

Veritabanı Yönetim Sistemleri

Dr. Öğr. Üyesi Ahmet Arif AYDIN

B+ Tree

Dosya Yapıları

- ☐ Page (sayfa)
- ☐ Files layer (dosya katmanı)
- ☐ Disk space manager (disk alan yöneticisi)
- ☐ Buffer manager (ara bellek yöneticisi)

Index yapıları

- ☐ Hash based indexing
- ☐ Tree based indexing

- ❑ Esnek ve dinamik bir yapısı bulunmaktadır.
- ❑ Her bir **düğüm** disk üzerinde bir **sayfa**dır.
- ❑ İki adet düğüm çeşidi bulunmaktadır:
 1. Yaprak Düğümler (leaf nodes)
 2. Yaprak olamayan düğümler (non-leaf nodes)

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



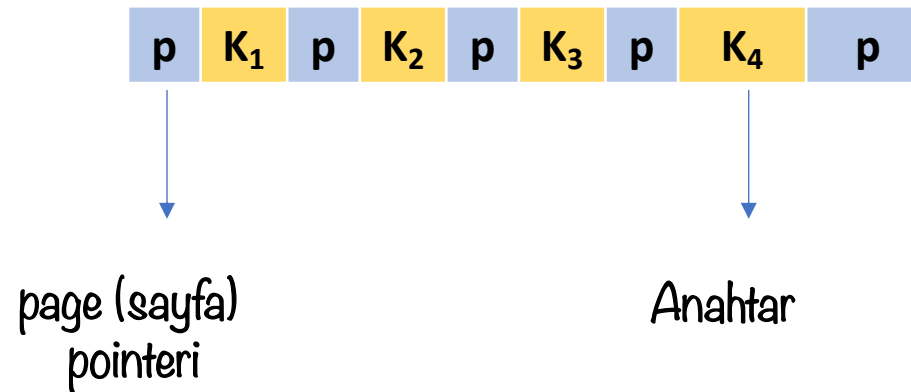
```
struct YaprakNode {  
    vector<Key> keys;  
    vector<Value> values;  
    PagePointer sonraki_sayfa;  
}
```



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

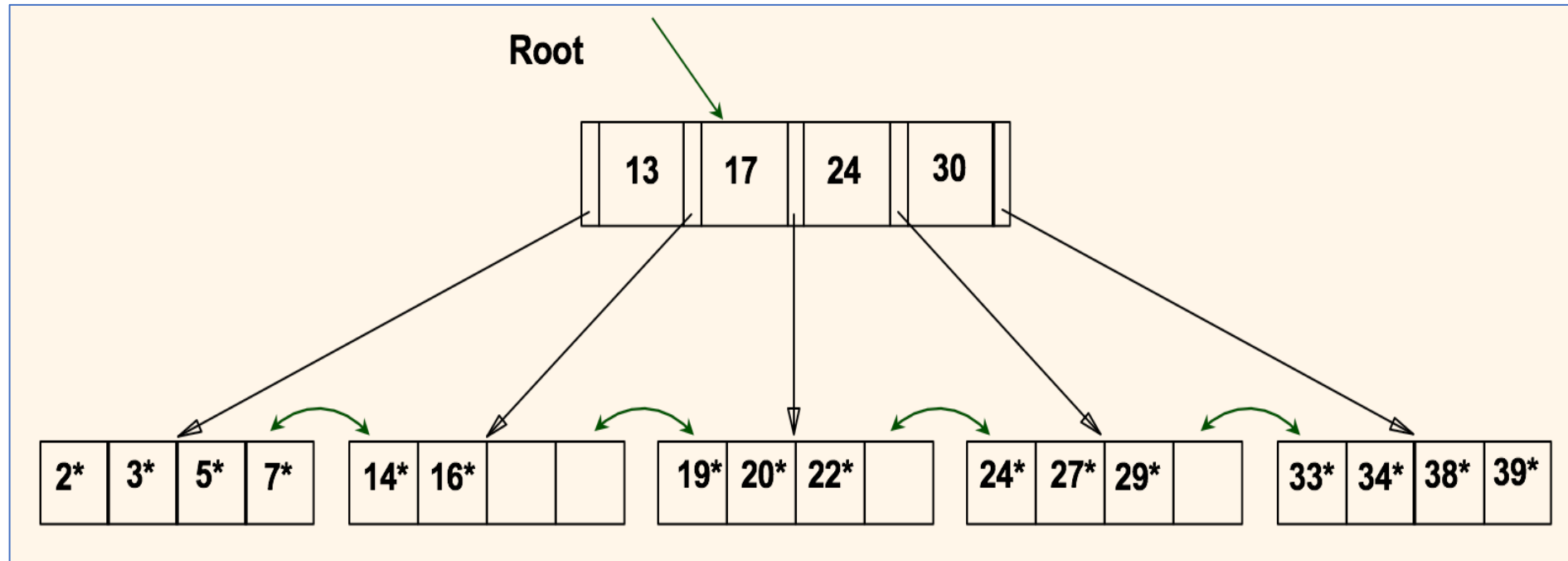
Düğümelerde bulunan anahtarlar (key) sıralıdır.
Yapraklarda (leaf) bulunan değerlerde sıralıdır.

