

Pamukkale Üniversitesi Bilgisayar Mühendisliği Veri Yapıları Laboratuvarı Dönem Sonu Sınavı Soruları (26.12.2019)



Öğrenci Numarası: _____ Adı Soyadı: _____

Soru	1	2	3	4	5	6	7	8	9	10	Toplam
Puan	12	12	12	12	12	12	12	12	12	12	120
Not											

Son sayfada bütün sorularda kullanılan sınıfların kurucu metotları olmadan elemanları verilmiştir. Lütfen bu kısmı inceleyin.

1. (12P) Parametre olarak verilen ve iki yığıtın(java.util.Stack) birbirine eşitliğini kontrol eden esit metodunu yazınız. Yığıtların bir kopyasını oluşturmanıza gerek yoktur, içinden eleman silebilirsiniz. java.util.Stack sınıfının int size() metodu yığıttaki eleman sayısını, boolean isEmpty() metodu da boş olup olmadığını verir.

```
public static<T> boolean esit(Stack<T> s1, Stack<T> s2){
```

2. (12P) Tamsayıların bulunduğu tek bağlı listenin düğümünü(Node<Integer>) parametre olarak alan ve bağlı listedeki sayıların toplamını hesaplayıp döndüren topla metodunu yazınız.

```
public static int toplam(Node<Integer> head) {
}
```

3. (12P) İkili bir ağacın kökünü parametre olarak alan ve kökün ardıl(successor) değerini bulan successor metodunu yazın. Kökün sağ çocuğunun her zaman var olduğunu varsayınız.

```
public static<T extends Comparable<T>> T successor(BTNode<T> root) {
}
```

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;//Yrğrnr tutacak dizi
   public int es;//Eleman sayrsr
   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.

```
public static<T> int ara(HashEntry<T>[] dizi, T aranacak) {
   int h=aranacak.hashCode();//Orjinal konum
```

7. (12P) Parametre olarak gönderilen çizgedeki çıkan kenar sayısı(out degree) en fazla olan düğümün(vertex) değerini geri döndüren enCokCikan metodunu yazınız.

8. (12P) Parametre olarak verilen çizgenin tüm kenarlarının hangi düğümler arasında olduğunu ekrana yazdıran kenarListesi metodunu yazınız. Bir kenar A -> B şeklinde yazılabilir.

```
public static <T> void kenarListesi(Graph<T> g) {
}
```

9. (12P) İki farklı çizgeyi düğüm(vertex) sayılarına göre karşılaştıran GraphKarsilastir sınıfının compare metodunu yazınız.

```
class GraphKarsilastir<T> implements Comparator<Graph<T>> {
    @Override
    public int compare(Graph<T> g1, Graph<T> g2) {
    }
}
```

10. (12P) Kök düğümü parametre olarak verilen bir ikili ağaçtaki düğüm sayısını hesaplayan dugumSayisi metodunu yazınız.

```
public static<T> int dugumSayisi(BTNode<T> node){
```

```
/**
 * 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 BTNode<T> { // İkili ağaç düğümü
   public BTNode<T> left;//Sol cocuk
   public BTNode<T> right;//Sağ çocuk
   public T value;//Değer
class HashEntry<T> {
   public T value;//Değer
   \verb|public boolean isActive|; //aktif de \"gil ise de \~ger 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
```