

DOSYA YAPILARI DERSİ (FİNAL SINAVI)

⌊

⌊-İkisinde tamsayı değerler bulundurduları kabul edilen dizi üzerinde Self Organizing Sequential Search metodlarından Move To Front metodunu kullanarak crane işlenmesi yapan fonksiyonu istediğiniz herhangi bir programlama dilinde yazınız

$$len, char^{*} list)$$
$$return len; // return the length of the list;$$

2- 48, 35, 36, 38, 34 ve 41 key değerlerini key mod 7 hash fonksiyonu kullanılarak Linear Quotient Collision Resolution Metoduyla uygun adım adım yaparak ekle. Ortalama adım sayısı hesapla.

	key
0	35
1	36
2	41
3	38
4	34
5	
6	48

- $48 \bmod 7 = 6$. adres boş eklenir.
- $35 \bmod 7 = 0$. adres boş eklenir.
- $36 \bmod 7 = 1$. adres boş eklenir.
- $38 \bmod 7 = 3$. adres boş eklenir.
- $34 \bmod 7 = 6$. adres dolu 48 var. COLLISION
 artış miktarı = $Q(34/7) \bmod 7 = 4$
 boşmu $(6+4) \bmod 7 = 3$. adres dolu 38 var.
 boşmu $(3+4) \bmod 7 = 0$. adres dolu 35 var.
 boşmu $(0+4) \bmod 7 = 4$. adres boş. EKLENEK
- $41 \bmod 7 = 6$. adres dolu 48 var. COLLISION
 artış miktarı = $Q(41/7) \bmod 7 = 5$
 boşmu $(6+5) \bmod 7 = 4$. adres dolu 34 var.
 boşmu $(4+5) \bmod 7 = 2$. adres boş. EKLENEK

Ortalama adım sayısı = $\frac{48 \quad 35 \quad 36 \quad 38 \quad 34 \quad 41}{1 + 1 + 1 + 1 + 4 + 3} = \frac{11}{6} = 1,8$

3

ikinci sorudaki key değerlerini key mod 7 hash fonksiyonu ile Brent's Method Collision Resolution kullanarak ekle. Ortalama adım sayısı?

48, 35, 36, 38, 34 ve 41

	Key
0	35
1	36
2	
3	38
4	
5	
6	48

• $48 \bmod 7 = 6$. adres boş eklenir.

• $35 \bmod 7 = 0$. adres boş eklenir.

• $36 \bmod 7 = 1$. adres boş eklenir.

• $38 \bmod 7 = 3$. adres boş eklenir.

• $34 \bmod 7 = 6$. adres dolu 48 var. **COLLISION**

* 2. soruda Linear Quotient'a göre 4 adımda ekleneceğini gördük. 's' değeri 4 olarak hesaplanmış oldu.

(1. kontrol 6. adres dolu, 2. kontrol 3. adres dolu, 3. kontrol 0. adres dolu, 4. kontrol 4. adres boş) olarak hesaplanır.

Pij değer olarak. $i+j$ toplam s'den küçük $i+j=2$ $i+j=3$ (4'den küçük) kontrol edilir.

34

$q(48) = 6$ artırmak

$(6+6) \bmod 7 = 5$. adres boş

$P(1,1)$

⇒ $P(1,1)$ boş. 6. kayıttaki 48 $P(1,1)$ adresine taşınır.

⇒ 34'de boş hale gelen P1 adresine yani 6. adrese eklenir.

Yeni Tablo

0	35
1	36
2	
3	38
4	41
5	48
6	34

• $41 \bmod 7 = 6$. adres dolu 34 var. **COLLISION**

artırmak $= Q(41/7) \bmod 7 = 5$

başını $(6+5) \bmod 7 = 4$. adres boş.

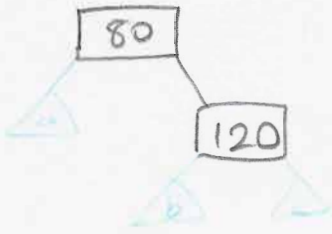
s değeri 2" olduğu için Linear Quotient'a göre ekleme yapılır. (Taşıma olmaz.)