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

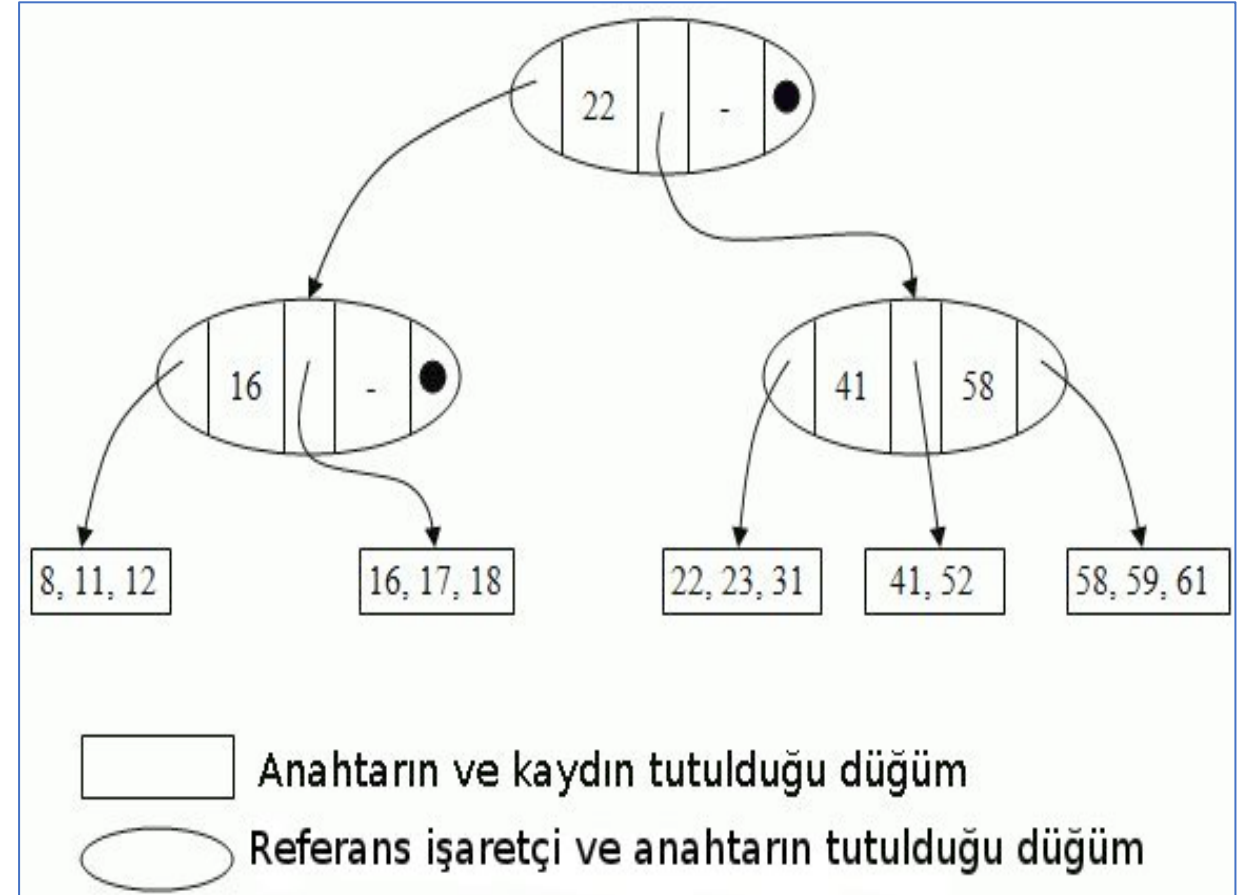


Pointer sayısı anahtar
sayısından 1 fazla

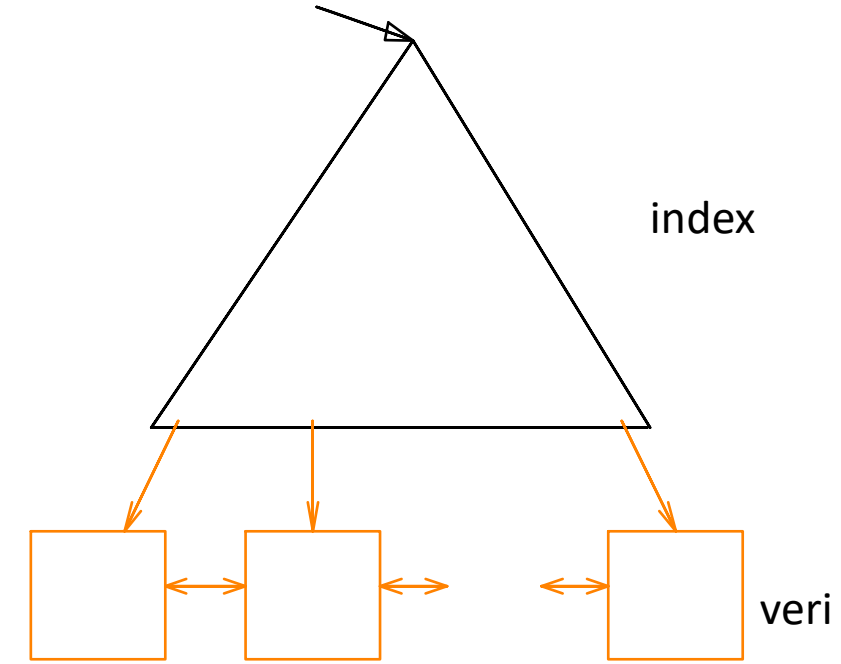
- ❑ B+ ağacının yapısının ara düğümleri (node) aramayı yönlendirmektedir.
- ❑ Leaf nodes (yaprak düğümleri) kaydedilen verileri içermektedir.
- ❑ B+ ağacının yapısı eklenen veya silinen anahtara göre değişebilir.



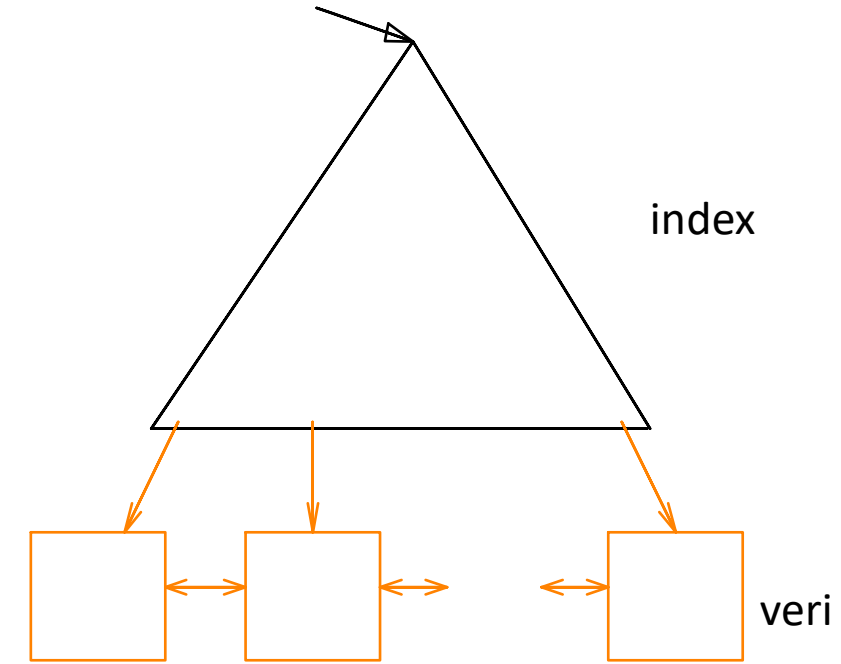
- ❑ B+ Tree dinamik olarak büyüyüp (grow) küçülmektedir (shrink) ve ağacın yüksekliği de dinamik olarak değişmektedir.
- ❑ Anahtarlar ve işaretçileri (pointer) yaprak olmayan (non leaf) düğümlerde tutulur.



- ❑ Bir tane root (ana düğüm) bulunmak zorundadır.
- ❑ Her bir düğüm için ortama %50 doluluk oranı (root hariç) istenmektedir
- ❑ $d \leq$ düğümde bulunabilecek anahtar sayısı $\leq 2d$
 - ❑ d parametresi (order of tree) bir düğümde bulunabilecek minimum kayıt sayısı.
 - ❑ d değeri sayfada tutulacak olan kayıtların boyutuna göre belirlenir
- ❑ B+ ağacının yüksekliği: $h = \log_k N$
 - ❑ k = düğümün kapasitesi , N = kayıt sayısı



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



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 deger bu düğümdedir veya yoktur.

Find record with search-key value V .

1. $C = \text{root}$

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 $C = \text{last non-null pointer in } C$

3. Else { if ($V = K_i$) Set $C = P_{i+1}$ else set $C = P_i$ }

}

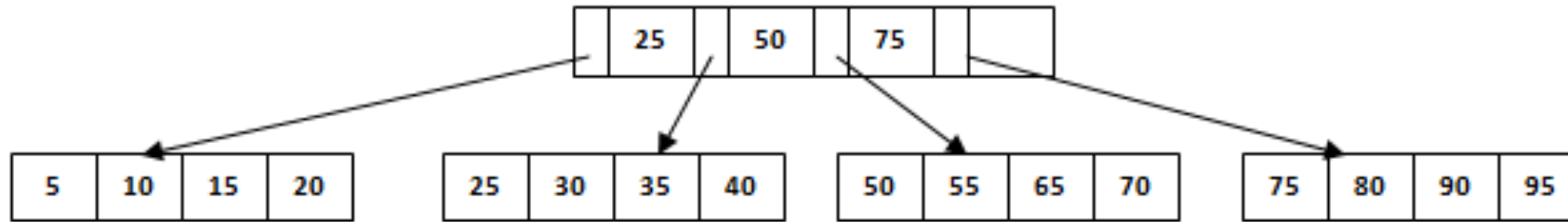
3. Let i be least value s.t. $K_i = V$

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.

