Ara Sınav 1
Çözümleri

- Dağıtılan sınav kitapçığını, size söylenene kadar açmayın. Talimatları dikkatlice okuyun.
- Kitapçığın her sayfasına isminizi yazın.
- Sınav, 4 problem ve alt bölümlerinden oluşmaktadır. 80 puanlık sınav için 80 dakikanız var.
- Kitapçık, bu sayfa ile beraber 13 sayfadan oluşmakta. Karalama yapabilmeniz için 2 boş sayfa bulunmakta. Bu kağıtları, sınavı teslim etmeden önce kitapçıktan lütfen ayırın.
- Bu sınav, kapalı kitap tarzındadır. Hesap makinesi ve programlanabilir cihaz kullanmak yasaktır. Sınava el yazımı bir A4 formül kağıdı ile girebilirsiniz.
- Cevaplarınızı size ayrılan yerde cevaplayınız. Daha fazla yere ihtiyacınız olursa, sorunun bulunduğu kağıdın arkasını kullanın. Bir sorunun cevabını, başka bir sorunun kağıdına yazmayın, çünkü kağıtlar notlandırma esnasında birbirinden ayrılabilir.
- Daha once öğrendiklerimizi ispatlamak veya elde etmek için zaman ve yer harcamayın. Bildiğimiz cevaplardan bahsetmeniz yeterli olacaktır.
- Bir soruda gereğinden fazla zaman harcamayın. Bütün soruları okuyun, daha sonra en fazla bilgi ve fikir sahibi olduğunuz sorudan cevaplamaya başlayın.
- Gidiş yoluna da puan verileceğinden, cevabınızı açıkça ifade edin. Cevabın doğruluğunun yanında cevabı veriş şekliniz de puanlamaya etki edecektir.
- Bol şanslar!

Problem	Alt bölüm	Puan	Not	Notlandıran
1	4	12		
2	1	7		
3	11	44		
4	3	17		
Toplam		80		

İsim : _____

Problem 1. Asimptotik Koşma Süreleri [12 puan] (4 bölüm)

Aşağıda listelenen her algoritma için,

- en kötü koşma süresini ifade eden bir yineleme yazın.
- en kötü koşma süresini ifade eden bir Θ-simgelemi yazın.

Cevabı açıklamanıza gerek bulunmamakta.

(a) İkili Arama

Çözüm: $T(n) = T(n/2) + \Theta(1) = \Theta(\lg n)$

(b) Araya yerleştirme sıralaması

Çözüm: $T(n) = T(n-1) + \Theta(n) = \Theta(n^2)$

(c) Strassen Algoritması

Çözüm: $T(n) = 7T(n/2) + \Theta(n^2) = \Theta(n^{\lg 7})$

(d) Birleştirme Sıralaması

Çözüm: $T(n) = 2T(n/2) + \Theta(n) = \Theta(n \lg n)$

Problem 2. Yerine Koyma Metodu [7 puan]

$$\begin{aligned} T(n) &= T(n/2) + T(n/4) + n\ , \\ T(m) &= 1 \text{ için } m \leq 5. \end{aligned}$$

Yerine koyma metodunu ve O-simgelemini kullanarak, yinelemenin çözümüne sıkı bir üst sınır bulun.

Çözüm: T (n) = O(n) olduğunu tahmin ediyoruz. Bu bizi tümevarıma götürür.
Bütün m < n'ler için, T(m) ≤ cm.
c ≥ 1 için, temel durumda n ≤ 5 için, T(n) =1≤ cn'dir.

Tümevarımla,

$$T(n) = T(n/2) + T(n/4) + n \leq cn/2 + cn/4 + n = (3c/4 + 1)n.$$

sonucunu elde ederiz. Eğer c=4'ü seçersek, T(n) ≤ (3+1)n= 4n = cn'dir. n'nin üzerinden tümevarımla, c ≥ 4 ve n ≥ 1 olan bütün n'ler için; T(n) ≤ cn'dir.

Problem 3. Doğru veya Yanlış ve Doğrula [44 puan] (11 bölüm)

Aşağıdaki ifadelerin doğru veya yanlış olduğunu belirtmek için **D** veya **Y'**yi daire içine alın. Eğer ifade doğru ise, neden doğru olduğunu kısaca açıklayın. Eğer ifade yanlış ise, neden yanlış olduğunu kısaca açıklayın. Ne kadar ayrıntılı içerik sağlarsanız, o kadar yüksek puan alacaksınız, ama cevabınız kısa olsun. Açıklamanız, D veya Y seçiminizden daha fazla not getirecek.

D Y $T(n) = 3T(n/3) + O(\lg n)$ yinelemesinin çözümü $T(n) = \Theta(n \lg n)$.