Ortalama adım sayısı = $\frac{48 \quad 35 \quad 36 \quad 38 \quad 34 \quad 41}{2 + 1 + 1 + 1 + 1 + 2} = \frac{8}{6} = 1,3$

4 80, 120, 130, 50, 70, 105, 90 ve 95 IPR Tree ile ekleyin.

80

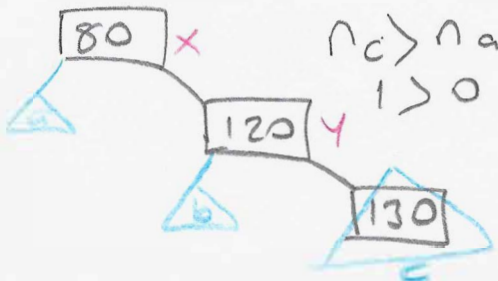
- 80'nin kök olduğu ağacın sağ alt ağactaki düğüm sayısı ile sol alt ağactaki düğüm sayısı eşit 0. DENGELİ



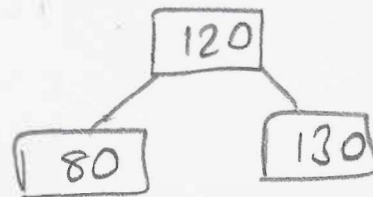
ACIKLAMA

Ağaca ekleme yapılırken denge kontrolü yapılır. Herhangi bir anda n_b veya n_c değerlerinden biri n_a 'den büyük olduğu durumda Dengesizlik olduğu için ilgili ağacın notasyon uygulanır. Aksi durumda ağac DENGELİDİR.

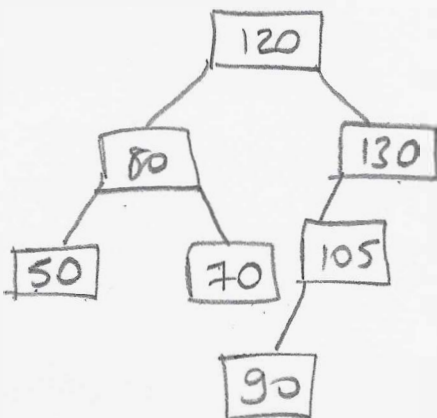
- 120'nin kök olduğu ağacın sağ ve sol alt ağaclardaki düğüm sayısı eşit. DENGELİ
- 80'nin kök olduğu ağacın sağ alt ağacın fazlalaştığını $n_a = 0, n_b = 0, n_c = 0$ ağac denge



- 130'un kök olduğu ağacın denge
- 120'nin kök olduğu ağacın $n_a = 0, n_b = 0, n_c = 0$ denge
- 80'nin kök olduğu ağacın $n_a = 0, n_b = 0, n_c = 1$ denge



denge hâle geldi.



- 50'yi ekledik hepsi denge
- 70' " " "
- 105' " " "
- 90' ekleyince kontrol etmemiz lazım.

