Insertion Sort Sorusu:

[22,27,16,2,18,6]

1. Adım: 22| 27,16,2,18,6

22’yi öncelikle bir gruba ayırdık. 27 sayısını bu gruba dahil ediyoruz. Büyük olan sayı sağda kalır.

1. Adım: 22, 27 | 16,2,18,6

16 sayısını da bu gruba dahil ediyoruz. En küçük olduğundan en sola geçer.

1. Adım: 16, 22, 27| 2,18,6

2 rakamını da bu gruba dahil ediyoruz. En küçük olduğundan en sola geçer.

1. Adım: 2, 16, 22, 27| 18,6

18 sayısını da bu gruba dahil ediyoruz. 16 dan sonra en küçük olduğu için 16 ve 22 arasına koyarız.

1. Adım: 2, 16, 18, 22, 17 | 6

6 rakamını da bu gruba dahil ediyoruz. 2 den sonra en küçük olduğu için 2 ve 16 arasına koyarız.

1. Son hali: 2, 6, 16, 18, 22, 17

Big O notation olarak O (N^2) gösterebiliriz. Sorudaki açıklamaya göre 18 ortadaki sayı olduğundan “Average Case” e girer.

Selection Sort Sorusu:

[7,3,5,8,2,9,4,15,6] i = 0 yani 0. Eleman ile dizideki en küçük elemanı yer değiştiriyoruz.

[2,3,5,8,7,9,4,15,6] i’yi 1 arttırıyoruz. i=1 Şimdi ise 1. Eleman ile dizideki diğer en küçük elemanı yer değiştiriyoruz. Ama 3 ten küçük eleman olmadığından 3 sabit kalıyor.

[2,3,5,8,7,9,4,15,6] i’yi 1 arttırıyoruz. i=2 Şimdi ise 2. Eleman ile dizideki diğer en küçük elemanı yer değiştiriyoruz.

[2,3,4,8,7,9,5,15,6] i’yi 1 arttırıyoruz. i=3 Şimdi ise 3. Eleman ile dizideki diğer en küçük elemanı yer değiştiriyoruz.

4. Adımdaki hali: [2,3,4,5,7,9,8,15,6]

Binary Search Tree Sorusu:

[7,5,1,8,3,6,0,9,4,2]

1. Adım : Root 7’dir. 5, 7’den küçük olduğu için sol tarafına geçer.

makas içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Adım : 9, 7’den büyük olduğu için sağ tarafına geçer. 8 den de büyük olduğu için 8in sağ tarafına geçip sağ child node olur.

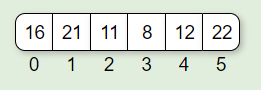
6. Adım: 0, 7den küçük olduğundan soldan devam eder. 5ten de küçük olduğundan 5in de sol nodeuna bakarız. 1den küçük olduğu için 1’in sol child nodeu olur.

5. Adım: 6, 7den küçük olduğundan soldan devam eder. 5ten büyük olduğundan 5in de sağ nodeu olur.

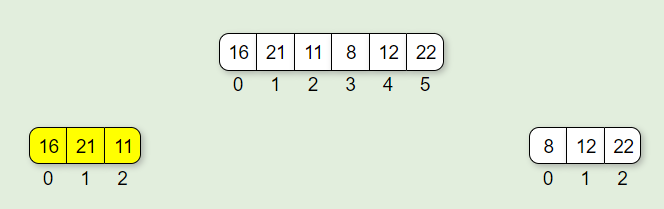
4. Adım: 3, 7den küçük olduğundan soldan devam eder. 5ten de küçük olduğundan 5in de sol nodeuna bakarız. 1den büyük olduğu için 1’in sağ child nodeu olur.

1. Adım : 8, 7’den büyük olduğu için sağ tarafına geçer. Ve 7 yani root’un child node u olur.
2. Adım : 1, 7’den küçük olduğu için sol tarafa bakarız. 5’ten de küçük olduğu için 5 in soluna geçer.

