



# Bölüm 8. Ayrık Küme

Olcay Taner Yıldız

2014



## Ayrık Küme Tanımı

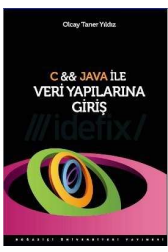
Temel Ayrık Küme İşlemleri

---

Uygulama: Şifre Çözme

---

# Ayrık Küme Tanımı

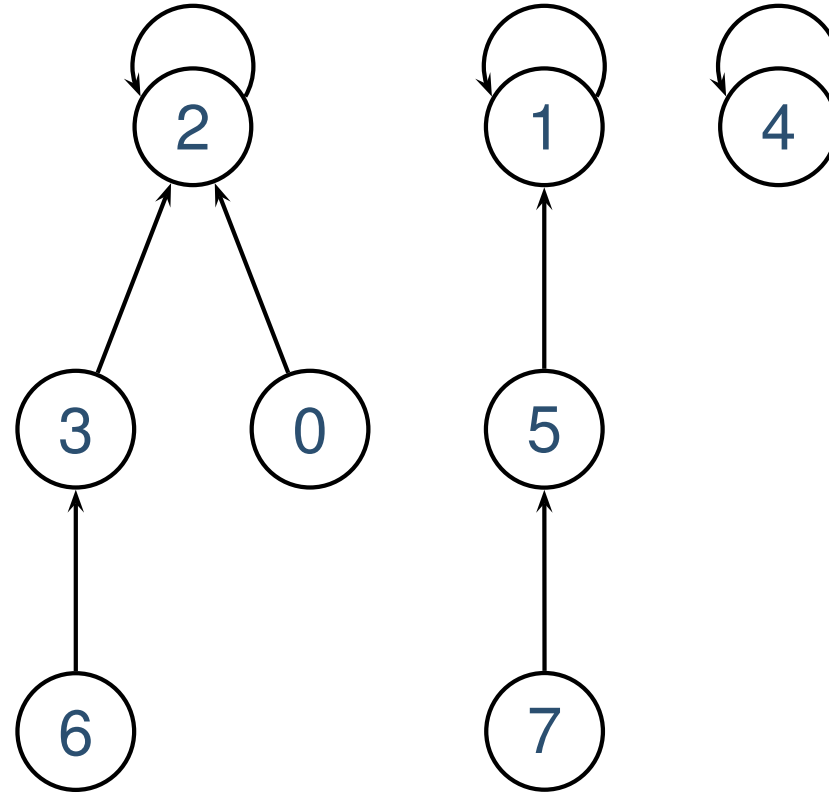


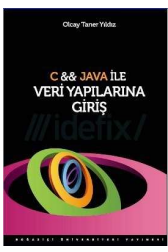
# Sekiz ögeden oluşan bir ayırık küme veri yapısının gösterimi

Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme





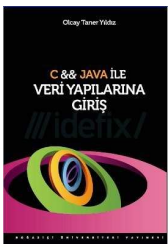
# *Sekiz öğeden oluşan bir ayrık kümeler grubunun üç ana parçasının gösterimi*

Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme

	0	1	2	3	4	5	6	7
İçerik	0	1	2	3	4	5	6	7
Ebeveyn	2	1	2	2	4	1	3	5
Derinlik	1	3	3	2	1	2	1	1



# *İçeriği bir sayı olan öge gösterimi*

Ayrık Küme Tanımı

1

Temel Ayrık Küme İşlemleri

2

3

Uygulama: Şifre Çözme

4

5

6

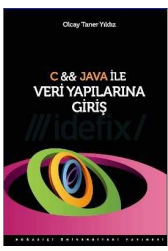
7

8

9

10

```
public class Oge{
    int icerik;
    int ebeveyn;
    int derinlik;
    public Oge(int icerik){
        this.icerik = icerik ;
        ebeveyn = icerik;
        derinlik = 1;
    }
}
```



