**PROJE 3**

**ARAMA AĞAÇLARI, YIĞINLAR ve HASH TABLOSU: BİSİKLET KİRALAMA SİSTEMİ**

**Her bir öğrenci kendi başına hazırlayacak, programlar paketlenerek sisteme yüklenecek, rapor halinde arka arkaya getirilip istenen bilgiler de yazılarak EGE Derse yüklenecektir. Raporda, her bir soru için ilgili kaynak kod ve ekran görüntüsü yer almalıdır. İstediğiniz soruda C#, istediklerinizde Java tercih edilebilir. Raporda Bölüm Numara ve Başlıkları belirgin olarak yazılmalıdır: 1. İkili Arama Ağacı (Binary Search Tree), 2. Hash Tablosu (Hash Table), 3. Yığın Ağacı (Heap), 4. Sıralama (Sorting).**

Duraklar Nesnesi (**Durak Adı**, **Boş Park**, **Tandem Bisiklet, Normal Bisiklet**)

String[] duraklar = { “İnciraltı, 28, 2, 10”, “Sahilevleri, 8, 1, 11“, “Doğal Yaşam Parkı, 17, 1, 6”, “Bostanlı İskele, 7, 0, 5“ };

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

1. **a)** duraklar dizisine 5 tane daha istasyonun bilgilerini ekleyiniz (**bisim**.com.tr gibi bisiklet kiralama sistemlerini araştırarak tahmini değerler yazabilirsiniz), 9’a tamamlayınız.duraklar dizisindeki string’leri sahalarına ayrıştırarak **Durak Nesnelerini** oluşturup, ***Durak Adı’na*** *göre DurakAğacı adlı* **ikili arama ağacı**na yerleştiren kodu yazınız. Her bir Durak nesnesini ağaca eklerken, düğüme List tipinde bir veri yapısı içinde 1 ile 10 adet arasında random sayıda rastgele Müşteri (Müşteri ID, kiralama saati) ekleyiniz (Sistemde toplam 20 müşterinin kayıtlı olduğunu varsayınız ID:1 – ID:20). Boş park ve bisiklet sayılarını güncelleyiniz. Hazır ağaç kodlarından yararlanabilirsiniz. Sayısal elemanlar için uygun veri tiplerini belirleyiniz. Şekil 1’de ağacın 3 düğümlü durumu gösterilmektedir.

**Durak Nesnesi**

**DurakAğacı**

**Durak Adı: İnciraltı**, BP=28, TB=2, NB=10

**Şekil 1: Örnek bir   
durak ağacı**

**List**

ID:5, 13:15 ID:9, 14:17 …

**Müşteriler**

**Sahil Evleri, 8, 1,  
...**

**Doğal Yaşam Parkı**, 17 ...

**b) Ağacın derinliğini bulduran metodu yazınız. Ağaçtaki tüm bilgileri (Listeye göre kaç müşterinin kiralama yaptığı bilgisi ve Liste içindeki bilgiler dahil) ekrana listeleyen metodu yazınız. (5)**

**c) Klavyeden verilen bir Müşteri ID’si için ağacın tüm düğümlerini dolaşarak içlerindeki listelerdeki bilgilere bakarak hangi istasyonlardan saat kaçta bisiklet kiraladığını ekrana listeleyen metodu yazınız. (5)**

**d) Kiralama işlemi: Adı verilen bir istasyondan, Müşteri Numarası (ID) verilen kişinin normal bir bisiklet kiralama işlemini yapıp, Listeye ilgili bilgiyi (ID + random saat) ekleyen metodu yazınız. BP sayısı da 1 artacak. (5)**

**2) a)** 1. Soruda belirtilen **Durak** Bilgilerini **Durak Adı’na göre** bir **Hash Table’a** yerleştiren kodu yazınız. (10) [duraklar dizisini kullanabilirsiniz. Bu soruda müşteri eklemenize gerek yok]

**b)** **Boş Park Yeri sayısı 5’ten fazla olan** tüm istasyonlara **5’er tane normal bisiklet ekleyerekHash Tablosunda** güncelleyen kodu yazınız. (5)

**3) a)** Ders kitabı Bölüm 12’yi okuyunuz. C# / Java ile bir Heap Veri Yapısı (sınıfı) tasarlayınız ve metotları ile beraber kurşunkalemle yazınız ve yapılan işlemleri yazarak anlatınız. Altyapıda elemanlarını tutmak için dizi veya List / Vector kullanabilirsiniz. Kodlayıp çalıştırınız. (5)

**b)** Sadece **Normal Bisiklet** sayılarına göre düğümleri bir Max. Yığın’a yani **Max.** **Heap’e** (Java’daki PriorityQueue Heap düzenindedir) yerleştiren kodu yazınız. (5)

**c)** O anda en fazla **Normal Bisiklet** olan üç İstasyonu **Heap’ten çekerek** ilgili durak / istasyon bilgilerini listeleyen kodu yazınız. (5)

**4) a)** Ders kitabındaki 3. Bölümden (**Simple Sorting**) bir sıralama algoritması seçerek okuyunuz, öğreniniz, mantığını anlayınız. Kitaba veya hazır koda bakmadan bir A4 kağıdına kurşunkalemle yazınız. Kodlayarak, Debug içerisinde değişkenlerin değişimini izleyiniz. (4)

**b)** Ders kitabındaki 7. Bölümden (**Advanced Sorting**) bir sıralama algoritması seçerek okuyup öğreniniz, mantığını anlayınız. C# / Java’da kodlayınız. [Bu seçenekte elle yazım yok] (4)

**c)** Her iki yöntemin **zaman karmaşıklıklarını** hesaplayınız ve **1 paragraf açıklama** yazarak **yöntemleri karşılaştırınız**. **Basit sıralama algoritmaları ne zaman daha avantajlı olur? (4)**

**d)** [**https://www.youtube.com/watch?v=kPRA0W1kECg**](https://www.youtube.com/watch?v=kPRA0W1kECg)adresindeki sıralama algoritmaları görselleştirme videosunu izleyiniz. Görselleştirmenin, algoritmaları anlamanıza etkisini belirtiniz. Veri Yapılarını öğrenme açısından **Videolar** ile **Etkileşimli Görselleştirme Araçlarını (Duyurularda Bağlantıları var) karşılaştırınız. (8)**

**5) Özdeğerlendirme tablosunu doldurarak raporun sonuna ekleyiniz (10 puan):**

**Özdeğerlendirme Tablosu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Proje 3 Maddeleri (her maddede kodlar ve ekran görüntüleri de yer alacaktır)** | **Not** | **Tahmini Not** | **Açıklama** |
| **1 a) Ağaç (Durak Ağacı Oluşturma)** | **10** |  |  |
| **1 b) Derinlik Bulma, Ağacı Listeleme** | **10** |  |  |
| **1 c) Verilen Müşterinin Tüm Kiralamaları** | **10** |  |  |
| **1 d) Kiralama İşlemi** | **10** |  |  |
| **2) Hash Tablosu** | **15** |  |  |
| **3) Yığın Ağacı (Heap)** | **15** |  |  |
| **4) Sıralama Algoritmaları** | **20** |  |  |
| **5) Özdeğerlendirme Tablosu** | **10** |  |  |