**BMÜ-221 Veri Yapıları Vize Soruları**

**1.(8p)**Aşağıdaki her bir kod bölümü için Big-O notasyonuna göre n’e bağlı olarak en kötü çalışma zamanlarını veriniz.

|  |  |
| --- | --- |
| Kod parçası | Big-O |
| for(int i=0; i<n; i+=2){  for(int j=0; j<n\*n/3; j++)  x+=j;  } |  |
| for(int i=0; i<n; i\*=2){  fof(int j=0; k<n\*n/3;j++)  x+=j;  } |  |
| for(int i=0; i<n; i++){  if(n<1000){  for(int j=0; j<n; j++)  x+=1;  }  Else x+=1;  } |  |
| İnt foo(int n) {  İf(n<10) return n;  else if(n<1000) return n\*foo(n-2);  else return n\*foo(n/2) |  |

**2. a)** Aşağıdaki AVL ağacını veren tüm olası ekleme dizilerini yazınız**(5p)**

|  |  |
| --- | --- |
| 2  / \   1. 4   /  3 |  |

**b)** Boş bir AVL Ağacına, (a)'daki ekleme sıralarından hangisi herhangi bir döndürme olmaksızın dengeli AVL ağacı oluşturur?**(2p)**

|  |
| --- |
|  |

**c)** Aşağıdaki AVL ağacını göz önüne alarak her bir durum için ağacı gösterip dengeli hale getiriniz **(6p).**

|  |  |
| --- | --- |
| 6  / \  2 10  \  14 | Rotasyon gerektirmeyen bir ekleme |
| Tek rotasyon gerektiren bir ekleme | |
| Çift rotasyon gerektiren bir ekleme | |

**d)** Aşağıda verilen ikili arama ağacında 7 elemanı silindikten sonraki ağaç yapısını gösteriniz **(2p)?**

7

/ \

3 9

/ \ \

1 5 10

**e)** d seçeneğinde verilen ağacın preorder dolaşımını aşağıya yazınız **(2p)**?

|  |
| --- |
|  |

**4.** Tam sayıları tutan bir bağlı listede en büyük elemanı listenin sonuna atan void listeSonuMax(BListe L) {} metodunu yazınız? Örneğin; 1🡪5🡪6🡪2🡪4 olarak verilen bağlı liste 1🡪5🡪2🡪4🡪6 olmalıdır. Bağlı listedeki elemanların pozitif tamsayılar olup ve birbirinden farklı olduğu varsayılacaktır.**(10p)**

**5.** Bağlı liste ile tanımlanmış bir tamsayı yığıtını parametre olarak alan ve yığıtın elemanlarının yukarıdan aşağıya azalmayan bir sırada olmasını sağlayan bir yöntem yazınız. Bunu yapmak için, yığıtın tepesindeki herhangi bir elemandan daha küçük olan elemanlar yığıttan silinir.

alt[**4**, 20, 15, 15, **8, 5, 7**, 12, **3**, 10, 5, 0]ust

Yukarıdaki yığıtta **3, 7, 5, 8 ve 4** elemandan küçük olduğu için çıkarılır. Ek bir yığıt kullanabilirsiniz. **(15p)**

**6. a)** Bir Yığıt sınıfınız olduğunu ve tüm metotlarının yazıldığını kabul ederek aşağıdaki işlemler sonucunda yığıtın son halini çiziniz **(5 p).**

static void Main(string[] args)

{

Yigit y = new Yigit(5);

y.Push(10);

y.Peek();

y.Push(-5);

y.Push(-2);

y.Peek();

y.Pop();

y.Push(30);

y.Peek();

y.Push(99);

y.Peek();

y.Pop();

y.Pop();

y.Push(50);

y.Push(10);

y.Push(14);

}

**b)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

Yukarıda verilen şekilde bir dairesel dizi kuyruk yapısı bulunmaktadır. Kuyrukta sırası ile aşağıdaki işlemler gerçekleştirilirse;

Kuyruğun son hali nasıl olur çiziniz **(3p).**

Front ve Rear ifadelerinin değerlerini yazınız **(2p).**

Insert(B)

Insert(C)

Delete()

Insert(A)

Delete()

Delete()

Insert(M)

Insert(Y)

Insert(O)

Insert(K)

Insert(L)

Delete()

Insert(N)

**7.** Aşağıdaki gibi bir kod yazdığınızı ve BinarySearchTree sınıfında tanımlanan insert, delete ve find metotlarının gövdelerinin yazılmış olduğunu varsayınız.

public class BinarySearchTreeNode {

public int key;

public SomeType value;

public BinarySearchTreeNode left;

public BinarySearchTreeNode right;

}

public class BinarySearchTree {

private BinarySearchTreeNode root;

public void insert(int key, SomeType value) { ... }

public void delete(int key) { ... }

public SomeType find(int key) { ... }

}

Buna göre;

1. Ağacın yüksekliğini hesaplayan bir public int treeHeight() metodu yazınız **(10p).**
2. Ağaçtaki maksimum elemanın bulunduğu düğümü silen bir metot yazınız **(10p).**

**8.** Çift yönlü bir bağlı liste sınıfı oluşturarak LL isimli bir bağlı liste nesnesi oluşturunuz. LL listesinde bir head(bas) düğümü ve M ile gösterilen ve son düğüm olmayan bir düğüm olduğunu düşünerek aşağıdaki işlemler için gerekli kodları yazınız.

1. Head düğümünü siliniz **(4p).**
2. M düğümünün next’ine yeni bir P düğümü ekleyiniz **(3p).**
3. Head düğümü ve M düğümlerini değiştiriniz(swap) **(4 p).**