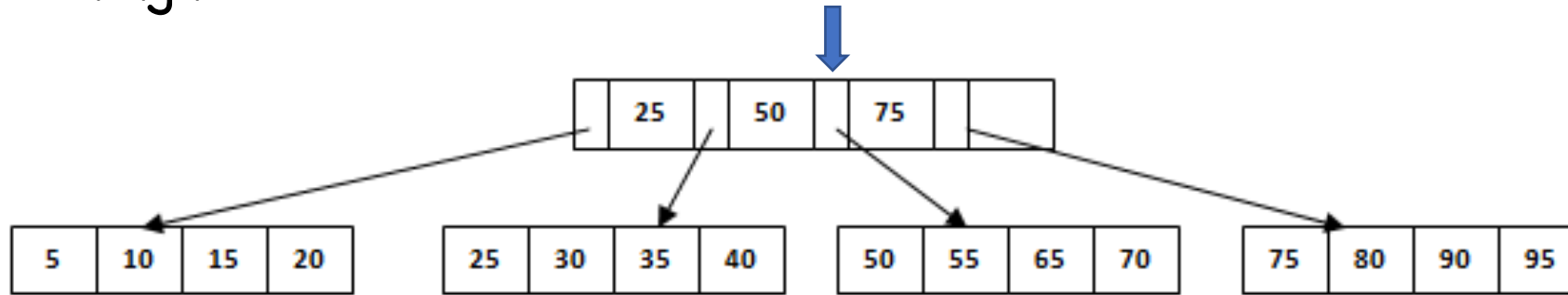
65 değerini arayalım

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



65 değerini arayalım

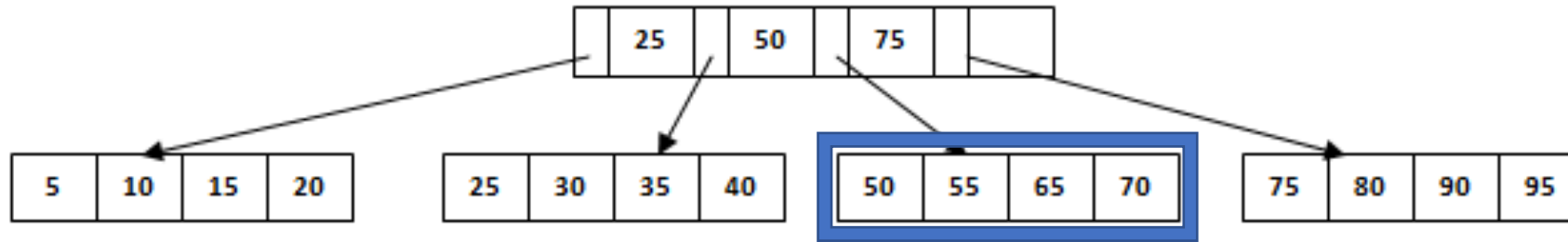
<https://www.tutorialcup.com/dbms/b-tree.htm>



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

65 değerini arayalım

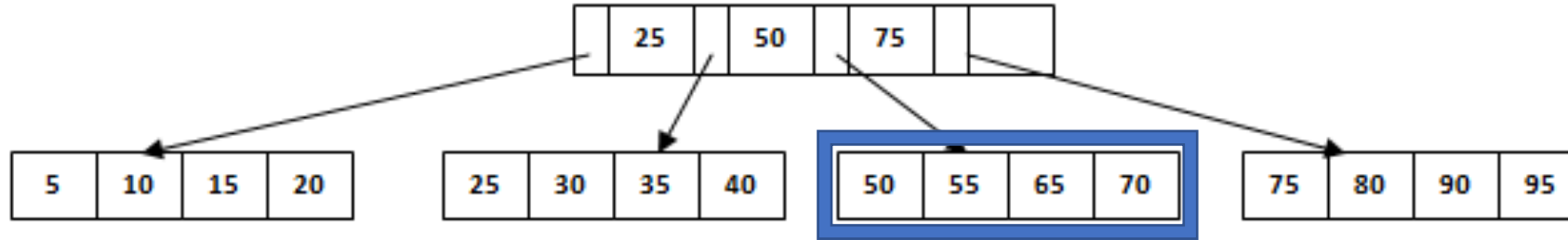
<https://www.tutorialcup.com/dbms/b-tree.htm>



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

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) olduğu için bu node içerisinde bulunan kayıtlar seri olarak (sequential) olarak taranır.

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

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 ve elemanların yarısı sırası değişmeden oluşturulan yeni yaprağa eklenir.
- Yeni oluşturulan düğümün anahtarı bir üst seviyedeki düğüme eklenir
- Üst seviyedeki düğüm dolu ise ikiye ayrılır ve ortadaki anahtar üst 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.

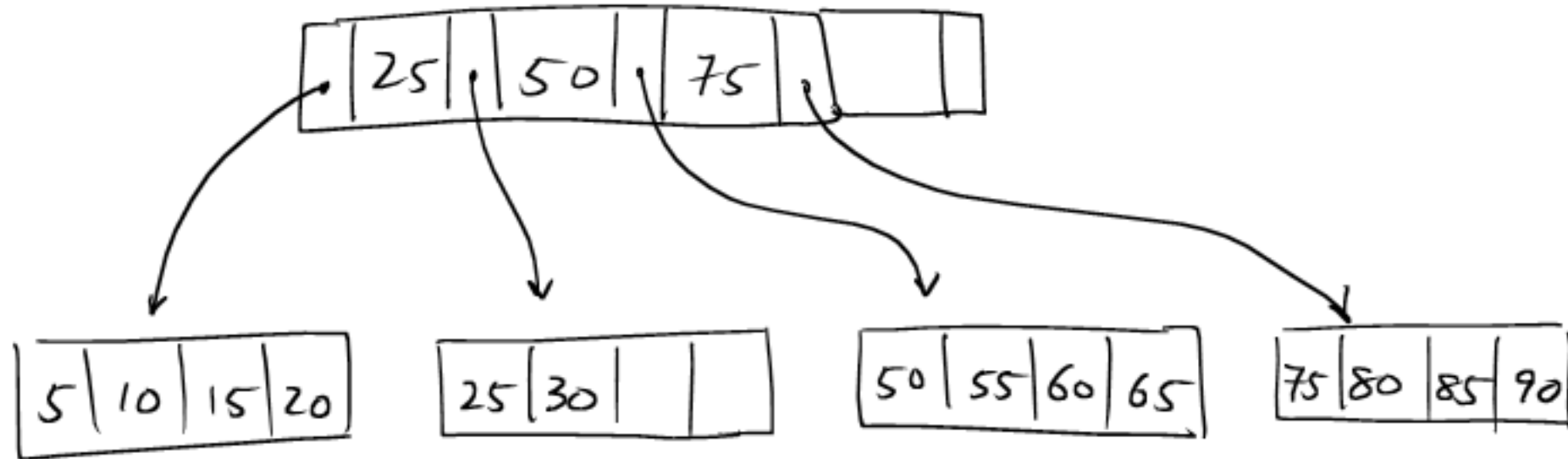
B+ ağacının düğümlerin de bulunan anahtar sayısı
 $d \leq \text{anahtar sayısı} \leq 2d$

Root (kök) düğüm istisnadır. Kökte bulunan anahtar sayısı
 $1 \leq \text{anahtar sayısı} \leq 2d$

