

Ders Ödev Bilgileri	
Konu	Açıklama
Final Uygulama saati	<p>Final Ödevi için Uygulama Sınav Saatleri:</p> <p>1. Öğretim: 13:30-16:30 arası</p> <p>2. Öğretim: 09:00-12:00 arası</p> <p>Öğrencilerin ödevleri flash belleklerinde olmalıdır. Daha önceden laboratuvar da denenmiş olması gerekir. Önceden kontrol etmeniz de fayda vardır. Çalışmayan ödevler kabul edilmeyecektir. Ödevler ile ilgili rapor hazırlamanız gerekmektedir. Rapor bir kapak dosyasına sahip olmalı. Ödev tarihi öğrenci no, ad soyad vb. bilgileri içermelidir. Duyurulur.</p>
Ödev Final	Altan alanlar için ders ödevi duyurular kısmından atılmıştır.
Ödev-1 Bağlı Liste	<p>1. Bir bağlı liste yapısı ile tek parametreye sahip bir polinom (Örneğin $P(x) = x^3 + 3x^2 + 5x + 10$) tutulmak isteniyor. Bağlı listedeki her bir düğüm polinomun ilgili teriminin katsayısı ve derecesini tutmaktadır. Buna göre;</p> <p>-Bağlı Liste ve düğüm veri yapısını veriniz?</p> <p>-Bağlı listeye ekleme işlemini dereceye göre sıralı yapan kodu veriniz?</p> <p>2. Her bir düğümünde bir adet karakter tutan bir bağlı liste tasarlanmak isteniyor. Bu bağlı liste üzerinde aşağıdaki işlemleri gerçekleştiren JAVA uygulamasını kodlayınız.</p> <p>Ekleme,</p> <p>Baştan ekleme</p> <p>İstenilen sırada ekleme</p> <p>Sondan ekleme yapıları</p> <p>Silme</p> <p>Baştan silme</p> <p>İstenilen sırada silme</p> <p>Sondan silme</p> <p>NOT: Ödevler bilgisayar ortamında hazırlanarak kodlanacaktır. Ayrıca yazılan kodlar ile ilgili el ile yazılmış bir rapor hazırlanarak laboratuvar da teslim edilecektir.</p>

<p>Ödev-2 Yığit</p>	<p>Verilen Tarih: 09.10.2018 BMÜ-221 Veri Yapıları Ödev-2</p> <p>1.Verilen bir Stringteki parantezlerin eşli olup olmadığını yığit yapısı kullanarak belirleyen metodu yazınız? Örneğin parantezler ({ () [{ }] } ()) şeklinde ise parantezler eşli olup doğru sırada parantezler açılmıştır. (}]) (ifadesi ise eşli olmayıp doğru sırada açılmamıştır. Verilen ifadede sadece (, {, [,), },] parantez işaretleri bulunmaktadır.</p> <p>2.Yığit kullanarak onluk tabandaki bir sayı ikilik tabana dönüştürülmek isteniyor. Örneğin 233 sayısının 2'lik tabana dönüştürülmesi aşağıda verilmiştir. Buna göre kendisine parametre olarak bir sayı alan ve sayının ikilik tabandaki karşılığını bir yığita yazdıran metodu ve yığiti parametre olarak alıp yığit içeriğini bir string'e aktarıp geri döndüren metodu yazınız?</p> <p>233/2 = 116 kalan =1 = push(1) 116/2 = 58 kalan =0 = push(0) 58/2 = 29 kalan = 0 = push(0) 29/2 = 14 kalan=1 = push(1) 14/2 = 7 kalan=0 = push(0) 7/2 = 3 kalan=1 push(1) 3/2 = 1 kalan=1 push(1) 1/2 =0 kalan=1 push(1)</p>
<p>Ödev-3 Yığıtlar</p>	<p style="text-align: center;">BMÜ-221 Veri Yapıları ÖDEV-3</p> <p>1.String olarak alınan bir infix ifade için yığit kullanılarak aşağıdaki işlemler yapılacaktır.</p> <ul style="list-style-type: none"> -Verilen infix ifadeyi postfix'e dönüştüren metodu yazınız. -Verilen infix ifadeyi prefix'e dönüştüren metodu yazınız. -Hesaplanan postfix ifadenin sonucunu hesaplayan metodu yazınız. <p>2.N uzunluktaki bir dizi üzerinde iki tane yığit yapısı kurulmak istenmektedir. Y adlı yığit dizisinin Y[0] elemanı birinci yığit için, Y[N-1] elemanı da ikinci Yığit için G/Ç uçlarıdır. İki yığita konulacak toplam eleman sayısı N olup ilgili gözleri iki yığit tarafından da boş olduğu sürece kullanılabilir. Yığitin dolu olması için tüm gözlerin dolu olması gerekir. Bir yığitin boşalttığı bellek gözünü diğer yığit kullanabilir. Bu yapıdaki Yığit için gerekli veri yapısını tanımlayarak Yığit işlemlerini gerçekleştiren kodları yazınız.</p>

