük olan **en büyük değerden** sonraki pointeri bul



B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

Ekleyeceğiniz değerden küçük olan **en büyük değerden** sonraki pointeri bul



Değerin eklenmesi gereken düğüm bulunur.

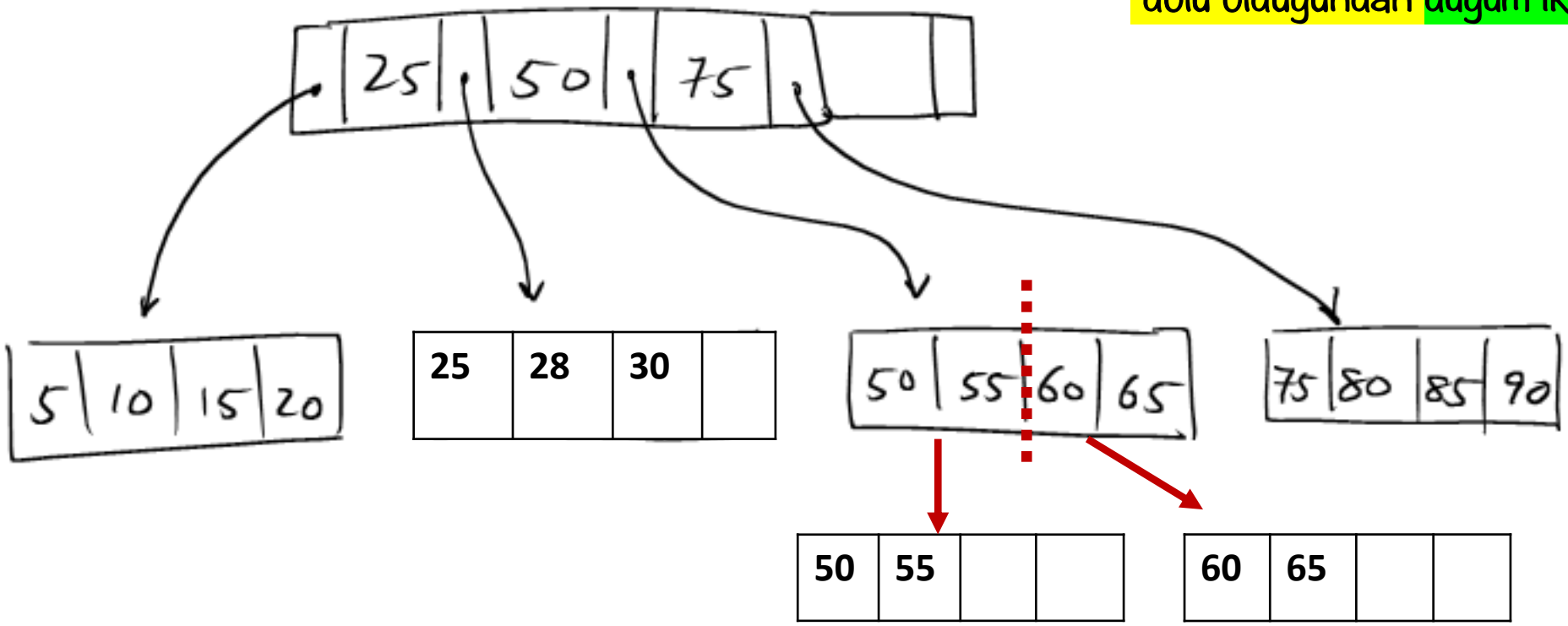
Yaprak düğümde değerin eklenmesi için eklenmesi gereken alan yok !!!!

