

2. Ödev [Tüm Şubeler İçin]

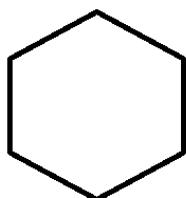
Teslim Tarihi

14 Aralık 2025 (Pazar günü saat 23:50'ye kadar.)

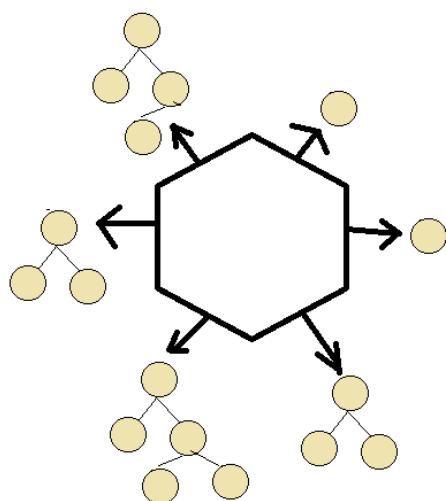
Ödev İçeriği

Ön Uyarı: Kodlama ödevlerinde üretken yapay zeka araçlarının kullanımı, intihal tespit sistemlerinde yüksek benzerlik oranlarıyla karşılaşılması riskini artırmaktadır. Bunun temel nedeni, yapay zeka modellerinin benzer girdilere (komutlara) karşılık standart algoritma ve kod yapıları üretme eğilimidir. Akademik dürüstlük ve mesleki gelişim açısından değerlendirildiğinde, yalnızca yapay zeka tarafından oluşturulan çalışmaların öğrenciye herhangi bir yetkinlik kazandırmadığı açıklıdır. Bilişim sektöründe yetkin bir profesyonel olabilmek; özgün kod yazabilme, mevcut kodu analiz edebilme ve algoritmik düşünme becerilerinin bizzat geliştirilmesini gerektirir. Yapay zeka araçları ile kod yazarak benzer ödevler gönderilmesinin tespiti durumunda tarafların ödev notu sıfır olarak değerlendirilecektir. Bu noktada sorumluluk öğrenciye aittir.

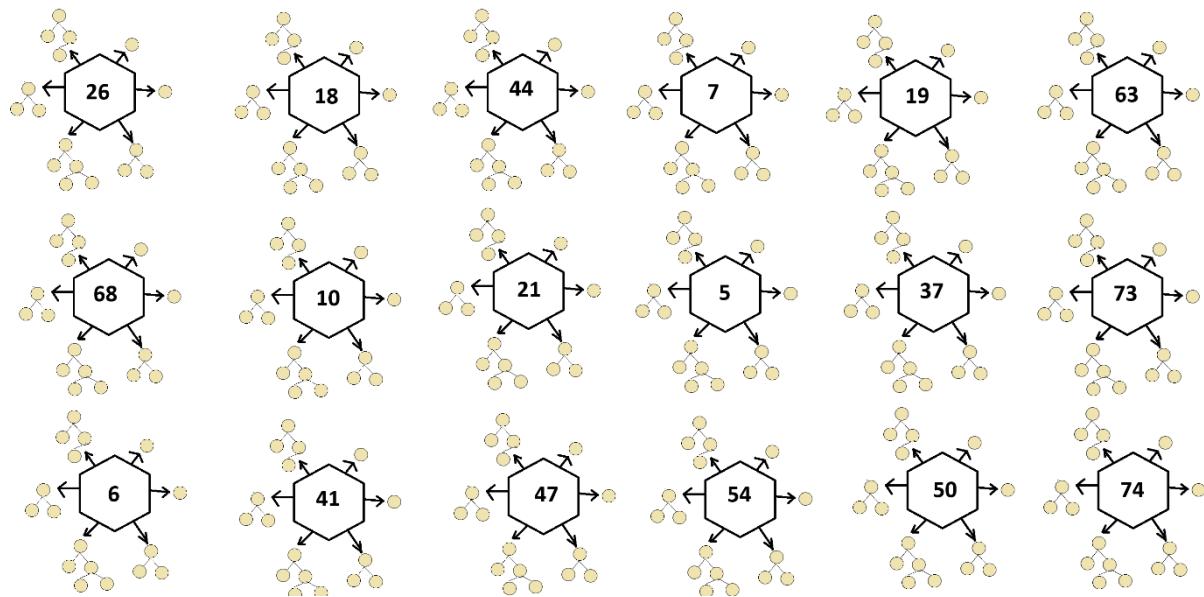
MinGW C++ ortamında derlenen bir program yazılmacaktır. Bu ödevde aşağıda verilen altigen şekil, bir öncelikli kuyruk veri yapısını temsil etmektedir. Kuyruğa en fazla 6 veri eklenebilmektedir. Her kenar bir veri ve bu veri ikili arama ağacını tutmaktadır.