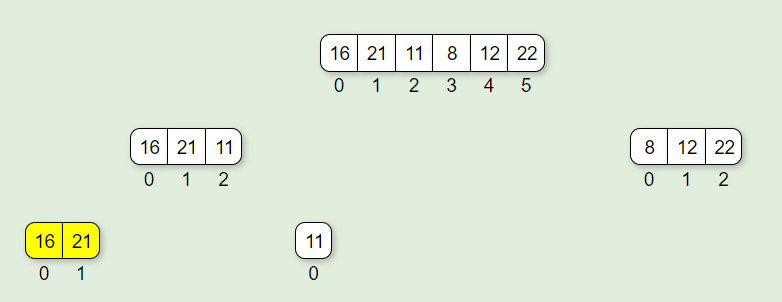
MERGE SORT SORUSU



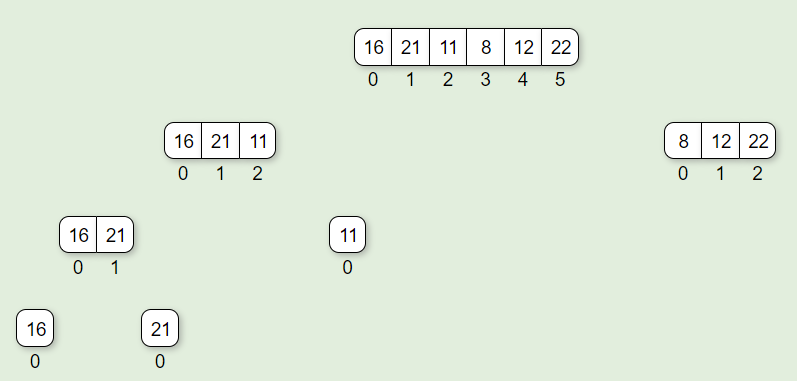
Tüm diziyi seçiyoruz ve mümkün olduğunca eşit olarak bölmeye çalışıyoruz.



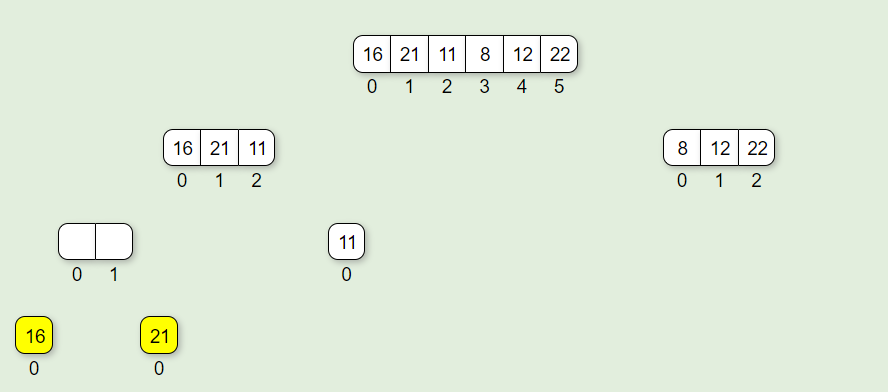
Böldükten sonra sol alt diziyi seçiyoruz ve onun için de aynı adımı yapıyoruz.(mümkün olduğunca eşit bölelim)



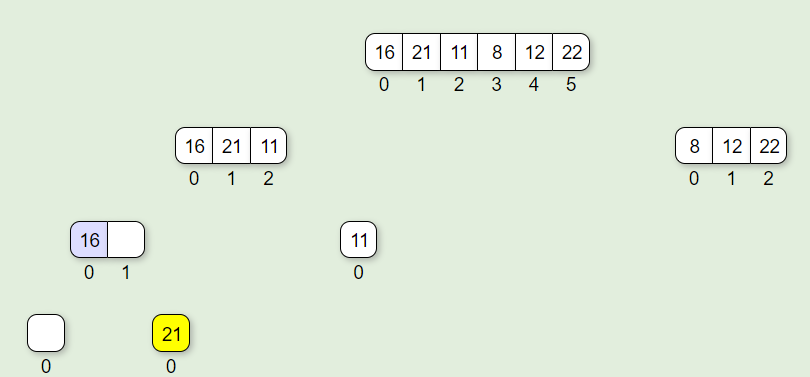
Böldükten sonra sol altdizinin sol altdizisini seçiyoruz ve onun için de aynı işlemi yapıyoruz.(mümkün olduğunca eşit bölelim)



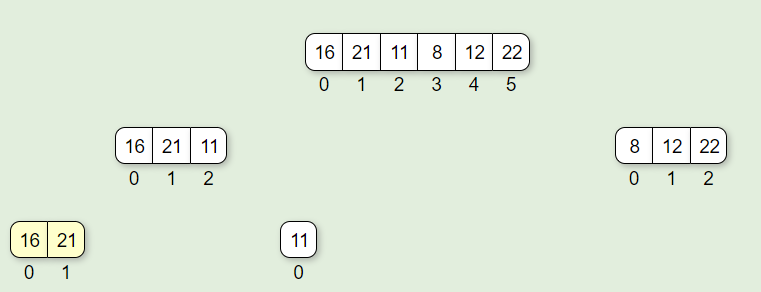
Dizimizin sol kısmını olabildiğince böldük. Şimdi bölünmüş dizi parçasının boyutunu kontrol ediyoruz ve her bir alt dizi için uzunluğun 1 olduğunu görüyoruz.(16 ve 21)