http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

Değerin eklenmesi gereken düğüm dolu olduğundan düğüm ikiye bölünür

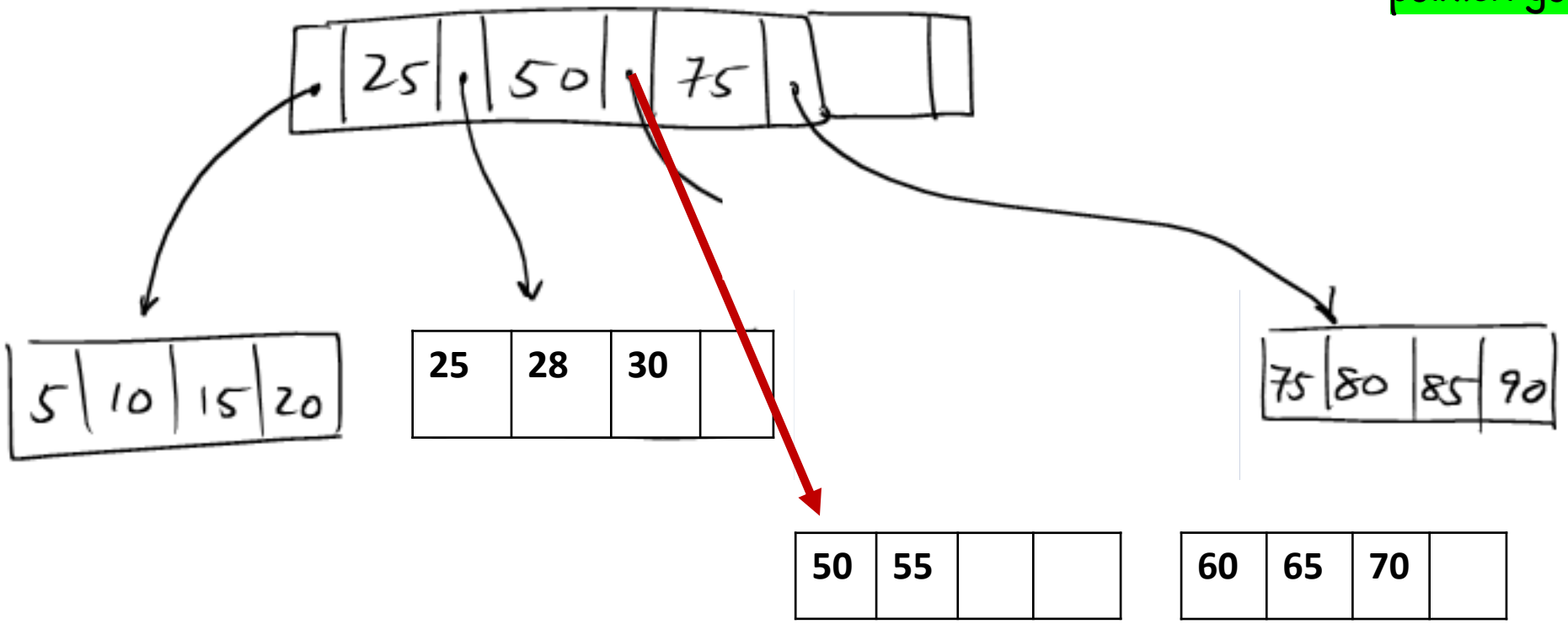


http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

Oluşturulan yeni düğüme yeni değer eklenir.
Fakat oluşturulan yeni düğümün pointeri yok !