İkili arama ağaçlarında eşit değer gelirse sola eklenecektir.



Tüm yapı aşağıdakine benzer olacaktır. Buradaki sayılar kuyrukta çıkmak üzere olan ağacın kökündeki değerin öncelikli çıkmak üzere olan ağac kökündeki değere bölünmesi ile elde edilir. Öncelik ağac yüksekliği ile belirlenir. Ağac yüksekliği fazla olan önceliklidir. Örneğin çıkmak üzere olan bir ağacın kökü 75, öncelikli çıkmak üzere olan ağacın kökü 25 ise orada 3 değeri yazmalıdır. Sayı ve bölmeler tam sayı olarak yapılmalıdır. $75/25=3$



Kuyruk ve ağaçlar dosya okunarak oluşturulacaktır. Data.txt aşağıdaki gibi bir formatta olacaktır. Her satırda sayılar soldan sağa doğru ikili arama ağacına yerleştirilecektir. Satır bitince oluşan ikili arama ağacı kuyruğa eklenecektir. Altıgen yapı bitti mi sıradaki altıgen yapıya geçilecektir.

```

73 5 48
91 16 33 7 82
40
12 67 58 94
3 29 77 10 86 54
100 8
61 25 44 98 14
72 6 31 89 53 20 1
17 95
26 62 39 11 84
57 9 32 45 81 13 70 22
50 27 66 19 96 35

```

Dosya okunma aşamasında toplam kaç altıgen olabilir hesaplanıp ekranda boş olarak gösterilecek. Dosyadan satırlar okundukça ekran güncellenecek ve kuyruklardaki sayısal değişim gözlemlenebilecektir. Tüm dosya okunduktan sonra oluşan sayısal durum ekranda gösterilecektir. Ekranda her satırda en fazla 6 adet kuyruk gösterecektir. Sayılar anlaşılmasına için aşağıdaki sayılar rastgele ve temsili verilmiştir.