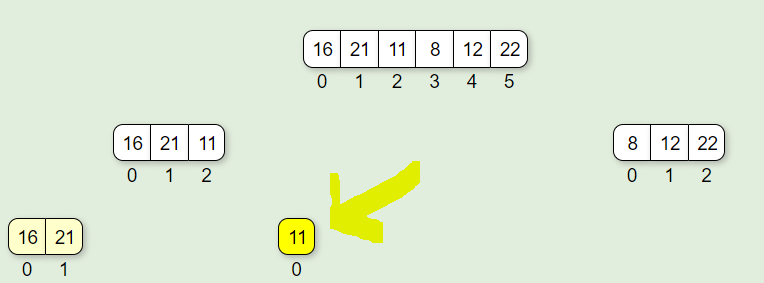
Değerleri sıralayarak birleştirme işlemine başlayacağız. Min'i seçeceğiz. bunlardan iki değer arasında bir değer alırız ve dizileri tekrar birleştiririz. Yani, bu diziye göre daha yüksek dizinin soluna 16 koyacağız.



Bundan sonra, bir liste boşaldığında, kalan dizinin tüm değerlerini sıralanmış diziye kopyalayacağız.



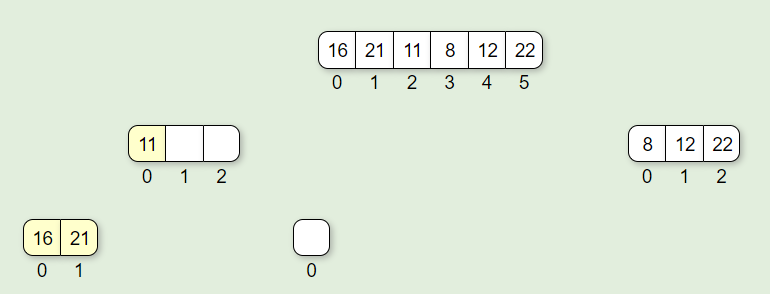
Dizinin bu bölümünde birleştirme işlemi tamamlanacaktır.

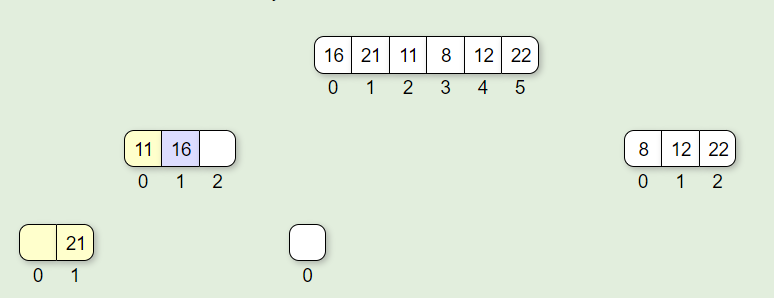


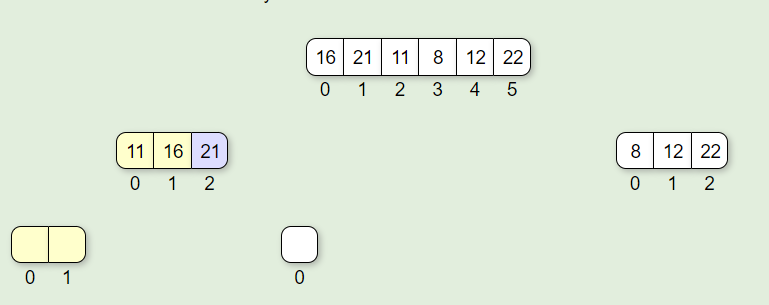
Dizi uzunluğu 1 olduğundan birleştirme işlemine de hazır demektir.

Her listeden en küçük değeri seçeceğiz.

Bu kısımda 11'i seçip üst dizinin başlangıcını koyacağız.