Çözüm: **Yanlış.** Ana Teoremin 3. durumu uygulanır. $f(n) = O(\lg n)$ için
$f(n) = O(n^{\log_3 3}) = O(n)$
yani $T(n) = O(n)$.

D Y F_k, k 'nıncı Fibonacci sayısı olsun. n^2 'ninci Fibonacci sayısı F_{n^2}, $O(\lg n)$ sürede hesaplanabilir.

Çözüm: **Doğru.** n^2 'inci Fibonacci sayısı kare alma ve matris çarpımı metodu ile $O(\lg n^2) = O(\lg n)$ zamanda bulunabilir.

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

D Y Bir dizilimin, n tane sayıyı içerdiğini düşünün. Bu sayıların her biri −1, 0 veya 1'dir. Bu dizilim, en kötü durumda O(n) sürede sıralanabilir.

Çözüm: **Doğru.** Sayma sıralamasını kullanabiliriz. Önce, girdi dizilimindeki her elamana 1 ekleriz. Böylece sayma sıralamasının ön şartı yerine getirilmiş olur. Sayma sıralamasını çalıştırdıktan sonra, çıktı dizilimindeki her elemandan bir çıkartırız.

Bölüntülere bağlı çözüm şöyle olur. A[1 . . n] girdi dizilimi olsun. Değişmezin tanımı;

- A[1 . . **i**], sadece −1'leri içerir,
- A[**i** + 1 . . **j**], sadece 0'ları içerir,
- A[h . . n], sadece +1'leri içerir.

Başlangıç olarak, **i** = 0, **j** = 0, ve h = n + 1. Eğer, h = **j** + 1 ise işimiz bitmiştir demektir. Dizilim sıralanmış demektir. Döngüde A[**j** + 1]'i inceleriz. Eğer A[**j** + 1] = −1 ise; A[**j** + 1]'i, A[**i** + 1] ile değiştiririz ve i ve j'yi 1 arttırırız. Eğer A[**j** + 1] = 0 ise; j'yi 1 arttırırız. Son olarak eğer A[**j** + 1] = +1 ise; A[**j** + 1]'i, A[h − 1] ile değiştiririz ve h'yi 1 azaltırız.

D Y Bir rakibin oluşturduğu, n uzunluğundaki bir girdi diziliminin rastgele hızlı sıralaması, algoritmayı, ω(n lg n) süresinde çalışmaya zorlar.

Çözüm: **Yanlış.** Derste gördüğümüz üzere, herhangi bir girdi için, hızlı sıralamanın beklenen koşma süresi O(n lg n)'dir; burada beklenti rastgele seçimi önemsiz kılar ve hızlı sıralama, girdinin seçiminden bağımsızdır.

D Y Bu dizilim,

20 15 18 7 9 5 12 3 6 2

bir (max-heap) yığını oluşturur.

Çözüm: Doğru.

- A[1] = 20, A[2] = 15 ≤ 20 ve A[3] = 18 ≤ 20 çocuklarına/ardıllarına sahip.
- A[2] = 15, A[4] = 7 ≤ 15 ve A[5] = 9 ≤ 15 çocuklarına sahip.
- A[3] = 18, A[6] = 5 ≤ 18 ve A[7] = 12 ≤ 18 çocuklarına sahip.
- A[4] = 7, A[8] = 3 ≤ 7 ve A[9] = 6 ≤ 7 çocuklarına sahip.
- A[5] = 9, A[10] = 2 çocuğuna sahip.
- A[6], . . . , A[10]'un çocuğu yok.

D Y Yığın sıralaması, taban sıralaması işleminde yardımcı sıralama metodu olarak kullanılabilir, çünkü yerinde çalışır.

Çözüm: Yanlış. Taban sıralamasında, yardımcı metod stabil olmak zorundadır, yani, aynı değere sahip sayılar, çıktı diziliminde girdi dizilimindeki ile aynı sırada olmalıdır. Yığın sıralaması stabil değildir. Yerinde çalışır, yani, girdi dizilimindeki elemanlardan sadece sabit sayıda bazıları dizilim dışında saklanır.

D Y 5 sayıyı sıralamak için, en kötü durumda 6 karşılaştırma ile karşılaştırma sıralaması yapılabilir.

Çözüm: Yanlış. Karar ağacının yapraklarının sayısı 5 sayı için 5!'dir, ve ağacın yüksekliği en az lg(5!).

5! = 120 olduğuna göre, $2^6 = 64$'tür, ve $2^7 = 128$'dir, yani

$6 < \lg(5!) < 7$. Yani en az 7 karşılaştırmaya ihtiyacımız olur.