B+ Ağaç Yapısı: Ekleme (insert) işlemi

VTYS-2019

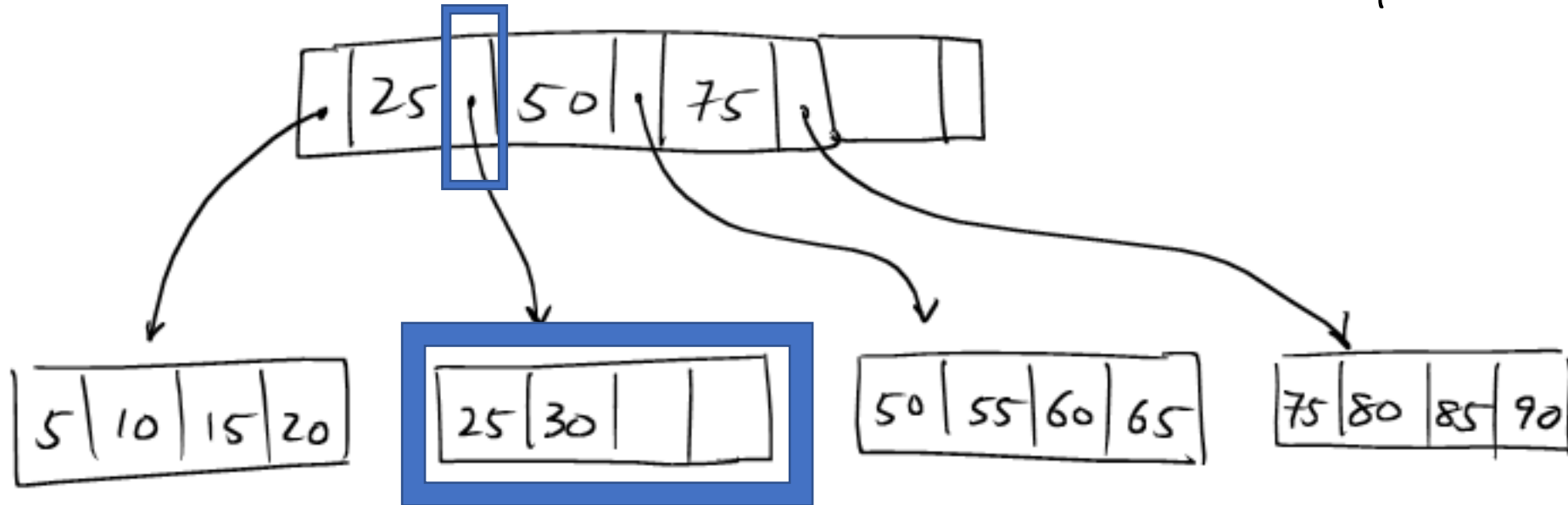
28 değerini ekleyelim



28 değerini ekleyelim

Search (root, 28)

Ekleyeceğiniz değerden küçük olan en büyük değerden sonraki pointeri bul

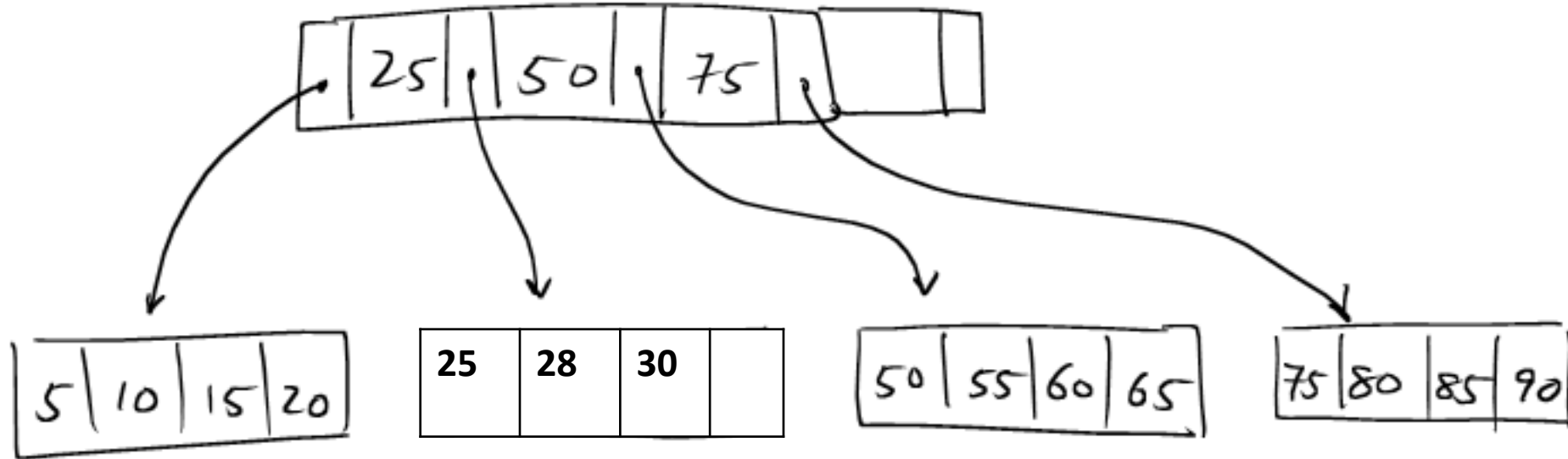


Yaprak da veri eklenecek alan var !

28 değerini ekleyelim

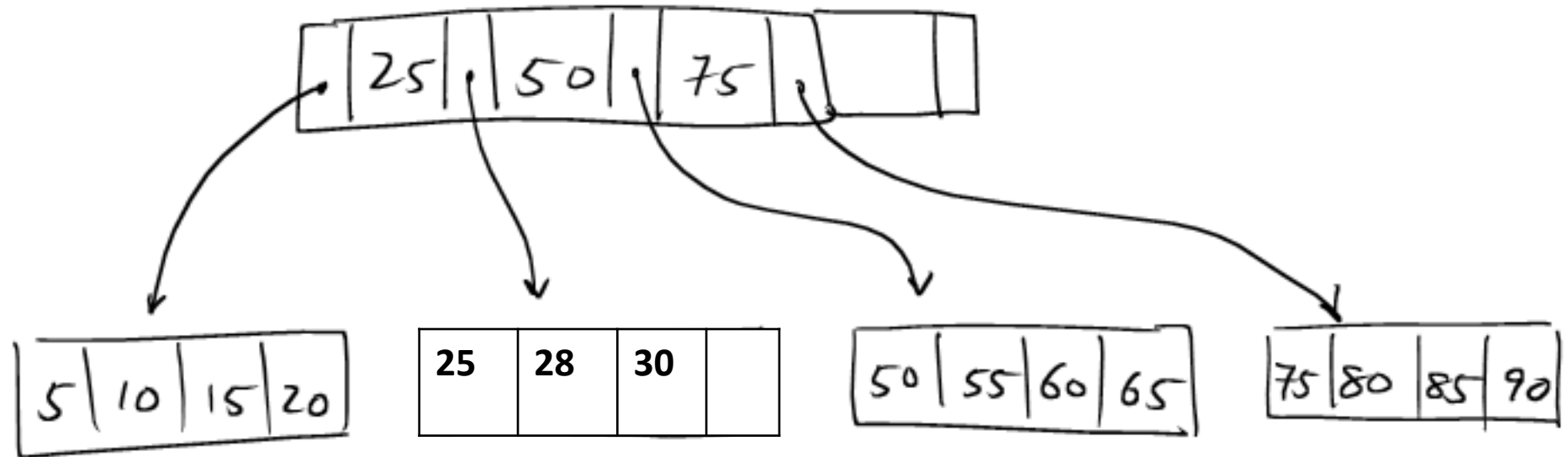
Search (root, 28)