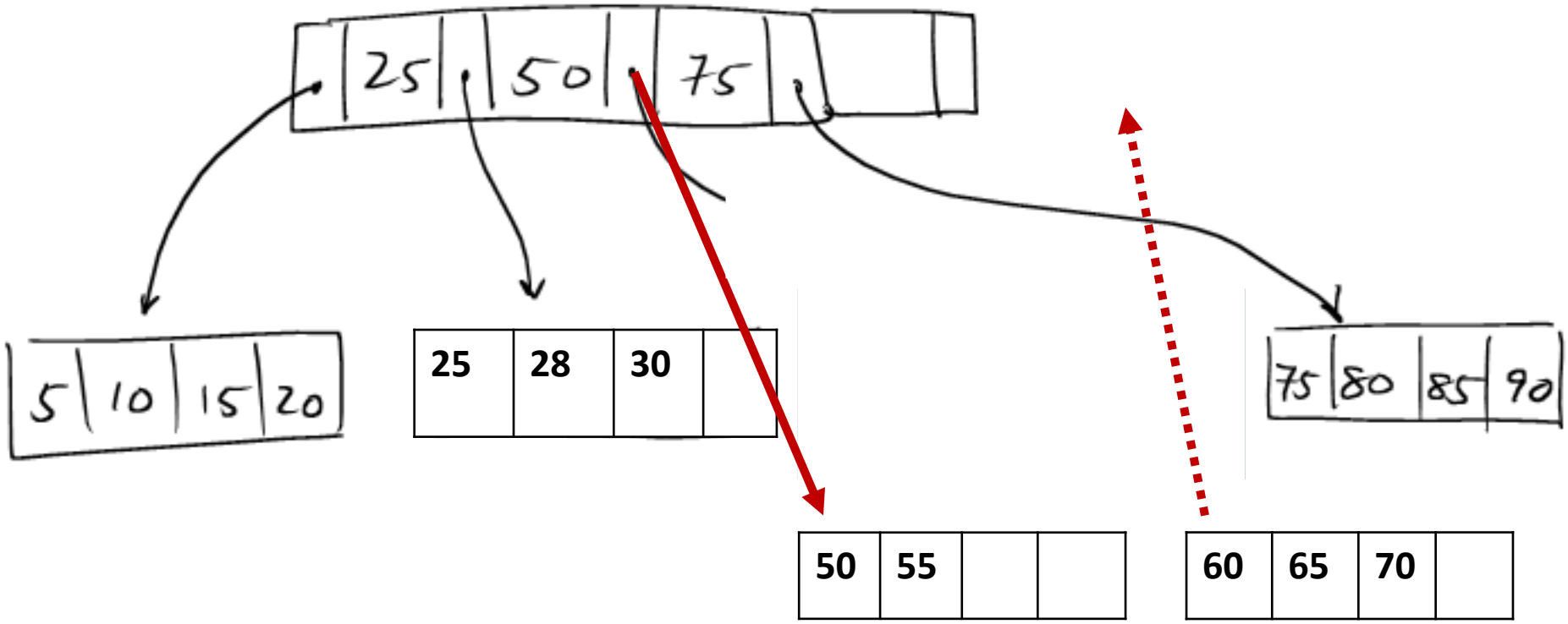


http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

Oluşturulan yeni düğüm için
bir üst seviyede
anahtar ve pointer gerekecek

