**PROJE 3**

**ARAMA AĞAÇLARI, YIĞINLAR ve HASH TABLOSU: YEMEK DAĞITIM ŞİRKETİ**

1. Bornova’daki mahalleler için bir **yemek sipariş ağacı** oluşturulacaktır (**Şekil 1**). Ağacın her bir düğümü bir mahallenin adını ve o mahalleden yapılan siparişlerin listesini (**SiparişlerListesi**) tutacaktır. Bu listedeki her bir sipariş de, 2. projedekinden farklı olarak birden fazla yiyecek/içecek türü içerebileceği için liste (**SiparişBilgileri**) şeklinde tutulacaktır. Bu listenin her bir elemanı **YemekSınıfı** nesnesi şeklinde olacaktır. YemekSınıfı, sipariş verilen yemeğin / yiyecek veya içeceğin **adını, adedini** ve **birim fiyatını** içermelidir. [Ağaç, mahalle adına göre ikili arama ağacı şeklinde oluşturulmalıdır.]

SiparişlerListesi

SiparişBilgileri

Mahalle Nesnesi

**Şekil 1: Yemek Sipariş Ağacı**(Yemek Dağıtım Şirketi Siparişleri)

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

|  |  |  |
| --- | --- | --- |
| Türlü  3  15 *TL* | Pilav  2  10 | Ayran  3  3 |

1. **Yemek Sipariş Ağacının oluşturulması :** Bornova’daki 5 mahallenin her biri için (Evka 3, Özkanlar, Atatürk, Erzene, Kazımdirik), 5-10 arasında rastgele sayıda sipariş içeren sipariş listesi oluşturun. Her bir siparişte de 3-5 arasında rastgele yiyecek/içecek türü bilgisi olsun. Yiyecek adetleri ise 1 ile 8 arasında rastgele değişebilir. Sipariş üretmek için, öncesinde menü bilgilerini içeren bir liste oluşturup rastgele yiyecek / içecek ad+fiyatı çekmek için kullanabilirsiniz. **(10)**
2. Ağacın derinliğini bulduran ve ağaçtaki tüm bilgileri (her mahalledeki sipariş bilgileri bazında) ekrana listeleyen metodu yazınız. **(10)**
3. Adı verilen bir mahalledeki 150 TL üstündeki siparişlerin bilgilerini listeleyen metodu yazınız. **(10)**
4. Adı verilen bir yiyecek/içeceğin tüm ağaçta kaç adet (kaç kişi tarafından değil) sipariş verildiğini döndüren metodu yazınız. Ağaçta o yemeğin geçtiği tüm listelerde yemeğin birim fiyatına %10 indirim uygulayarak güncelleme yapınız. **(10)**
5. **a)** Bornova’da seçtiğiniz 10 mahallenin Bilgilerini (Mahalle adı, nüfusu) **Mahalle Adı’na göre** bir **Hash Table’a** yerleştiren kodu yazınız. **(10)** <https://www.nufusune.com/bornova-ilce-nufusu-izmir>

**b)** **Baş harfi verilen mahallelerin toplam nüfusuna 1 ekleyerekHash Tablosunda** güncelleyen kodu yazınız. **(5)**

**3) a)** Ders kitabı Bölüm 12’yi okuyunuz. Heap Veri Yapısı (sınıfı) ile metotlarını öğreniniz. **C# veya Java ile** **Kodlayıp çalıştırınız.** Altyapıda elemanları tutmak için dizi veya List / Vector kullanabilirsiniz. **(5)**  **[Kurşunkalemle algoritma yazma işlemi bu sene kaldırılmıştır]**

**b)** Nüfuslarına göre Bornova’daki 10 mahalleyi **Max.** **Heap’e** (Java’daki PriorityQueue Heap düzenindedir) yerleştiren kodu yazınız. **(5)** [Dileyen a maddesinde yazdığı kodu güncelleyerek, dileyen Java’daki PriorityQueue sınıfını kullanabilir.]

**c)** Nüfusu en fazla olan 3 mahalleyi sıra ile **Heap’ten çekerek** ilgili mahalle adı ve nüfus bilgilerini listeleyen kodu yazınız. (5)

**4)** Ders kitabındaki 3. Bölümden (**Simple Sorting**) **veya** 7. Bölümden (**Advanced Sorting**) bir sıralama algoritması seçerek okuyunuz, öğreniniz, mantığını anlayınız.

**a)** **Kodlayarak**, Debug içerisinde değişkenlerin değişimini izleyiniz. (5)

**b)** **Zaman karmaşıklığını hesaplayınız**. Diğer bir sıralama algoritması ile karşılaştırınız, o algoritmaya göre hangi durumlarda avantajlı, hangi durumlarda dezavantajlı olduğunu kısaca açıklayınız. (5)

**c)** [**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. (10)**

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

**Özdeğerlendirme Tablosu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Proje 3 Maddeleri** | **Not** | **Tahmini Not** | **Açıklama** |
| **1 a) Ağaç (Yemek Sipariş Ağacı Oluşturma)** | **10** |  |  |
| **1 b) Derinlik Bulma, Ağacı Listeleme** | **10** |  |  |
| **1 c) Arama ve Listeleme** | **10** |  |  |
| **1 d) Ağaçta bir yemek türüne indirim yapma** | **10** |  |  |
| **2) Hash Tablosu** | **15** |  |  |
| **3) Yığın Ağacı (Heap)** | **15** |  |  |
| **4) Sıralama Algoritmaları** | **20** |  |  |
| **5) Özdeğerlendirme Tablosu** | **10** |  |  |

Not: Projeyi tek kişi yapabileceğiniz gibi 2 kişi ortak yapıp teslim edebilirsiniz.

Proje 3 için işletim ve kod anlatım videosu çekilmeyecektir.

Bazı soruların kodlamasında C#, bazılarında Java’yı karışık olarak kullanabilirsiniz.