**PROJE 3**

**ARAMA AĞAÇLARI, YIĞINLAR ve HASH TABLOSU: MİLLİ PARK BİLGİ SİSTEMİ**

1. **a) Milli Parklar (Proje 2’deki Milli Park nesnelerini içeren) ağacının oluşturulması:** Veriler, https://www.tarimorman.gov.tr/DKMP/Belgeler/Korunan%20Alanlar%20Listesi/MP-WEB-Son.pdfbağlantısındaki sırada alınarak Milli Park adına göre bir ikili arama ağacına eklenecektir (Şekil 1).

Eklemeden önce https://tr.wikipedia.org/wiki/T%C3%BCrkiye%27deki\_mill%C3%AE\_parklar\_listesibağlantısındaki milli park isimlerine tıklayıp Vikipedi’den her biri için birer veya daha fazla paragraflık bilgi alınmalıdır (yazdığım bu cümledeki işlemi kod ile yapmanız zorunlu değildir). Sonra kod yazarak bu paragraflardaki her bir cümleyi ayrıştırarak (generic) List’in bir elemanı yaparak generic List’i ilgili Milli Park nesnenize ekleyin (cümleleri ayrıştırırken, paragraf içindeki sayılarda binlik basamak veya ondalık ayırıcı anlamındaki . ları, bunları sonraki karakterin boşluk olmaması bilgisini kullanarak cümle sonu noktası olmadığını anlayabilirsiniz). **(10)**

…

Cümleler   
Generic List

**Şekil 1: Milli Parklar Ağacı**

Milli Park Nesnesi

Yozgat Çamlığı için

1. Ağacın derinliğini bulduran ve ağaçtaki tüm bilgileri (her Milli Park’taki tüm bilgileri) ekrana listeleyen metodu yazınız.Bu metoda ağacın düğüm sayısını da bulduran ve dengeli ağaç olsaydı derinliğinin kaç olması gerektiğini hesaplayan kodu da ekleyiniz. **(10)**
2. Adının ilk 3 harfi verilen bir Milli Parkın, hangi il veya illerin içinde yer aldığı bilgisini listeleyen metodu yazınız. **(10)**
3. Tüm ağaçtaki Vikipedi’den alınan bilgi cümlelerindeki kelimeleri bu ağacı dolaşırken (List’lerden) alarak ayrı bir ikili arama ağacına yerleştirerek saydıran metodu yazınız. İpucu: Yeni ağacın her bir düğümü ilgili kelimeyi (string) ve sayacı yani kelimenin ağaçtaki toplam sayısını (int) içerecek. **(10)**
4. **a)** 1. Maddede oluşturduğunuz Milli Park nesnelerini **Milli Park Adı’na göre** bir **Hash Table’a** yerleştiren kodu yazınız. **(10)**

**b)** **Adı verilen Milli Parkın ilan tarihini yine klavyeden verilen ilan tarihine göreHash 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 geçen sene kaldırılmıştır]**

**b)** Yüzölçümlerine göre Milli Parkları **Max.** **Heap’e** (Java’daki PriorityQueue Heap düzenindedir) yerleştiren kodu yazınız. **(5)** [Dileyen a maddesinde yazdığı kodu güncelleyebilir, dileyen Java’daki PriorityQueue sınıfını kullanabilir.]

**c)** Yüzölçümü en büyük olan 3 Milli Parkı sıra ile **Heap’ten çekerek** tüm 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 (Milli Parklar İkili Arama Ağacı)** | **10** |  |  |
| **1 b) Derinlik Bulma, Ağacı Listeleme, Düğüm Sayısı Buldurma, Dengeli Ağaç Derinliği Hesaplama** | **10** |  |  |
| **1 c) Arama ve Listeleme** | **10** |  |  |
| **1 d) Kelime Ağacı Oluşturarak Kelimeleri Sayma** | **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.