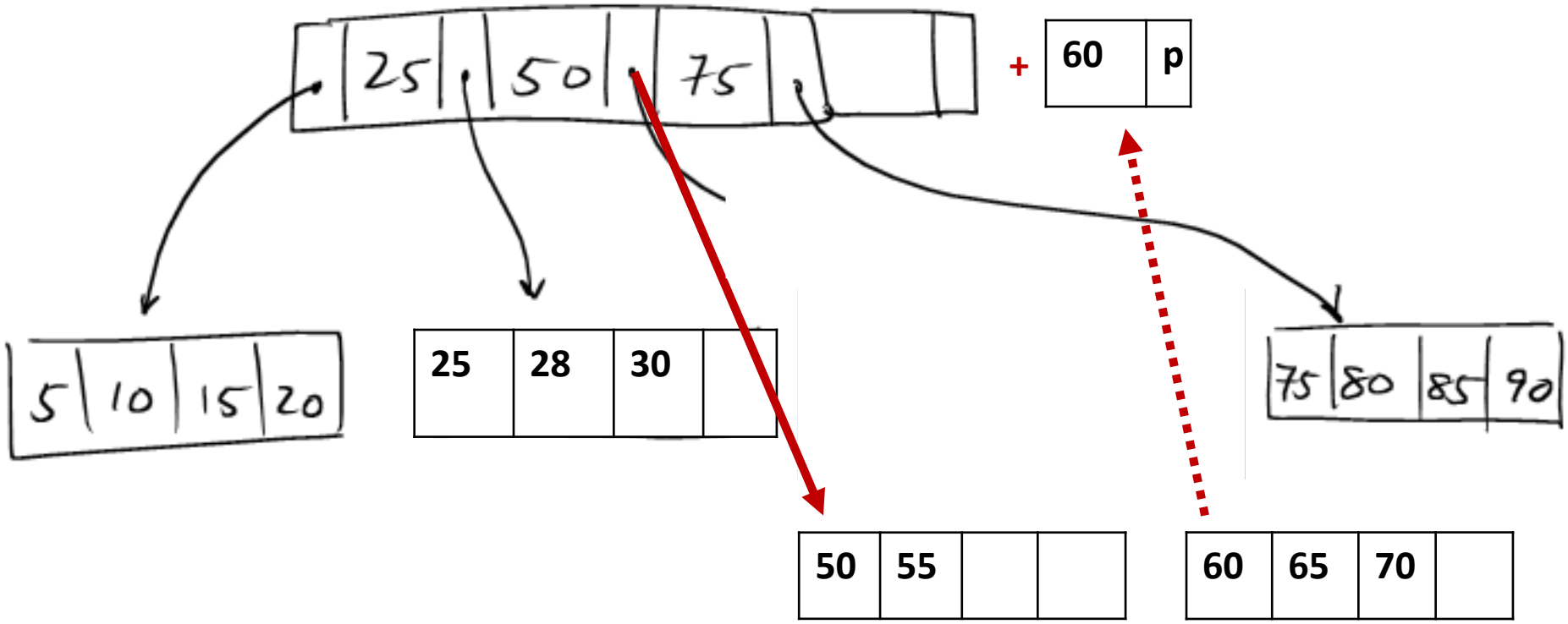


http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekleyelim

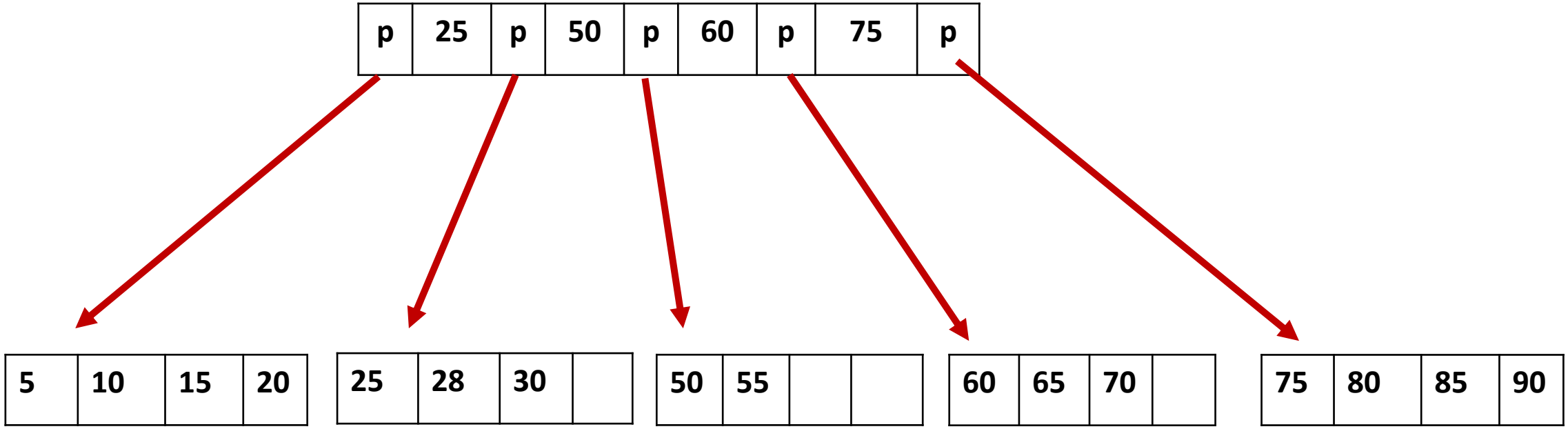
yeni düğümün anahtar ve pointeri bir üst seviyeye eklenecek