Ekleyeceğiniz değerden küçük olan en büyük değerden sonraki pointeri bul



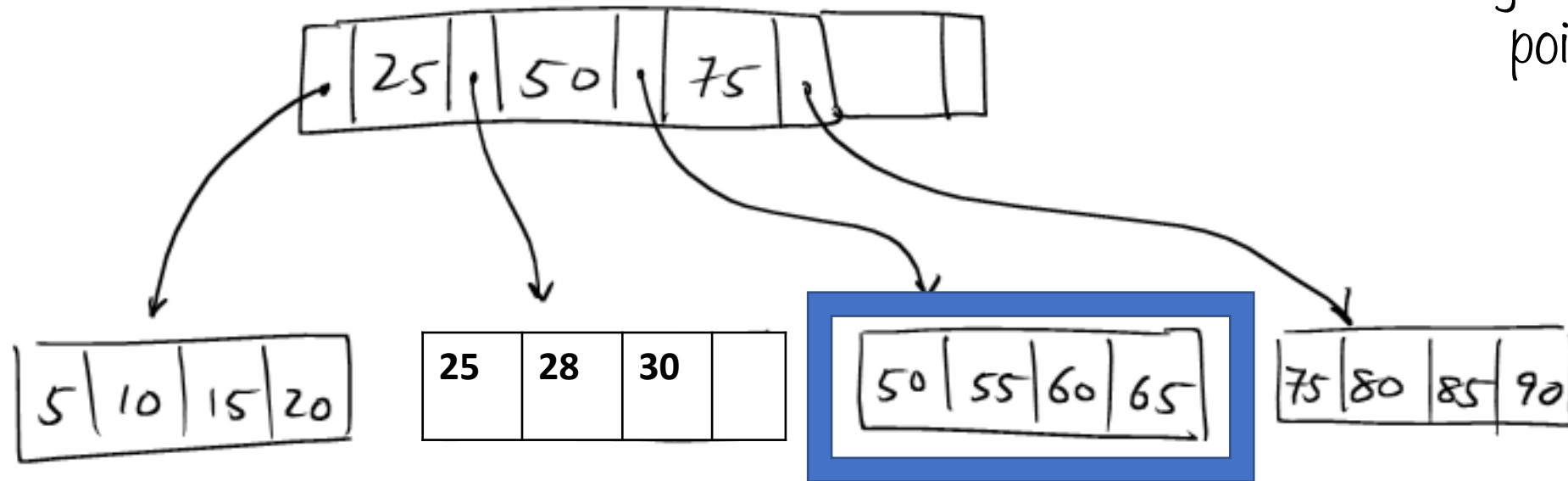
Yaprak düğümde bulunan değerler kendi içinde sıralandı

70 değerini ekleyelim



70 değerini ekleyelim

Ekleyeceğiniz
değerden küçük
olan en büyük
değerden sonraki
pointeri bul



Değerin eklenmesi
gereken düğüm bulunur.

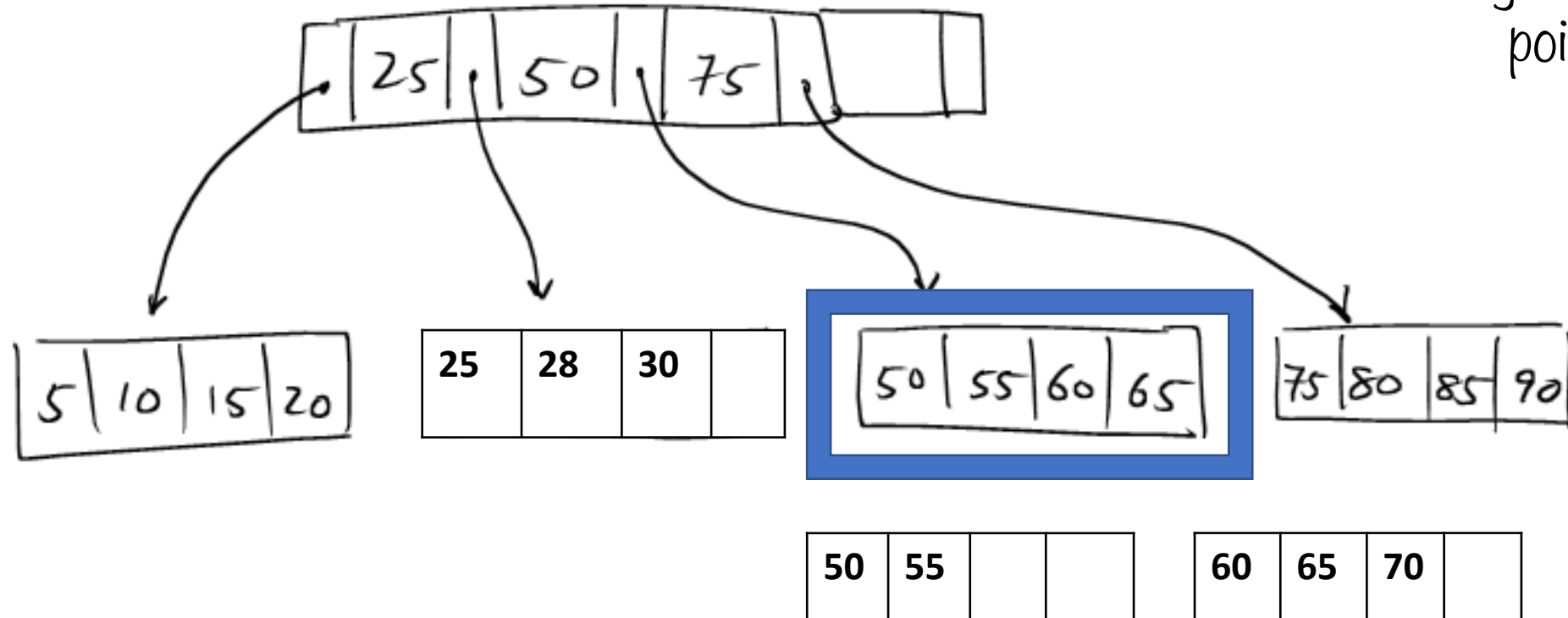
Yaprak düğümde değer eklenmesi için
eklenmesi gereken alan yok !!!!

B+ Ağaç Yapısı: Ekleme (insert) işlemi

VTYS-2019

70 değerini ekleyelim

Ekleyeceğiniz
değerden küçük
olan en büyük
değerden sonraki
pointeri bul



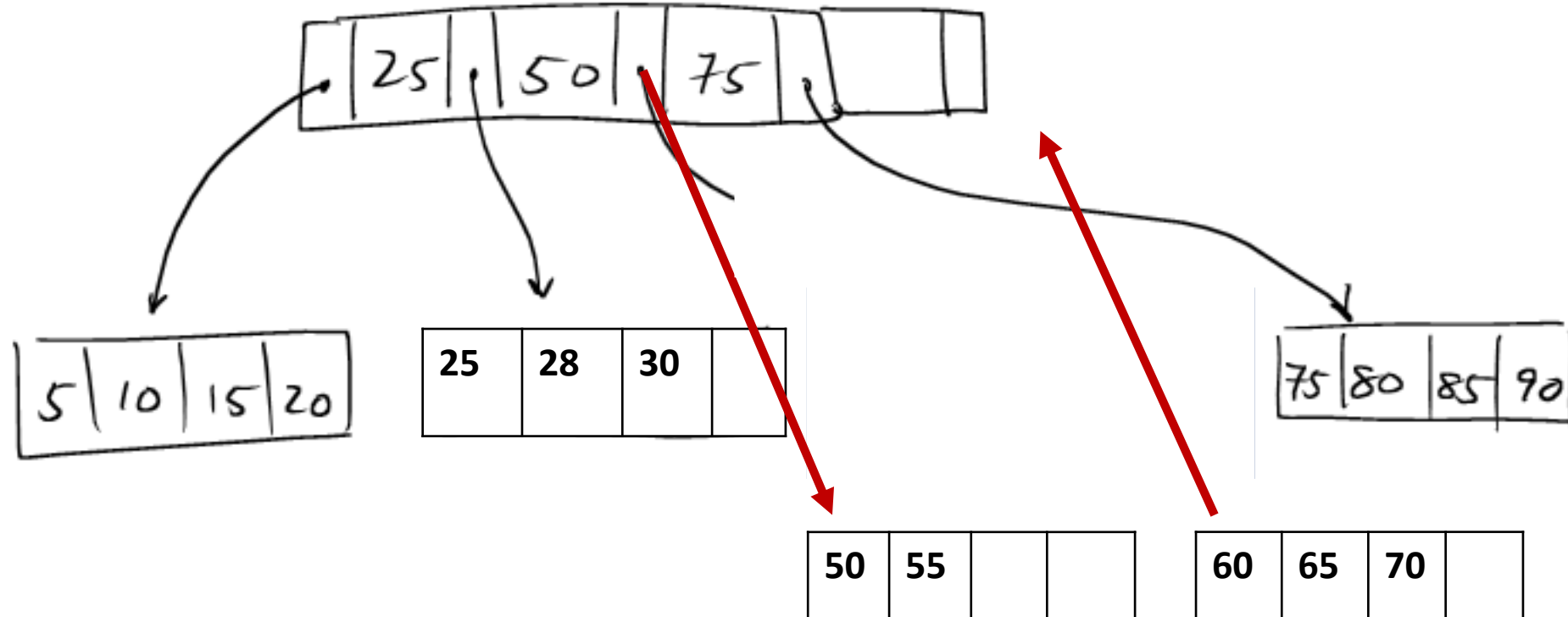
Değerin eklenmesi gereken düğüm
dolduğundan düğüm ikiye bölünür