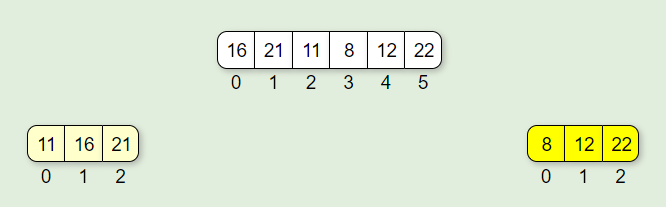




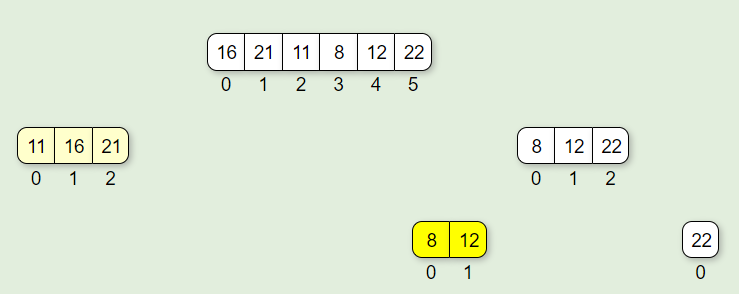


Bundan sonra, bir liste boşaldığında, kalan dizinin tüm değerlerini sıralanmış diziye kopyalayacağız.

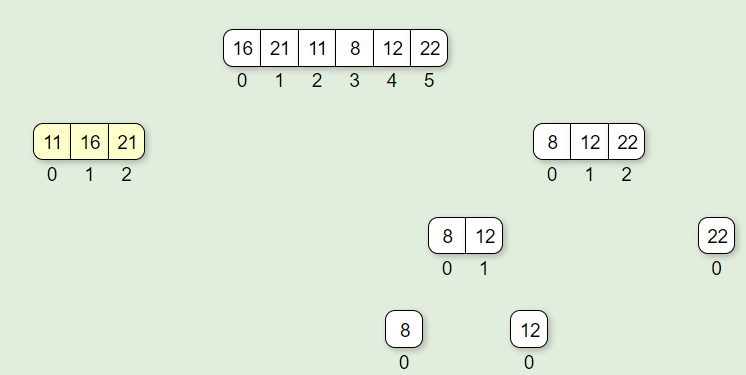
Birleştirme işlemi yapılır. Aynı işlemi sağ alt dizi için de yapacağız.



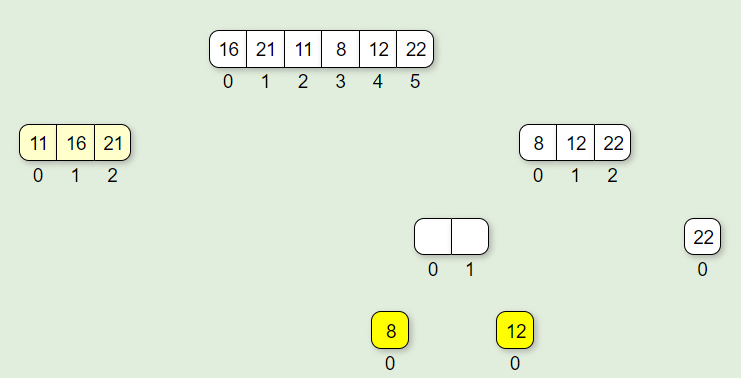
Sağ alt diziyi seçiyoruz.



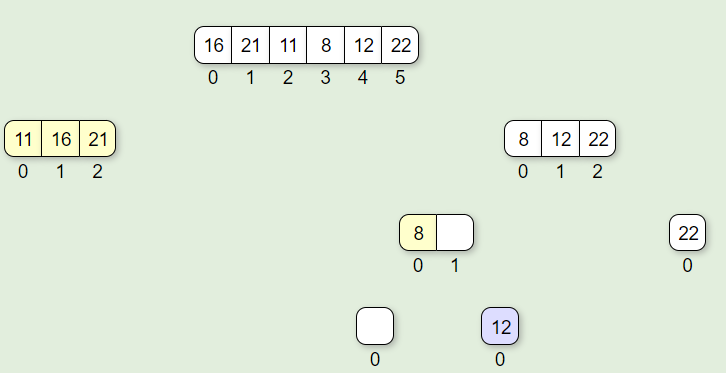
Sağ alt diziyi seçiyoruz ve bunun için aynı adımı yapıyoruz. (mümkün olduğunca eşit olarak bölün)