5

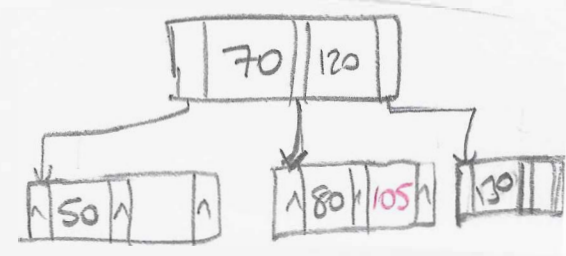
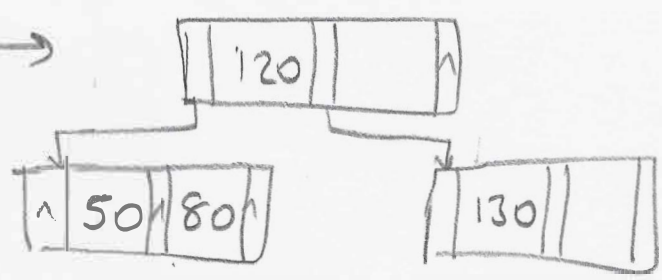
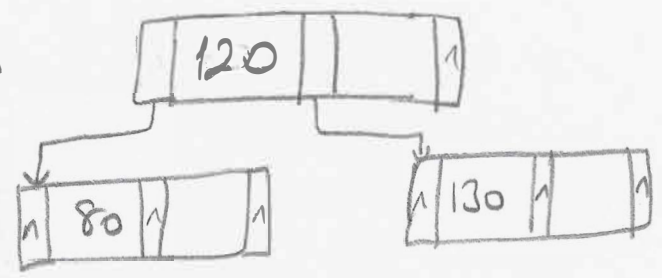
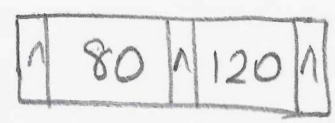
80, 120, 130, 50, 70, 105, 90 ve 95 capacity order
digeri 1 olacaktir

Capacity order 1.

B Tree ile ekle.

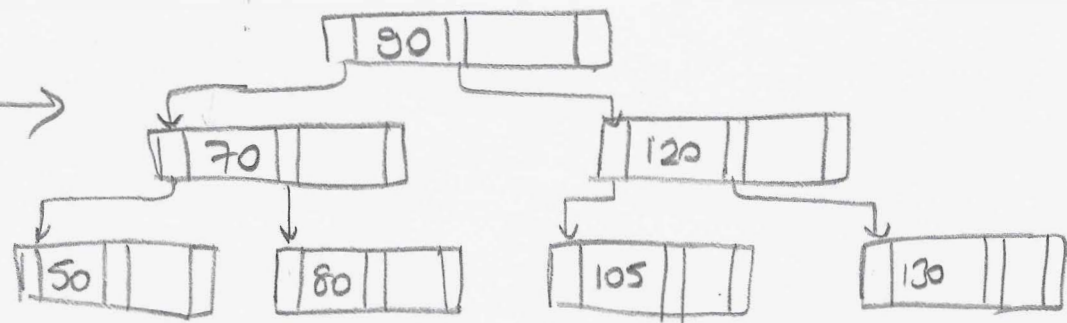
$$2 * 1 = 2 \text{ kayıt}$$

$$2 * 1 + 1 = 3 \text{ de diyannde ciken link sayisi}$$

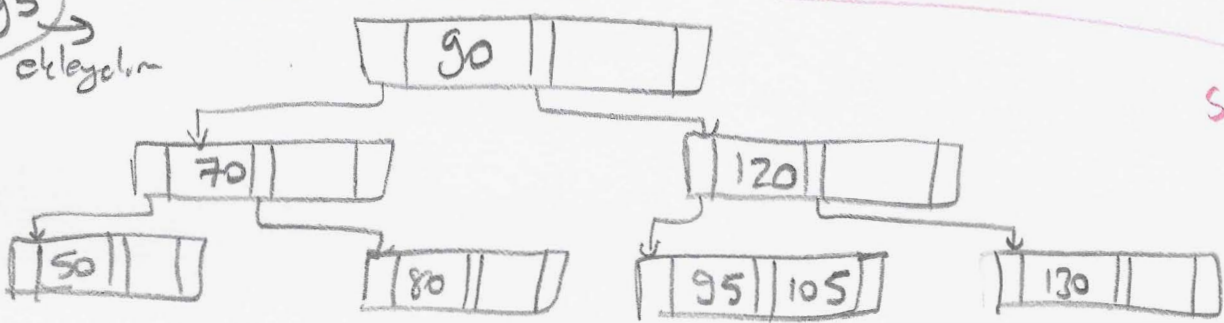


90 ->

90



5
ekleyelim



son hali