B+ Ağaç Yapısı: Ekleme (insert) işlemi

VTYS-2019

70 değerini ekleyelim

Oluşturulan yeni düğüm için bir üst seviyede anahtar ve pointer gerekecek

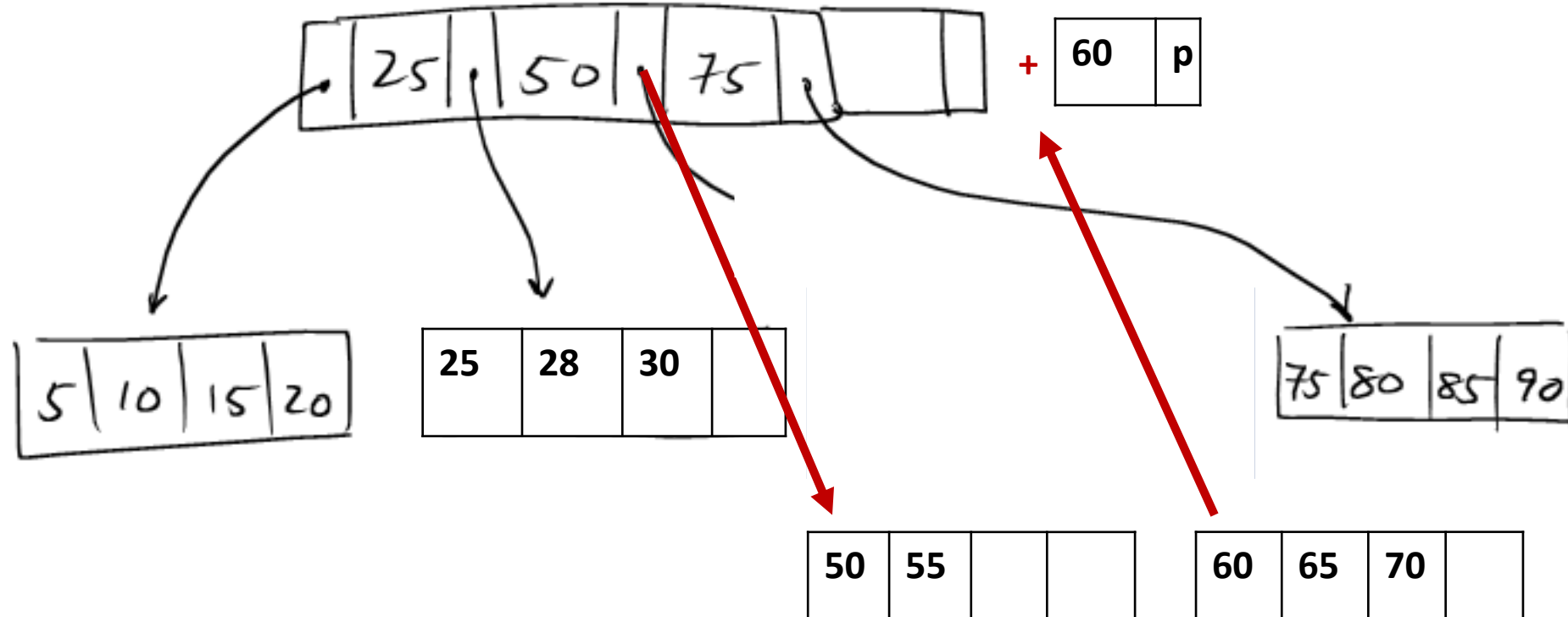


B+ Ağaç Yapısı: Ekleme (insert) işlemi

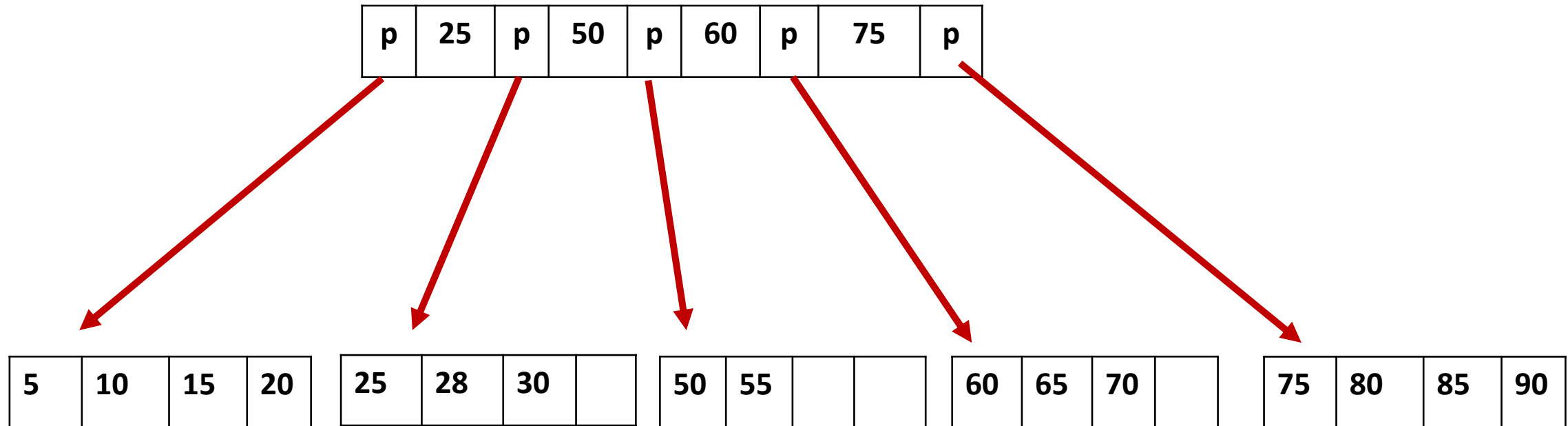
VTYS-2019

70 değerini ekleyelim

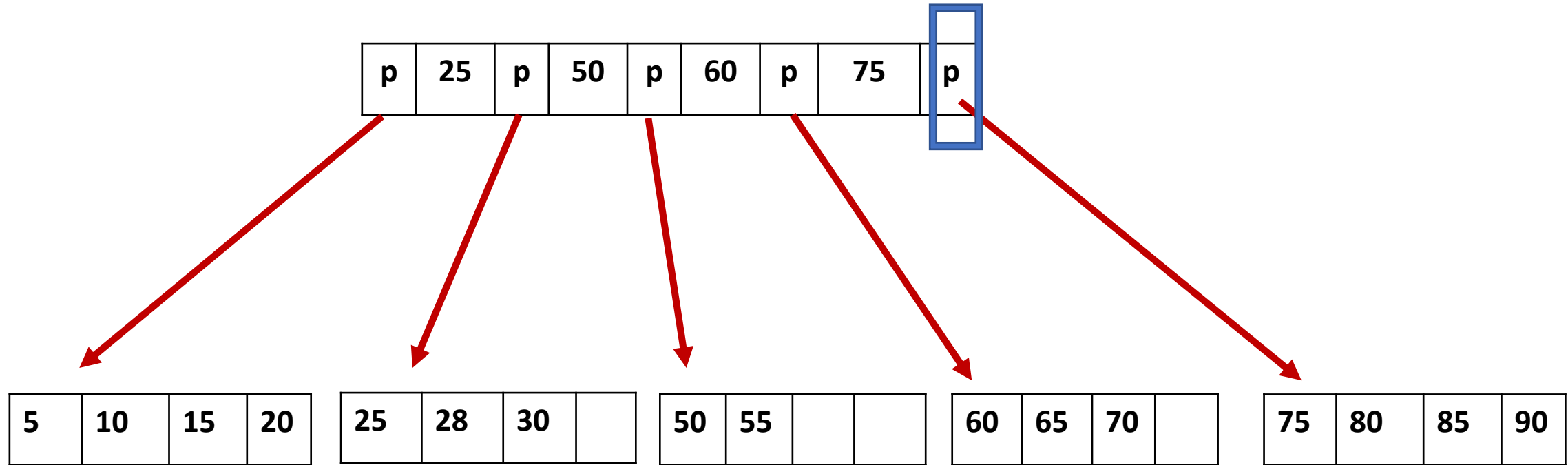
Oluşturulan yeni düğüm için bir üst seviyede anahtar ve pointer gerekecek



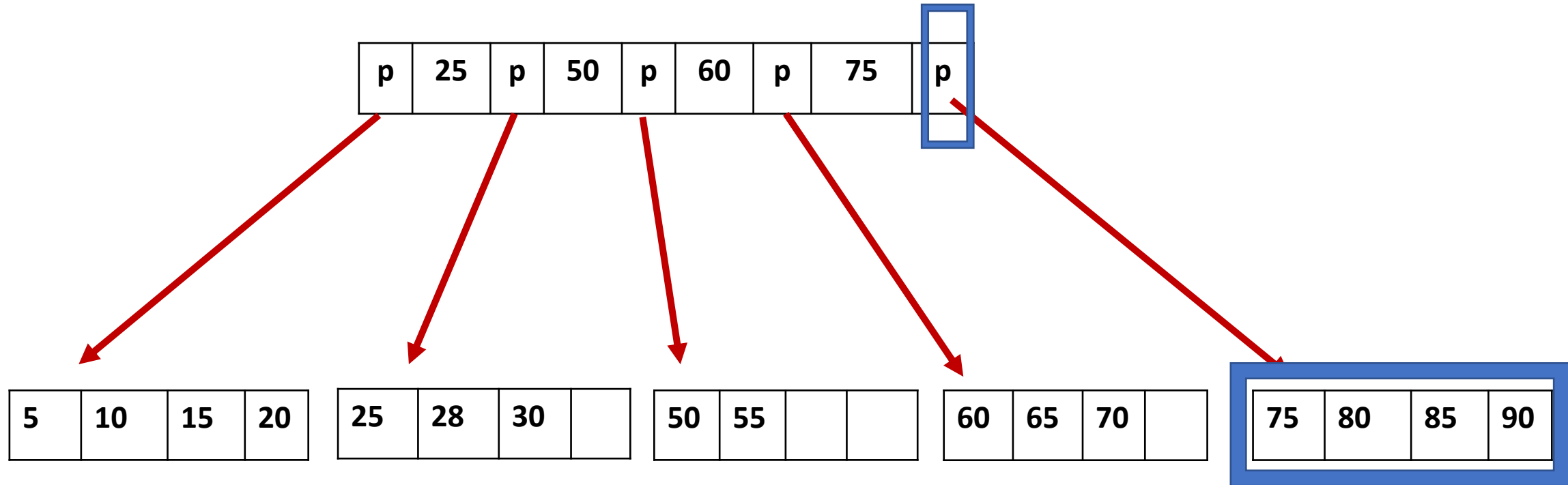
70 değerini ekledikten sonra elde edilen yeni ağaç