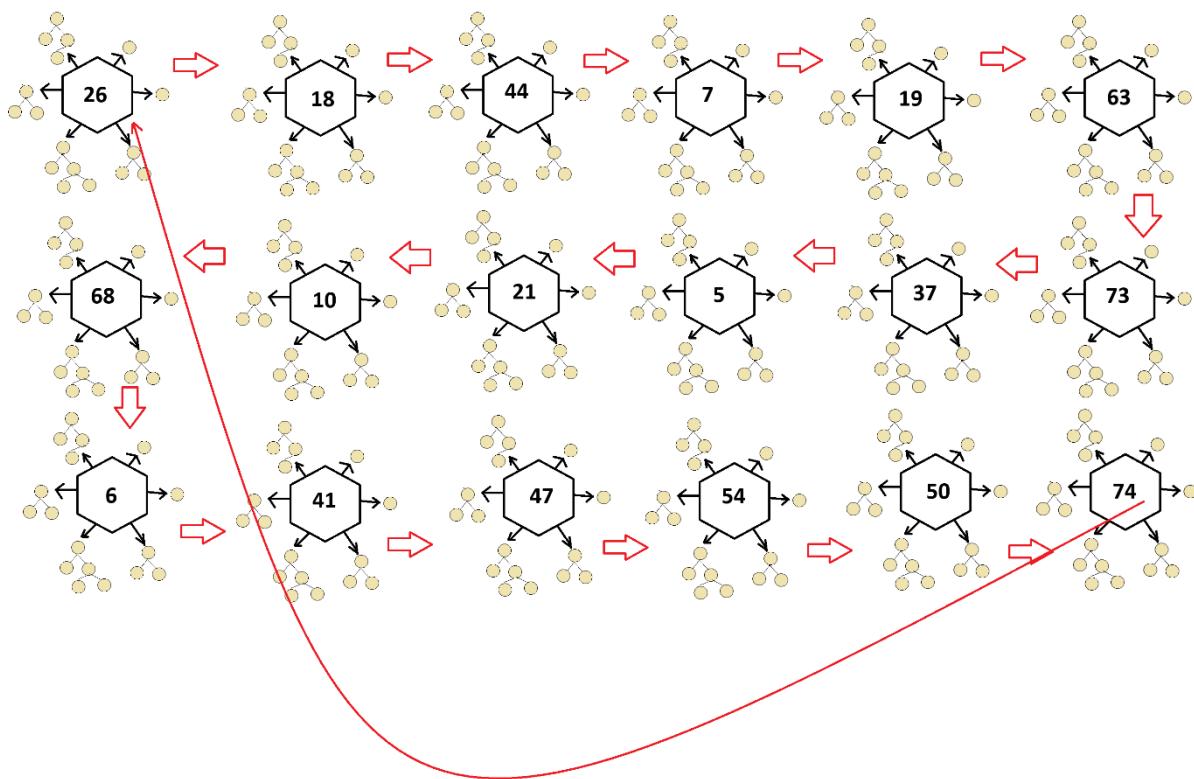
26	18	44	7	19	63
68	10	21	5	37	73
6	41	47	54	50	74

Kullanıcıdan kaç tur çalışacağı istendikten sonra bir tuşa bastığı gibi turlar başlayacak ve **ekranda hangi turda olduğu görülecek** ve sayıların da her turda son durumları ekranda görünecek yani ekran sürekli güncellenecektir. Ekranda daha önce yazılanlar temizlenmelidir. Turlar bitince sayıların son

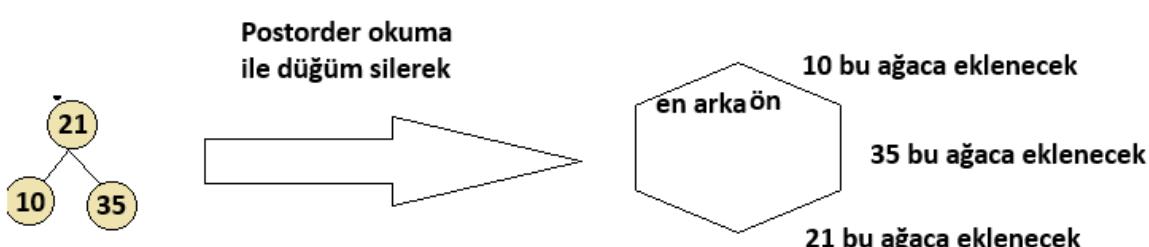
durumu ekranın üzerinde görünecektir. Ekran bu süre zarfında durdurulamayacak. Sadece son durum ekranın üzerinde görünecektir.

Tek numaralı turlarda kuyruklardan normal veri (ağaç) çıkarılacak bir sağindaki altigen yapıdaki ağaçlara eklenecektir. **Çift numaralı turlarda** kuyruklardan öncelikli veri (ağaç) çıkarılacak bir sağindaki altigen yapıdaki ağaçlara eklenecektir.

Ekleme yönü aşağıda verilmiştir.



Diğer altigene ekleme şu şekilde yapılacaktır. Bir ikili arama ağacı komple kuyruktan silinip altigene eklendirken postorder bir şekilde düğümler silip kuyrupta önden arakaya doğru her ağaca bir düğüm eklenecektir. Düğümler bitmez ama kuyruğun sonuna gelinirse tekrar aynı kuyruğun önünden başlanıp devam edilecektir.