# Her elemanı bir öğeden oluşan ayırık kümeler gösterimi

Ayrık Küme Tanımı

1

Temel Ayrık Küme İşlemleri

2

3

Uygulama: Şifre Çözme

4

5

6

7

8

9

10

11

```
public class Ayrikkume{
    Oge kumeler[];
    int kactane;
    public Ayrikkume(int N){
        int i;
        kumeler = new Oge[N];
        kactane = N;
        for (i = 0; i < N; i++)
            kumeler[i] = new Oge(i);
    }
}
```

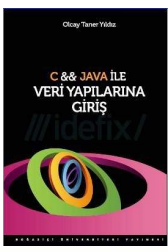


Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme

# Temel Ayrık Küme İşlemleri



# *Sıra numarası verilen bir ögenin ait olduğu kümeyi döndüren fonksiyon*

Ayrık Küme Tanımı

1

Temel Ayrık Küme İşlemleri

2

3

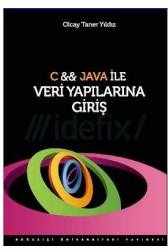
Uygulama: Şifre Çözme

4

5

```
int kumeBul(int sıra){  
    if (kumeler[sıra].ebeveyn != sıra)  
        return kumeBul(kumeler[sıra].ebeveyn);  
    return kumeler[sıra].ebeveyn;  
}
```



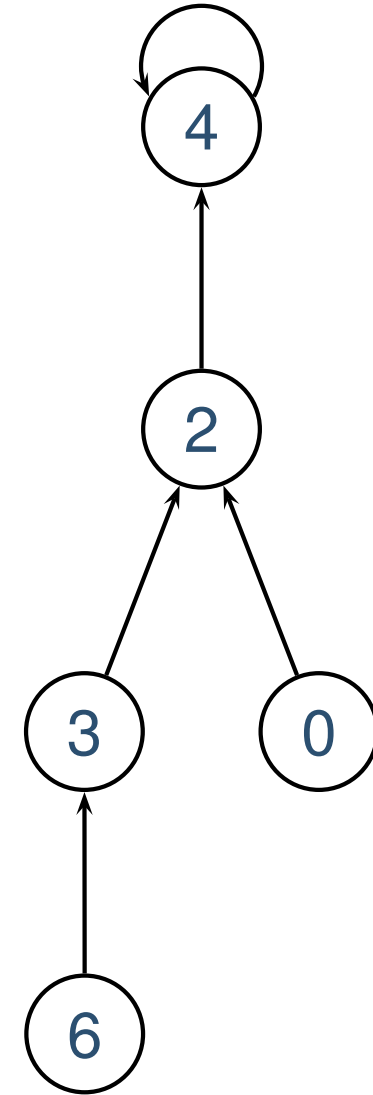
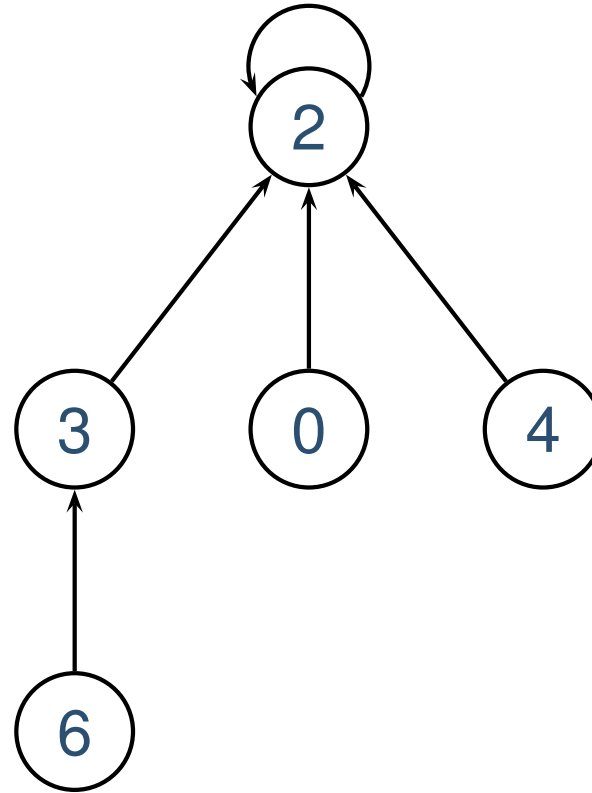