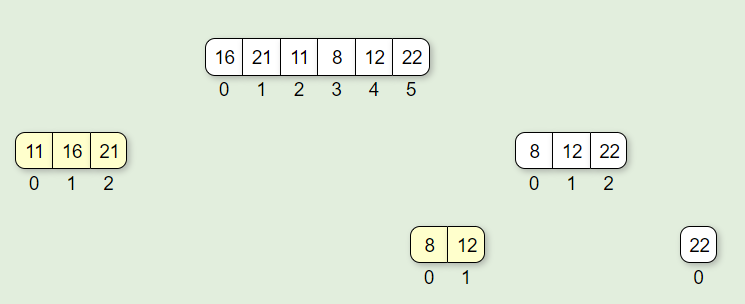
Böldükten sonra sağ alt dizinin sol alt dizisini seçip aynı işlemi onada yapıyoruz.(mümkün olduğunca eşit bölelim)



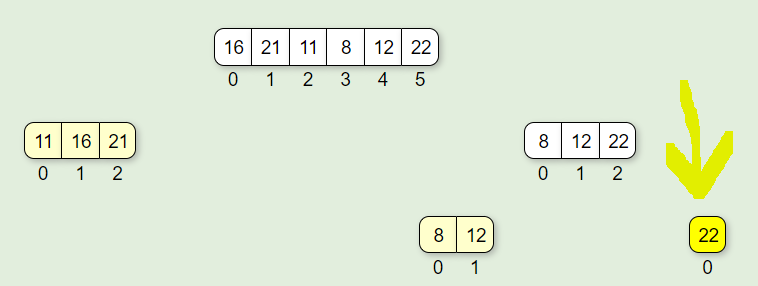
Değerleri sıralayarak birleştirme işlemine başlayacağız. Min'i seçeceğiz. bunlardan iki değer arasında bir değer alırız ve dizileri tekrar birleştiririz. Yani, bu diziye göre daha yüksek dizinin soluna 8 koyacağız.



Bundan sonra, bir liste boşaldığında, kalan dizinin tüm değerlerini sıralanmış diziye kopyalayacağız.



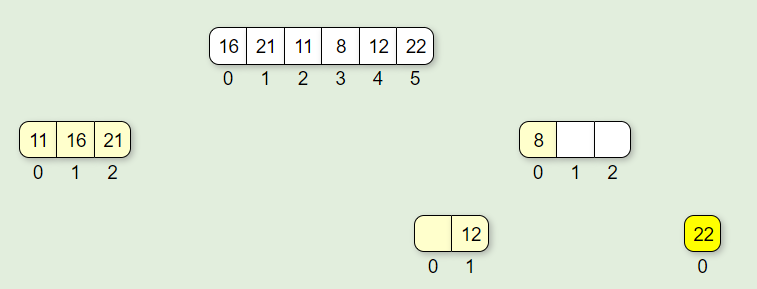
Dizinin bu bölümünde birleştirme işlemi tamamlanacaktır.

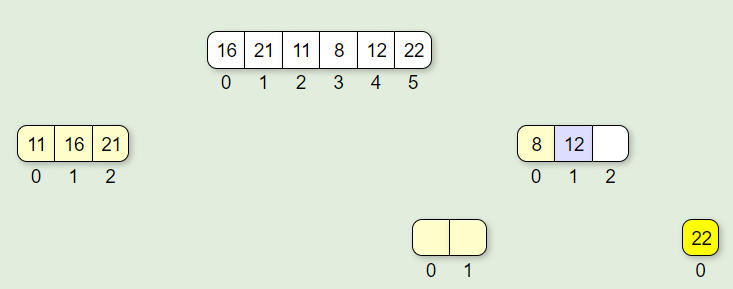


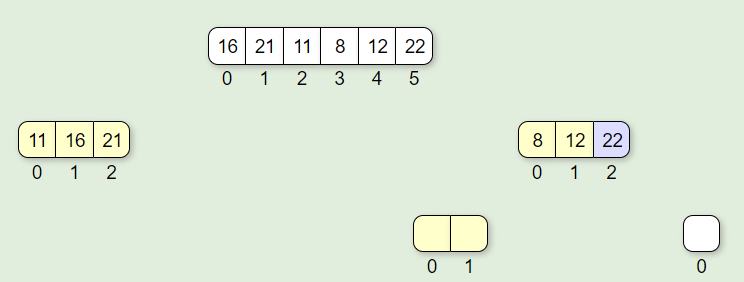
Dizi uzunluğu 1 olduğundan birleştirme işlemine de hazır demektir.

Her listeden en küçük değeri seçeceğiz.