Nesne yönelimli yaklaşım ile kod yazılmalıdır. Programın yavaş çalışması durumunda puan kırılacaktır. En az 50.000 satırlı bir Data.txt ile mutlaka deneme yapınız ve süreyi gözlemleyiniz.

Turlar bittiğinde son durum ekranın üzerinde gösterilmelidir.

Hiçbir şekilde Şablon (Generic) veri yapısı kullanılamaz. Her sınıfın başlık ve kaynak dosyası ayrı olmalı ve başlık dosyasında metot gövdesi bulunmamalıdır. Hazır veri yapısı kullanılamaz.

Önemli Not: Raporunuz detaylı olmalı ve kendi cümleleriniz olmalıdır. Kopya ödevler sıfır olarak değerlendirilecektir. SABİS şifreniz sizin sorumluluğunuza altındadır eğer arkadaşınız sizden habersiz ödevinizi alırsa bundan sizde sorumlu tutulur ve sıfır alırsınız.

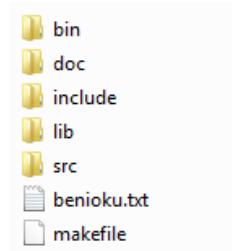
Teslim Formatı

Klasör Hiyerarşisi

- src Klasörü (Kaynak dosyalarınızın bulunduğu klasör)
- **lib Klasörü** (.o dosyalarının bulunduğu klasör)
- include Klasörü (Başlık dosyalarınızın bulunduğu klasör)
- doc Klasörü (Raporunuzun bulunduğu klasör.)
- bin Klasörü (Çalıştırılabilir dosyalarınızın (.exe) bulunduğu klasör.)
- benioku.txt (Program için ayrıca belirtmek istedikleriniz. Önerileriniz. Boş kalabilir.)
- **makefile**

makefile olmayan ödev derlenmemiş olarak kabul edilmektedir. (-50 puan)

Rapor Yoksa veya yetersiz ise (-15 Puan)



Rapor pdf formatında olmalıdır. Raporu ayrıca çıktı olarak getirmenize gerek yoktur. Raporunuzda kısaca sizden istenilen, öğrencikleriniz, ödevde yaptıklarınız, eksik bıraktığınız yerler, zorlandığınız kısımlar anlatılabilir. **Ödev raporunda yazı boyutu 12 puntodan büyük olamaz ve en az 1 sayfa en çok 2 sayfa olabilir.**

Klasörlerinizi bir ana klasör içine koyarak .rar veya zip'leyip **SABİS** üzerinden gönderiniz. Ana klasörünün adı sadece sizin öğrenci numaranız (örn: B111210090) olmalıdır. Yukarıda belirtilen teslim tarihinden sonra gönderilen ödev **kesinlikle kabul edilmeyecektir**.

Önemli: SABİS üzerinden bir ödev için dosya gönderme hakkınız 3 seferdir. Hoca ekranında en son yüklediğiniz dosya görünür.

Yazmış olduğunuz bütün kaynak kodların en başında aşağıdaki bilgiler bulunmalıdır. Bilgileri kendinize göre güncelleyiniz.

```
/**  
 * @file           Dosya adı  
 * @description    Programınızın açıklaması ne yaptığına dair.  
 * @course         Dersi aldığınız eğitim türü ve grup  
 * @assignment     Kaçinci ödev olduğu  
 * @date          Kodu oluşturduğunuz Tarih  
 * @author         Gruptakilerin yazar adları ve mail adresleri  
 */
```

Önemli: Yazacağınız kodlar MinGW derleyicisinde kesinlikle derlenmelidir. Derlenmeyen kod itibar görmez ve çok düşük puan üzerinden değerlendirilir. Sadece C++ kodları kullanılmalıdır.

Ödev Bireyseldir

KOPYA ÖDEV SIFIR OLARAK DEĞERLENDİRİLMEKTEDİR*

* Şubeler farklı dahi olsa kopya ödev gönderenler sıfır olarak değerlendirilecektir.