Ödev-4 Kuyruk	<p>Round robin iş zamanlaması algoritmasında, işlerin büyüklükleri değişken olabilmektedir. Quantum(q) adı verilen belirli bir boyuttan büyük işlerin q büyüklüğü kadarlık kısmı tamamlanıp kalanı kuyruğun sonuna tekrar eklenmektedir. Büyüklüğü q dan küçük olan işler ise tamamlanıp kuyruktan tamamen silinir.</p> <p>İşlemci 'ye gelen işler sırasıyla 12 5 ve 18 olsun. Gelme sırasına göre işler kuyruğa eklenir. q=10 seçilirse her adımda kuyruk aşağıdaki gibi olur.</p> <p>Adım 1: 12 5 8 Adım 2: 5 18 2 (Kuyruğun başındaki işin 10 birimi yapıldı. Kalan 2 birim kuyruğun sonuna eklendi) Adım 3: 18 2 Adım 4: 2 8 Adım 5: 8 Adım 6: Boş</p> <p>Buna göre kuyruk sınıfını, ekleme ve silme metotlarını yazınız? Silme işleminde tamamı bitirilemeyen işlerin kuyruğa yukarıda verilen şekilde eklenmesini de gerçekleştiriniz?</p>
Ödev-5 Hashing	<p>1. Hash kullanarak verilen iki diziden ikincisinin birinci dizinin alt kümesi olup olmadığı bulunmak isteniyor. Bu amaçla aşağıdaki algoritma kullanılabilir.</p> <p>a)dizi1[]'in bütün elemanlarını tutmak için bir hash tablosu oluştur.</p> <p>b)dizi2[]'nin her elemanını hash tablosunda ara. Eğer eleman yok ise false döndür.</p> <p>c)Bütün elemanlar bulundu ise true döndür.</p> <p>Zincir kuralına (Bağlı Liste) göre çakışmaların giderildiği bir Hash sınıfı oluşturarak yukarıdaki algoritmaya göre iki dizi yi parametre alan (boolean altküme(int[] dizi1, int[] dizi2){ }) ve ikinci dizinin birinci dizinin bir alt elemanı olup olmadığını bulan metodu yazınız?</p> <p>2. Elimizde bir iş yerindeki personellere ait numara(int) ve isim(String) bilgileri bulunmaktadır. Doğrusal sına (Lineer probing)'ya göre çakışmaların giderildiği bir hash tablosuna gelen elemanları (numara'yı baz alarak) yerleştiriniz. Arama, ekleme ve silme işlemleri için gerekli metotları oluşturunuz.</p>
Ödev-6 Ağaçlar	<p>Bir ikili arama ağacında öğrencilerin numarası(int), adı(String) ve notu(int) tutulacaktır. Ağaçta ekleme numaraya göre yapılmaktadır. Bu ikili arama ağacında Ekleme, silme, arama, maksimum ve minimum elemanı bulma, arama ağacında dolaşım (Preorder, inorder, postorder) ve ağacı gösterme metotlarını içeren sınıfı tasarlayınız ve test metodunu yazınız? İkili arama ağacında lever-order (seviye seviye) dolaşmayı yapınız. (Kuyruk yapısı kullanılacaktır)?</p>

Ödev-6 ile ilgili	<p>Merhaba Arkadaşalar; Ödev-6'da level order gezinme yaparken her bir ağaç düğümü aynı zamanda kuyruğa ekleneceğinden aşağıdaki gibi bir yapı kullanılabilir. sol ve sag baglar ağaç için gerekliyken ileri bağı kuyruk için gereklidir.</p> <pre> class Dugum { int icerik; Dugum sol,sag,ileri; Dugum(int icerik){ this.icerik=icerik; sol=null; sag=null; ileri=null; } } </pre>
Ödev-7 AVL Ağacı	<p>Bir AVL ağacında bir mağazadaki müşterilerin id'si(int), adı(String) ve borcu(double) tutulacaktır. Gerekli veri yapısını tanımlayarak aşağıdaki metotları içeren programı yazınız?</p> <ul style="list-style-type: none"> - Müşteri id'sine göre AVL ağacına ekleme işlemi için gerekli metodu yazınız? - AVL ağacını level order gezinme yapınız (Kuyruk yapısı kullanılacaktır)? - Bu AVL ağacının ikili arama ağacı olup olmadığını geri döndüren metodu yazınız <p>NOT: İkili arama ağacında her düğüm sol çocuğundan büyük ve sağ çocuğundan küçüktür?</p> <ul style="list-style-type: none"> - Kendisine parametre olarak borç bilgisi alan ve AVL ağacında bu borçtan daha yüksek borçları olan müşterilerin id, isim ve borçlarını gösteren metodu yazınız.
Ödev-8 Heap	<p>A ve B karakterler tutan iki farklı kümedir. Bu iki küme kullanılarak A birleşim B ve A kesişim B işlemleri yapılmak isteniyor. Buradaki en büyük zorluk küme elemanlarının sıralanmamış olmasından kaynaklanmaktadır. Her bir elemanın diğer kümenin bütün elemanları ile karşılaştırılması ise problemin zaman karmaşıklığını arttırmaktadır. Bu yüzden çözüm yolu olarak heap ağacının kullanımı uygun görülmüştür. Buna göre;</p> <ul style="list-style-type: none"> - A ve B kümeleri için binary heap ağaçlarını oluşturarak ekleme, silme ve arama metotlarını yazınız? - A ve B kümesinin kesişimini döndüren bir metot yazınız? - A ve B kümesinin birleşimini döndüren bir metot yazınız?