

## YMT219 VERİ YAPILARI ÖDEV-1

1. “İkiliBul” yordamı aşağıda verilmiştir. İkiliBul yordamı A dizisi içerisinde 2 tane eşit sayı bulursa “true” bulamazsa “false” döndürmektedir.

```
public boolean ikiliBul(int[] A){
    for(int i=0; i<A.length; i++)
        for(int j=i+1; j<A.length; j++)
            if(A[i] == A[j])
                return true;

    return false;
}
```

- (a) Bu fonksiyonun toplam yürütme zamanını bulunuz. ( $A.length = n$ ) (Karmaşıklık istenmediğine dikkat edin.)
- (b) Bu fonksiyonun zaman karmaşıklığını bulunuz.  $O(?)$
- (c) 100 elemanlı bir dizi için çalışma zamanı 10 saniye ise 400 elemanlı bir dizi için çalışma zamanı ne kadar olur?
2. Verilen bağlantılı listeyi tersine çeviren yordamı yazınız. Aşağıdaki algoritmayı kullanınız. Yığın sınıfını oluşturun, daha sonra listedeki elemanları yığma koyun bittiğinde yığındaki elemanları listeye koyun. Yığın ve Bağlantılı Liste sınıflarının gerçekleştiriminin olduğu düşünülecektir(Yığın ve Bağlantılı Liste yordamlarını direk çağırabilirsiniz).
3. Verilen dizindeki tüm dosyaların(dizin veya dosya) kaç tane olduğunu bulan ve geri döndüren yordamı yazınız. Bulunan sayı altdizinleri de kapsayacak şekilde olmalıdır.

Bu soruda dosya ve dizin sayısını belirlemek için “File” sınıfını kullanabilirsiniz. Aşağıda yordam prototipi verilmiştir.

File sınıfı aşağıdaki kullanışlı yordamlara sahiptir:

```
File[] listFiles(); //Dizindeki tüm dosyaları listeler.
//Bu yordamın “File” türünden bir dizi
//döndürdüğüne dikkat edin.

// Eğer d dosya ise return 0
// Eğer d dizin ise return 1 + her bir dizindeki dosya sayısı
public static int DosyaSay(File d)
{
```

4. Aşağıdaki her bir kod parçası için kodun yürütme zamanını ve zaman karmaşıklığını (Büyük-O Notasyonuna göre) n cinsinden veriniz.

```
a) for(i=n; i>=1; i=i-1){
    for(j=1; j<=i; j=j+1){
        printf(" ");
    }
}
```

```
b) for(i=5; i<=n; i=2*i){
    print(" ");
}
for(i=5; i<=5*n; i=i+1){
    printf(" ");
}
```

5. İçerisinde N tane farklı sayı bulunduran(negatif veya pozitif) A **sıralı** dizisinde  $1 \leq i \leq N$  ve  $A[i] = i$  şartlarını sağlayan en küçük i indisini bulan ve geri döndüren yordamı yazınız. Programınız çalışma zamanını Büyük-O (Big-O) notasyonuna göre veriniz.

**Not:** Bu programı farklı algoritmalar ile yapabilirsiniz. Eğer algoritmanızın çalışma zamanı  $O(\log n)$  ise 2 kat puan kazanacaksınız.

6. 8,1,6,3,6,7 sayıları Kabarcık, Seçmeli ve Eklemeli sıralama ile küçükten büyüğe sıralanması isteniyor. Her bir sıralama algoritması için sıralama işlemini **adım adım** gösteriniz.
7. Boş bir listeye aşağıdaki işlemler yapılıyor. Her bir işlemten sonraki listenin durumunu gösteriniz.

```
Liste.add(5);
Liste.add(4);
Liste.add(1,7);
Liste.add(0,9);
Liste.remove(2);
Liste.remove(0);
Liste.add(2,2);
Liste.clear();
Liste.add(0,2);
```

8. Kuyruk gerçekteřtirimini 5 kapasiteli bir dizi üzerinde yaptığımızı varsayalım. Ařağıdaki her bir iřlem için dizinin o anki durumunu, ön ve arka deęiřkenlerinin aldıęı deęerleri gösteriniz.

```
Kuyruk.ekle(1);
Kuyruk.ekle(23);
Kuyruk.cikart();
Kuyruk.ekle(11);
Kuyruk.cikart();
Kuyruk.ekle(22);
Kuyruk.ekle(33);
Kuyruk.ekle(13);
Kuyruk.cikart();
Kuyruk.cikart();
```

9. Ařağıdaki programın çiktısını yazınız.

```
public static void main(String[] args){
    Yigin y = new Yigin();

    if (y.bosmu())
        System.out.println("Yıęın boş");

    y.koy(12);
    y.koy(22);

    System.out.println("Yıęının ilk elemanı: "+ y.al());
    y.koy(44);

    System.out.println("Yıęının ilk elemanı: "+ y.ust());
    y.koy(22);

    System.out.println("Yıęının ilk elemanı: "+ y.al());
    System.out.println("Yıęının ilk elemanı: "+ y.al());
    System.out.println("Yıęının ilk elemanı: "+ y.ust());
    System.out.println("Yıęının ilk elemanı: "+ y.al());

    if (y.bosmu())
        System.out.println("Yıęın boş");
}
```

10. Bağlantılı liste gerçekleştirimi için kullanılan Dugum ve BaglantiliListe sınıflarının gerçekleştirimi verilmiştir.

```
// Dugum sınıfı aşağıda verilmiştir.
public class Dugum {
    public int veri;
    public Dugum sonraki;
}

// BaglantiliListe sınıfının bir kısmı
public class BaglantiliListe {

    private Dugum bas;
    private Dugum son;
    private int elemanSayisi;

    public baglantiliListe() {
        bas = son = null;
        elemanSayisi = 0;
    }

    public void add(int poz, int e){...} //Kullanılabilir
    public void remove(int poz){...} //Kullanılabilir
    public int indexOf(int e){...} //Kullanılabilir
    public boolean isEmpty(){...} //Kullanılabilir
    public int first(){...} //Kullanılabilir
    public int last(){...} //Kullanılabilir
    public int get(int poz){...} //Kullanılabilir
    public int size(){...} //Kullanılabilir

}
}
```

Aşağıda listelenen yordamların kodunu yazınız.

```
/**
    Bu yordam verilen elemanın listede olup olmadığını kontrol
    eder. Eğer varsa true yoksa false döndürür.
*/
public boolean contains(int element)

/**
    Bu yordam verilen indisteki elemanı verilen değer ile
    değiştirir. Değiştirilen elemanı geri döndürür.
    indis: Değiştirilmek istenen elemanın indisi
    eleman: Değiştirilecek elemanın yeni değeri
    return: Eski değer
*/
public int set(int indis, int element)
```

Ödevlerinizi **Word dokümanı** şeklinde hazırlayıp 11 Aralık Pazar günü saat 23:59'a kadar [varol.asaf@gmail.com](mailto:varol.asaf@gmail.com) adresine "**YMT219\_odev1**" konusu ile gönderiniz.

**Not:** Ödevlerinizi gönderirken gmail hesaplarınızı kullanınız.