http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

70 değerini ekledikten sonra elde edilen yeni *ağaç*

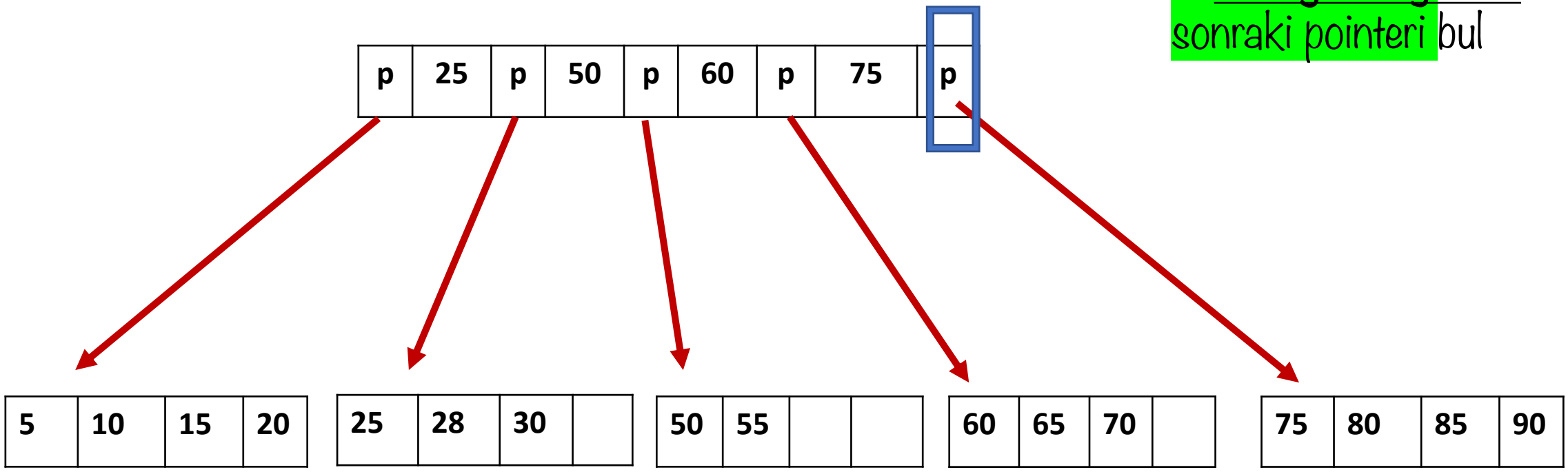


http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

Yeni ağaca 95 değerini ekleyelim

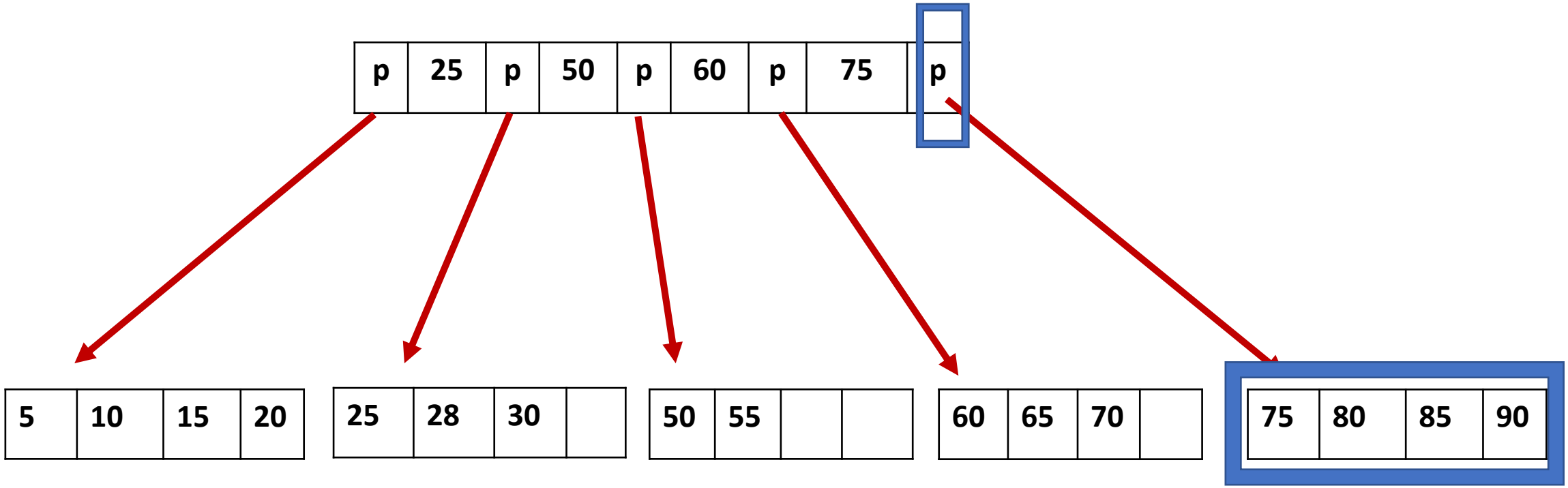
Ekleyeceğiniz değerden küçük olan en büyük değerden sonraki pointeri bul



