

4. (12P) Kökü parametre olarak verilen ikili ağacı ortada değer(inorder) biçiminde dolaşıp değerleri ekrana yazdıran **inorder** metodunu yazınız.

```
public static<T> void inorder(BTNode<T> root) {  
  
  
  
  
  
  
}
```

5. (12P) Aşağıda dizi temsili olarak verilen ikili yığın yapısında kökten başlayarak en sola doğru inecek ve geçtiği düğümün değerini ekrana yazdıracak olan **koktenSola** metodunu yazınız.

```
class BinaryHeap<T extends Comparable<T>> {
    public T[] dizi; //Yığın tutacak dizi
    public int es; //Eleman sayısı
    public void koktenSola() {

    }
}
```

6. (12P) Özetleme için kullanılan HashEntry dizisinde bir değeri arayan; varsa konumunu, yoksa -1 döndüren `ara` metodunu yazınız. Herhangi bir değerin özeti(hash) birinci satırda görüldüğü gibi `hashCode` metodu ile alınabilir.

[illegible]


```

/**
 * Sınav süresince ihtiyacınız olacak sınıf ve tanımlamalar burada
 * bulunmaktadır.
 */
class Node<T> { //Tek bağlı liste düğüm
    public T value; //Değer
    public Node<T> next; //Sonraki düğüm
}
class BTreeNode<T> { // İkili ağaç düğümü
    public BTreeNode<T> left; //Sol çocuk
    public BTreeNode<T> right; //Sağ çocuk
    public T value; //Değer
}
class HashEntry<T> {
    public T value; //Değer
    public boolean isActive; //aktif değil ise değer silinmiştir
}
class Edge<T> { //Kenar
    Vertex<T> from; //Kenarın çıktığı vertex
    Vertex<T> to; //Kenarın girdiği vertex
    int weight; //Ağırlık
}
class Vertex<T> { //Düğüm
    public T value; //Vertex değeri
    public List<Edge<T>> edges; //Çıkan kenar listesi
}
class Graph<T> { // Çizge
    public List<Vertex<T>> vertices; //Vertex listesi
    protected Map<T, Vertex<T>> verticesMap; //(Değer, Düğüm) haritası
}

```