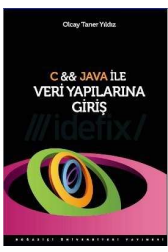
# Göstericisi 2 ve 4 olan kümelerin iki değişik şekilde birleştirilmesi

Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme



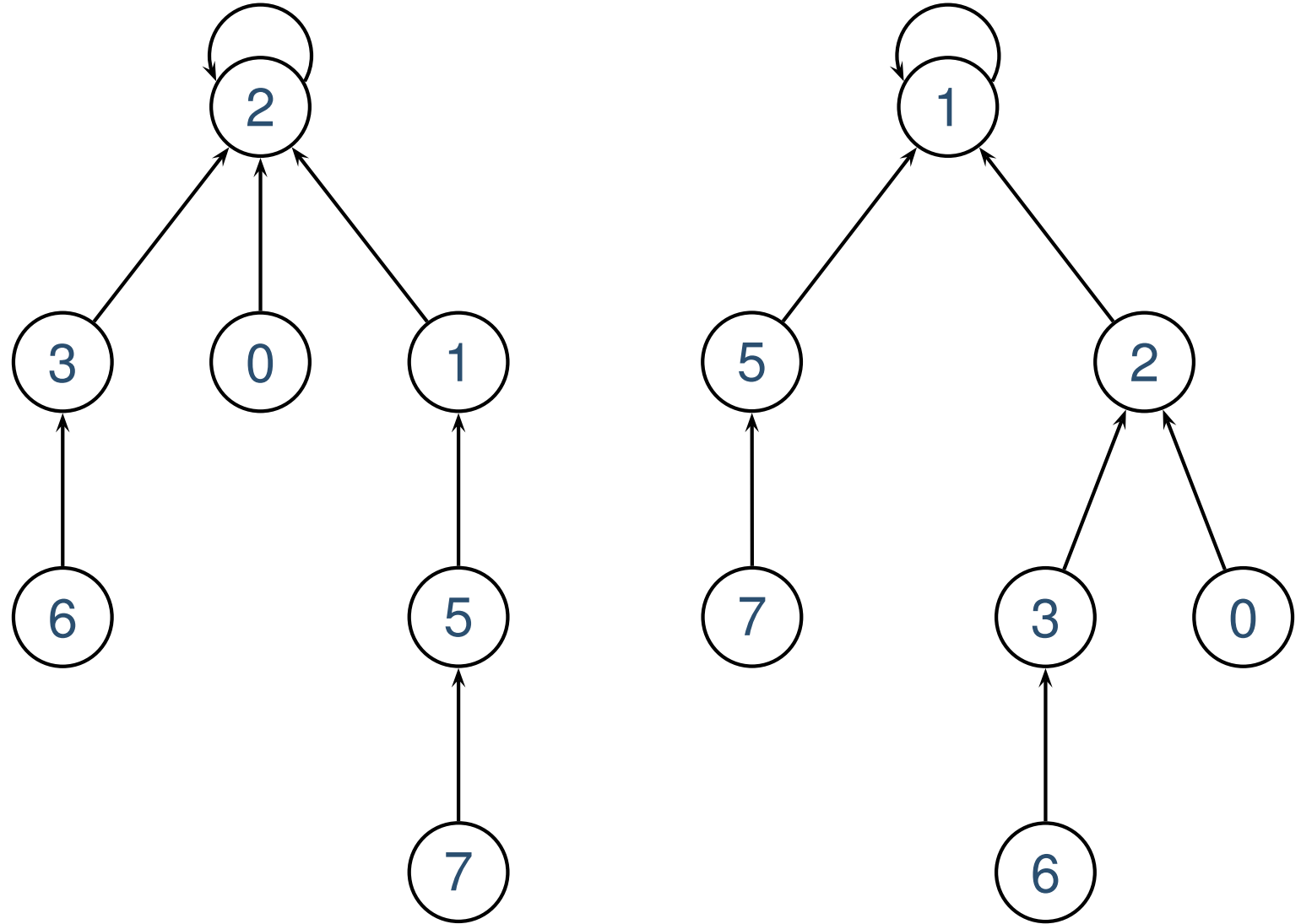


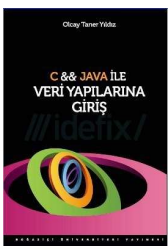
# Göstericisi 1 ve 2 olan kümelerin iki değişik şekilde birleştirilmesi

Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme





# Sıra numarası verilen iki ögenin kümelerini birleştiren fonksiyon

Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12

```
void kumeBirlestir(int sira1, int sira2){
    int x, y;
    x = kumeBul(sira1);
    y = kumeBul(sira2);
    if (kumeler[x].derinlik < kumeler[y].derinlik )
        kumeler[x].ebeveyn = y;
    else{
        kumeler[y].ebeveyn = x;
        if (kumeler[x].derinlik == kumeler[y].derinlik )
            kumeler[x].derinlik ++;
    }
}
```



# Ayrık Küme İşlemleri

Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme

- Küme Bulma:  $\mathcal{O}(\log N)$
- Küme Birleştirme:  $\mathcal{O}(\log N)$



Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme

## Uygulama: Şifre Çözme



# Örnek

Ayrık Küme Tanımı

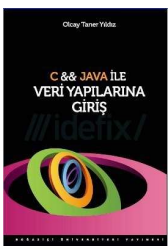
Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme

ABCEGFE

BCdFHGa

- birinci karakterden  $A = B$ ,
- ikinci karakterden  $B = C$ , dolayısıyla  $A = B = C$ ,
- üçüncü karakterden  $C = d$ , dolayısıyla  $A = B = C = d$ ,
- dördüncü karakterden  $E = F$ ,
- beşinci karakterden  $G = H$ ,
- altıncı karakterden  $F = G$ , dolayısıyla  $E = F = G = H$ ,
- yedinci karakterden  $E = a$ , dolayısıyla  $E = F = G = H = a$



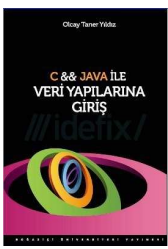
# Şifre çözümü probleminin uygulanması (1)

Ayrık Küme Tanımı

Temel Ayrık Küme İşlemleri

Uygulama: Şifre Çözme

```
1  boolean katarEsleme(String birinci, String ikinci ){
2      int kume1, kume2, i;
3      Ayrikkume a;
4      a = new Ayrikkume(26);
5      for (i = 0; i < birinci .length; i++){
6          kume1 = a.kumeBul(birinci.charAt(i) - 'A');
7          kume2 = a.kumeBul(ikinci.charAt(i) - 'A');
8          if ( birinci .charAt(i) >= 'a' && birinci .charAt(i) <= 'z')
9              if ( ikinci .charAt(i) >= 'a' && ikinci .charAt(i) <= 'z')
10                 if ( birinci .charAt(i) != ikinci .charAt(i))
11                     return false;
12             else{
13                 if (a.kumeler[kume2].icerik != - 1){
14                     if (a.kumeler[kume2].icerik != birinci .charAt(i) - 'a')
15                         return false;
16                 }else
17                     a.kumeler[kume2].icerik = birinci .charAt(i) - 'a';
18             }
```



# Şifre çözümü probleminin uygulanması (2)

Ayrık Küme Tanımı

19

Temel Ayrık Küme İşlemleri

20

Uygulama: Şifre Çözme

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

```
else
    if ( ikinci.charAt(i) >= 'a' && ikinci.charAt(i) <= 'z' ){
        if ( a.kumeler[kume1].icerik != -1){
            if ( a.kumeler[kume1].icerik != ikinci.charAt(i) - 'a' )
                return false;
        }else
            a.kumeler[kume1].icerik = ikinci.charAt(i) - 'a';
    }else{
        a.kumeBirlestir(kume1, kume2);
        if ( a.kumeler[kume1].icerik != a.kumeler[kume2].icerik)
            if ( a.kumeler[kume1].icerik == -1)
                a.kumeler[kume1].icerik = a.kumeler[kume2].icerik;
            else
                if ( a.kumeler[kume2].icerik == -1)
                    a.kumeler[kume2].icerik = a.kumeler[kume1].icerik;
            else
                return false;
        }
    }
return true;
}
```