http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

Yeni ağaca 95 değerini ekleyelim



http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

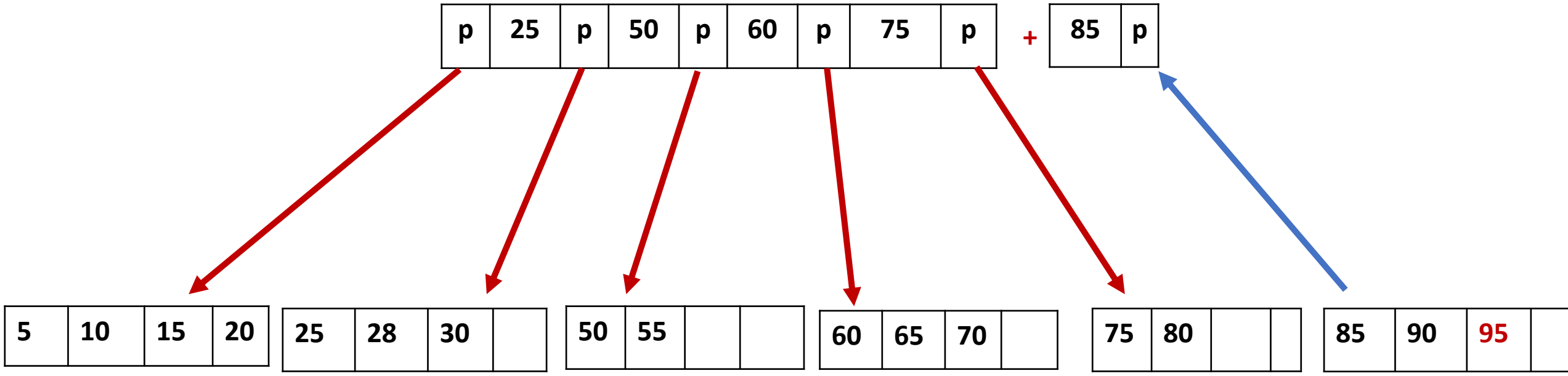
Yeni ağaca 95 değerini ekleyelim

yeni düğümün anahtar ve pointeri bir üst seviyeye eklenecek

http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

Yeni ağaca 95 değerini ekleyelim

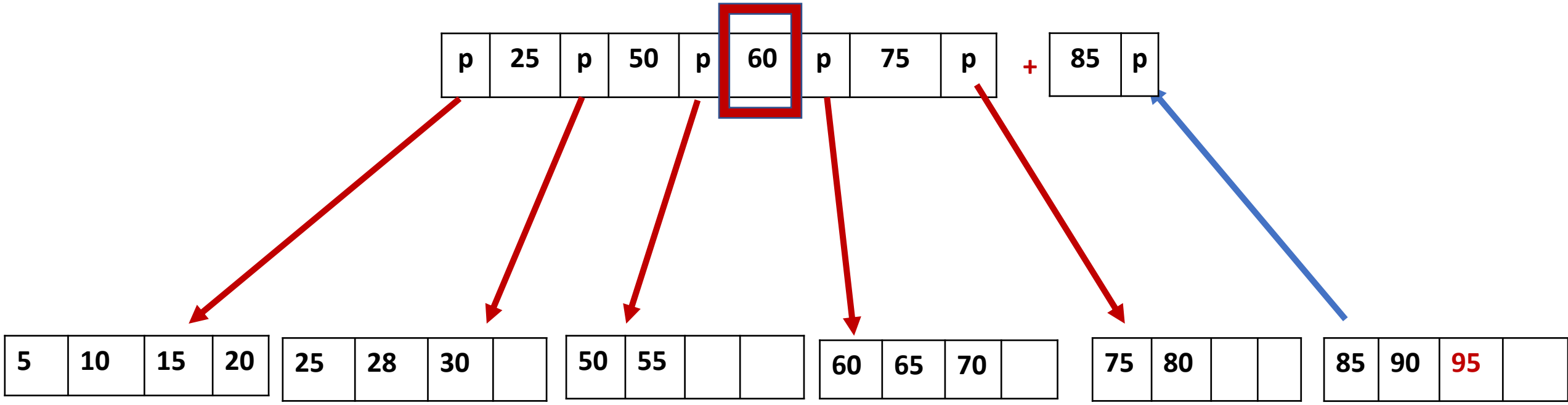


85 değerinin ve işaretçisinin eklenmesi için yeterli alan olmadığından kök düğüm bölünecek