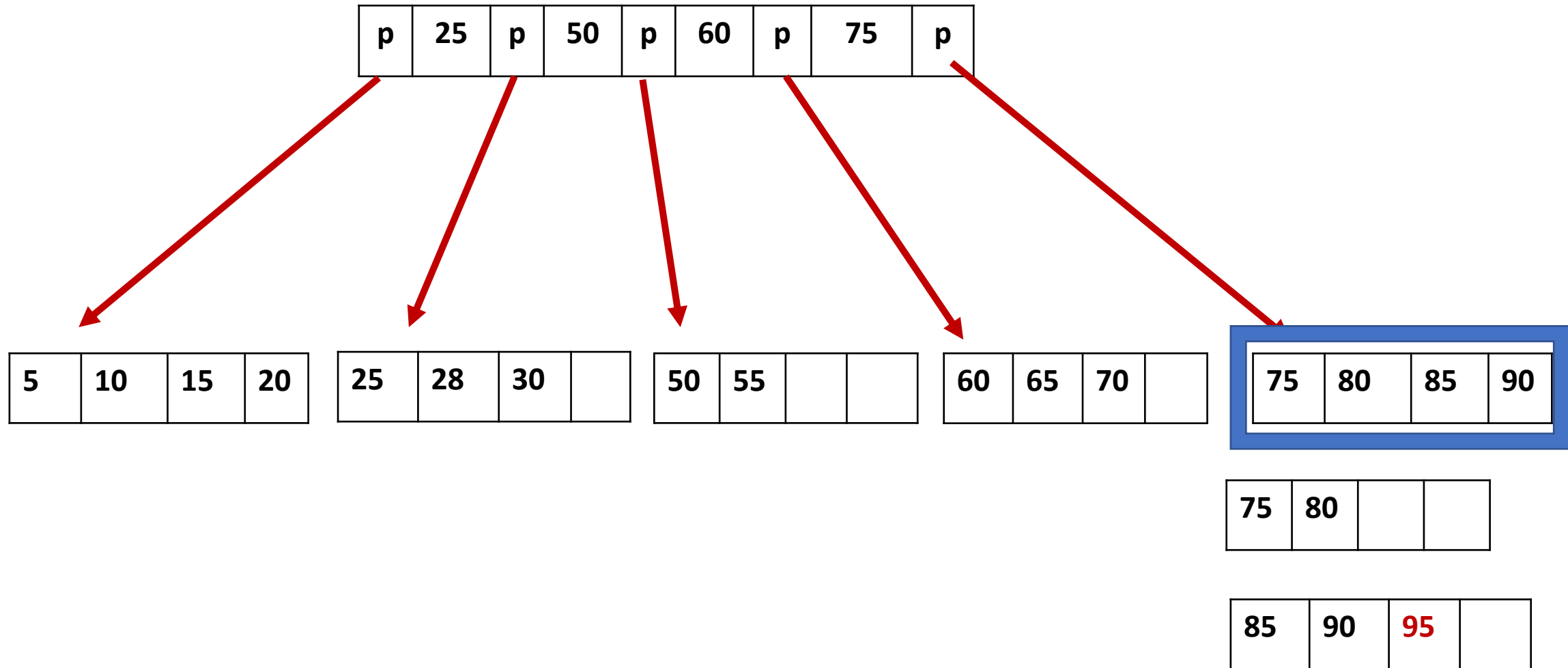
Yeni ağaca 95 değerini ekleyelim



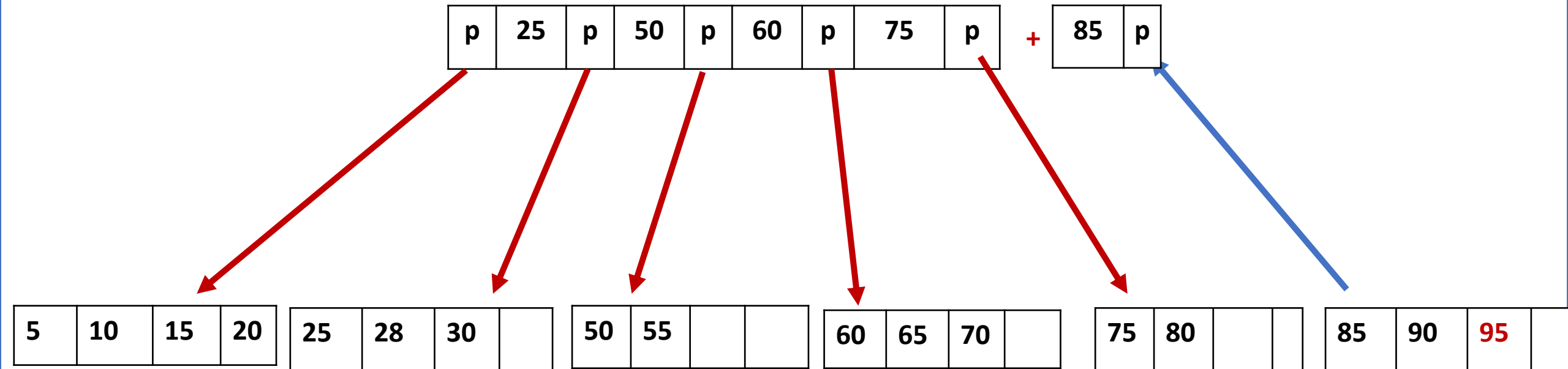
Yeni ağaca 95 değerini ekleyelim



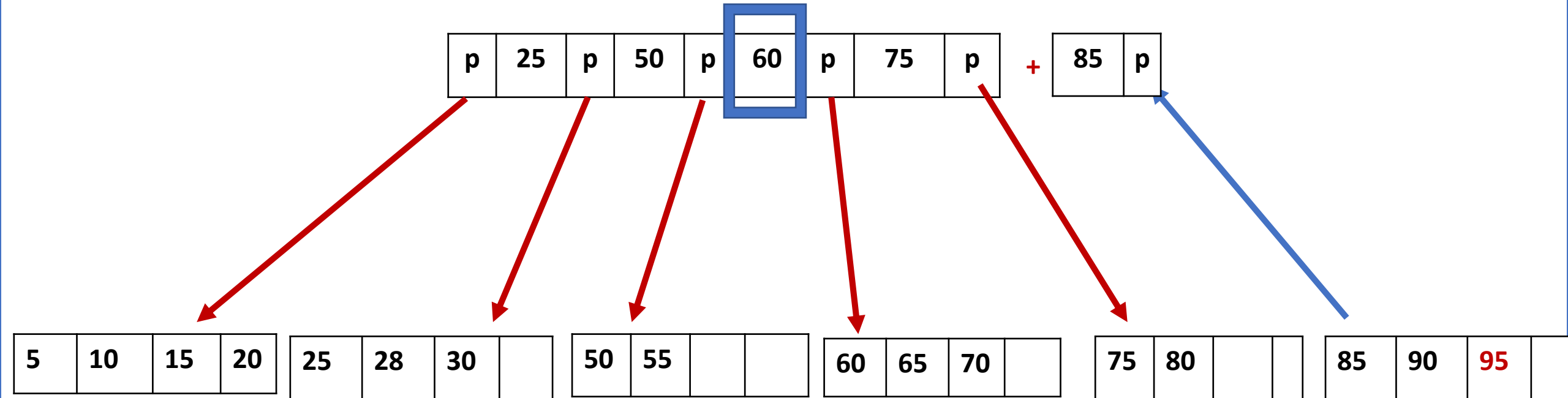
Yeni ağaca 95 değerini ekleyelim



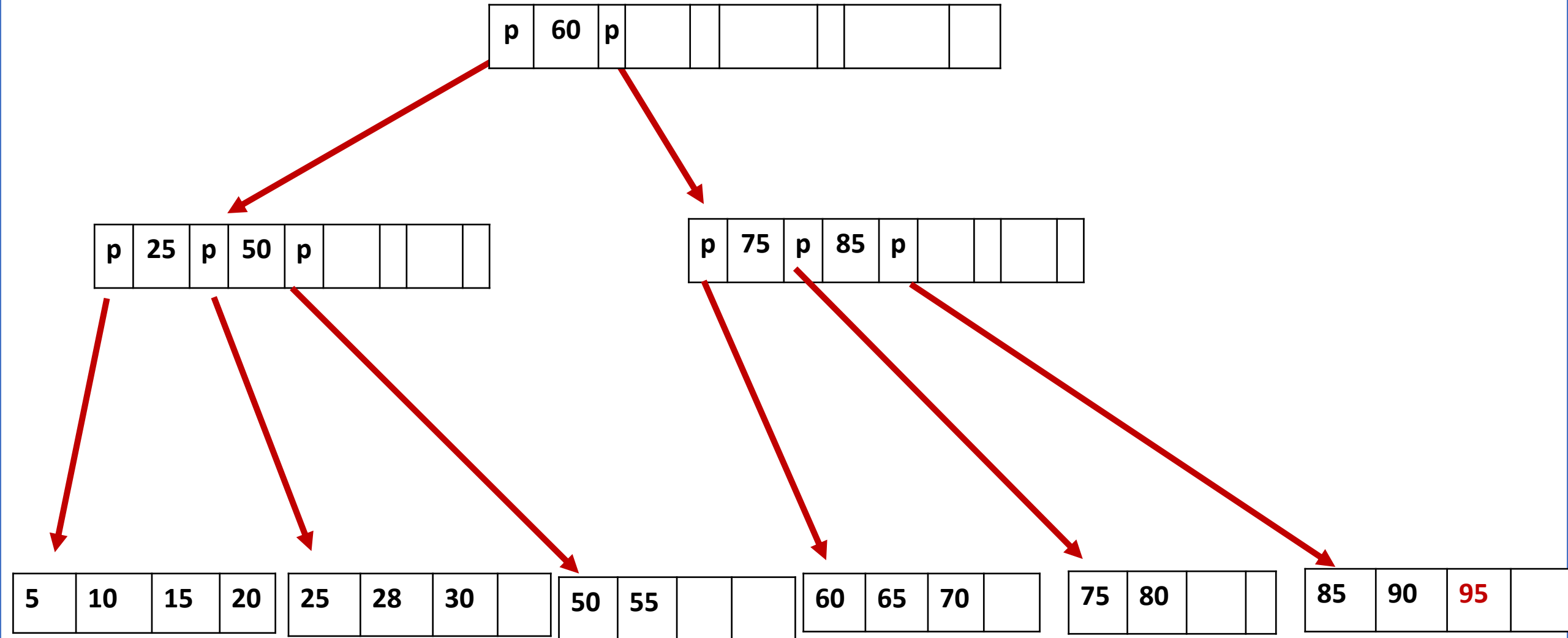
Yeni ağaca 95 değerini ekleyelim



Yeni ağaca 95 değerini ekleyelim



85 değerinin ve pointerinin eklenmesi için alan olmadığından düğüm bölünecek ve ortadaki değer ile yeni root (kök) dizin oluşturulacak



1. Silinecek kayıtın yaprak düğümü anahtar değeri yardımıyla bulunur.
2. Silme işleminin gerçekleştirileceği düğümde ağacın derecesine eşit veya büyük sayıda kayıt varsa silme işlemi gerçekleştirilir.
3. $d \leq \text{düğümdeki kayıt sayısı} \leq 2d$ şartını sağlamıyorsa düğümler aynı parent(ebeveyn) a sahip düğümler birleştirilir (tekrar dağıtma işlemi)
4. Birleştirme işlemi gerçekleştirilirken bir üst seviyede (ebeveyn) sağdaki düğümü gösteren pointer ve anahtarı silinir
5. Bu işlem root (kök) a kadar devam eder.