Bu kısımda 8'i seçip üst dizinin başlangıcını koyacağız.

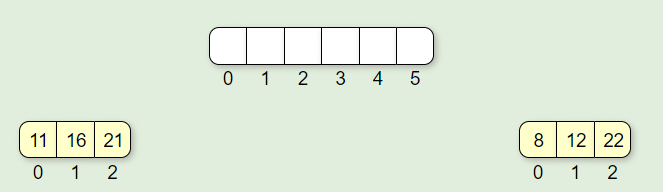






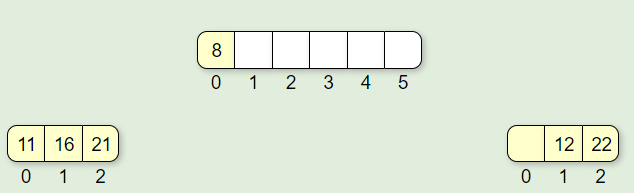
Bundan sonra, bir liste boşaldığında, kalan dizinin tüm değerlerini sıralanmış diziye kopyalayacağız.

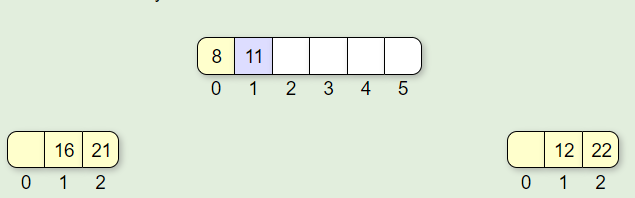
Birleştirme işlemi yapılır.

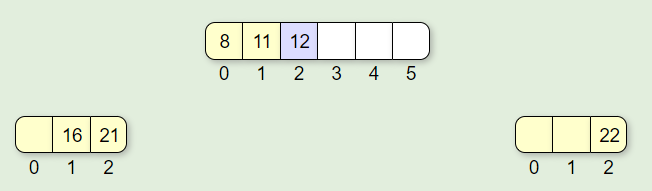


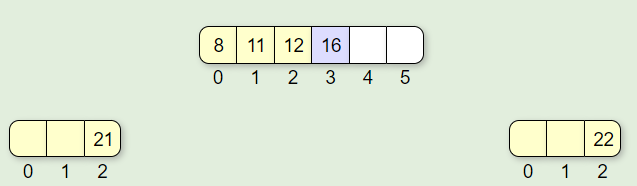
Dizileri tekrar bir araya getireceğiz. Her listeden en küçük değeri seçeceğiz.

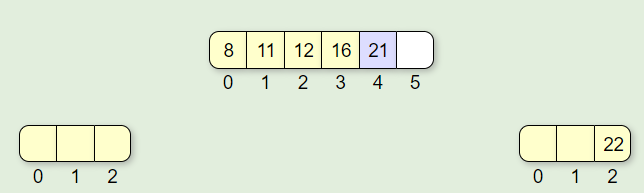
Böylece dizinin başına ilk 8 değerini koymaya başlayacağız.



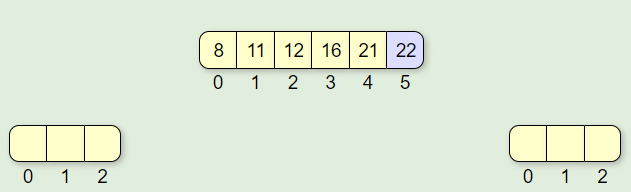




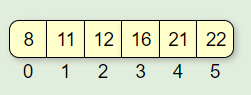




Bundan sonra, bir liste boşaldığında, kalan dizinin tüm değerlerini sıralanmış diziye kopyalayacağız.



Birleştirme tamamlandı.



Dizimizin son versiyonu. Sıralama tamamlandı!

**MERGE SORT BIG O NOTASYONU**

Buradaki fikir, daha küçük dizileri sıralamak ve ardından bu dizileri sıralı şekilde bir araya getirip birleştirmektir. Bu algoritmada “recursion” dediğimiz özyinelemeden de yararlanılır.

**Worst-Case:** O(n log n) - n tane öğeyi ayırmalı ve sonra onları yeniden birleştirmeliyiz. Sıralı alt dizileri oluştururken, sayılarını da efektif bir şekilde ikiye katlıyoruz (1'den 2, 2'den 4, …).

**Best-Case:** Ω(n log n) - Ancak yine de bu algoritmada diziyi bölmek ve yeniden birleştirmemiz gerekmektedir.