http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi

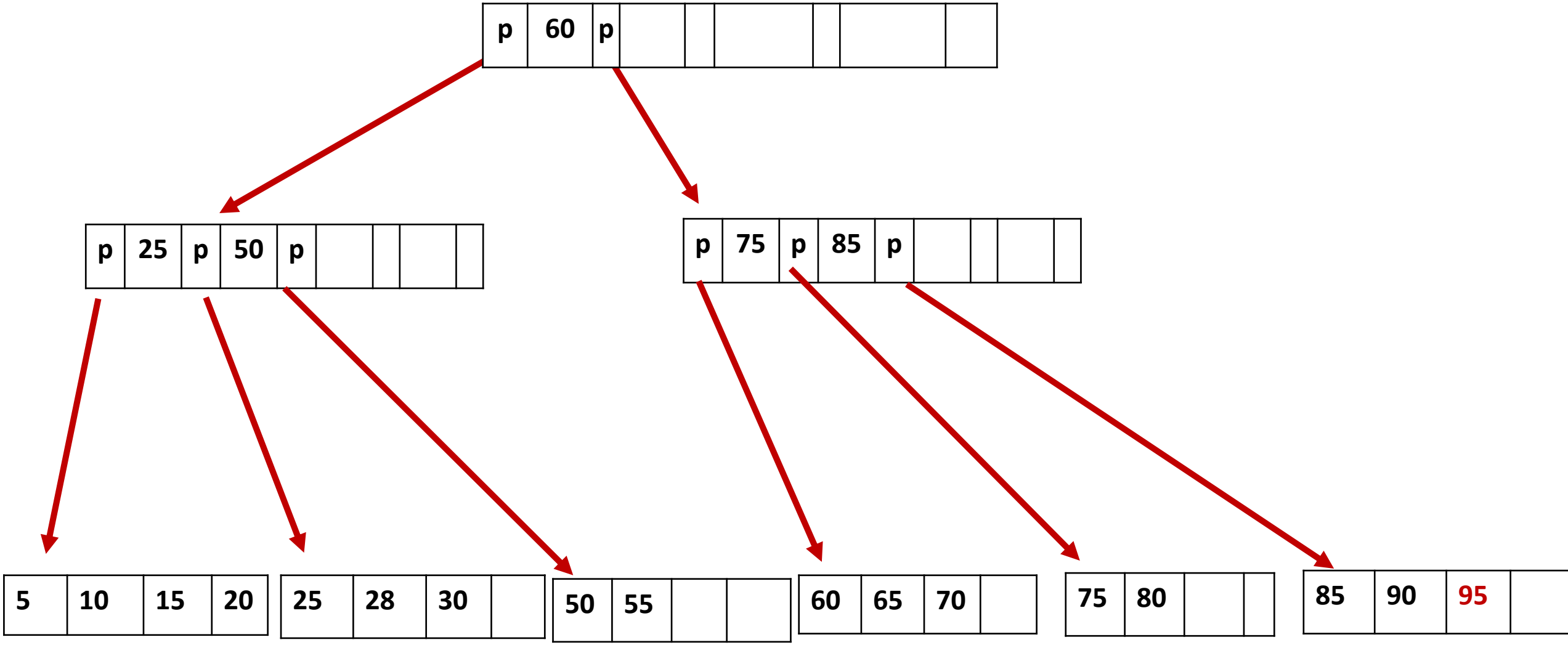
Yeni ağaca 95 değerini ekleyelim



ortadaki değer
İle yeni root (kök) dizin
oluşturulacak

http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Ekleme (insert) işlemi



http://www.teach.cs.toronto.edu/~csc443h/fall/posted_practice/btree-index.html

B+ Ağaç Yapısı: Silme (delete) işlemi

1. Silinecek kayıtın yaprak düğümü anahtar değeri yardımıyla bulunur.
2. Silme işleminin gerşekleştirileceği düğümde ağacın derecesine eşit veya büyük sayıda kayıt varsa silme işlemi gerçekleştirilir.
3. **d<= düğümdeki kayit sayisi <=2d** şartını saglamıyorsa düğümler aynı parent(ebeveyn) a sahıp düğümler birleştirilir (tekrar dagıtma işlemi)
4. Birleştirme işlemi gerşekleştirilirken bir ust seviyede (ebeveyn) sağdaki düğümü gösteren pointer ve anahtarı silinir
5. Bu işlem root (kök) a kadar devam eder.

Dinlediğiniz için

Teşekkürler...

İyi çalışmalar...