D Y Kıyım tablosundaki çarpışmaların, n ögenin zincirlemesi $\alpha = 1/\lg n$ kadar yük oranına sahip olduğunu varsayın. Basit düz kıyımda, bir ögenin aranması için beklenen süre $O(1/\lg n)$'dir.

Çözüm: Yanlış. Bir ögenin aranması için beklenen süre $O(1 + \alpha) = O(1 + 1/\lg n) = O(1)$'dir. Bir ögenin aranması için, en az $O(1)$ sabit koşma süresine ihtiyaç vardır.
$O(1/\lg n)$ gibi sabit olmayan bir süreden bahsetmek mümkün değildir.

D Y Varsayalım X, $E[X] = 1/2$ gibi bir göstergesel rastgele bir değişken olsun. Bu durumda

$$E\left[\sqrt{X}\right] = 1/\sqrt{2}.$$

Çözüm: Yanlış.

X biröstergesel rastgele değişken olduğundan $X = 0$ veya $X = 1$'dir.

İkisi için de muhtemel değer;

$$\sqrt{X} = X$$
$$E\left[\sqrt{X}\right] = E[X] = 1/2.$$

D Y m yuvası olan bir kıyım tablosunun k anahtarı ile tek bir eleman sakladığını, geri kalan yuvaların boş olduğunu düşünün. Bu tabloda r kere, k'ye eşit olmayan başka anahtarları aradığımızı düşünün. Basit düz kıyım olduğunu varsayarsak, r sayıda aramanın, tabloda saklanan bir elemanın saklandığı yuvayı sondalama olasılığı r/m'dir.

Çözüm: **Yanlış.** r aramadan birinin kıyım tablosundaki eleman ile çarpışma olasığı p, 1 eksi r aramanın hiç çarpışmaması ihtimaline eşittir. Bu da,

$p = 1 - (1 - 1/m)^r$'dir .

D Y *S, n* tane tamsayıdan oluşan bir küme olsun. En kötü durumda, O(1) süresinde bir x tamsayısının S'nin içinde olup olmadığını belirleyecek bir veri yapısı tasarlanabilir.

Çözüm: **Doğru**. Mükemmel Kıyım.

Problem 4. Yakın Sayılar [17 puan] (3 bölüm)

S'nin, $n \geq 2$ olan ayrışık sayılardan oluşan bir küme olduğunu varsayın. Basitlik açısından, $k \geq 0$ için, $n = 2^k + 1$ olduğunu düşünün. x, y Є S olan bir çift ayrışık sayı eğer,

$$|x - y| \leq \frac{1}{n-1}\left(\max_{z \in S} z - \min_{z \in S} z\right) ,$$

ise, S'nin içinde yakındır. Yani, eğer, x ile y arasındaki uzaklık, en fazla sıralanmış düzende birbirini takip eden sayılar arasındaki ortalama mesafe kadar ise bu sayılar yakın sayılardır.

(a) $n \geq 2$ olan ayrışık sayılardan oluşan her S kümesinin, neden bir yakın sayı çifti içerdiğini kısaca açıklayın.

Çözüm: Genellemeyi kaybetmeden, $S = \{z_1, z_2, \ldots, z_n\}$ ve $z_i \leq z_{i+1}$ olduğunu kabul edelim. z_i ve z_{i+1} ardışık sayıları arasındaki ortalama mesafe

$$\frac{1}{n-1}\sum_{i=1}^{n-1}(z_{i+1} - z_i) = \frac{1}{n-1}(z_n - z_1).$$

Biribirini takip eden ve x ile y arasındaki ortalama uzaklığı, ortalamadan az veya ona eşit olan en az bir sayı çifti mevcuttur. Bu sonuca *yakın çiftlerin* tanımından ulaşılır.

(b) S'yi, p $\in$ S olan bir sabit elemanla S: $S_1 = \{x \in S \mid x \leq p\}$ ve $S_2 = \{x \in S \mid x \geq p\}$ olacak şekilde iki alt kümeye bölüntülediğimizi varsayalım.

Bu ikisinden birini kanıtlayın.

1. x, y $\in S_1$ olan yakın her çift, aynı zamanda S'de de yakın sayılardır, veya
2. x, y $\in S_2$ olan yakın her çift, aynı zamanda S'de de yakın sayılardır.

k $\in \{1, 2\}$ için, x, y $\in S_k$ olan her yakın sayı çiftinin S'de de yakın olduğunun, O(n) süresinde nasıl belirlenebileceğini gösterin.

Çözüm: Genellemeyi kaybetmeden, S_i'deki elemanların sıralı olduğunu varsayalım. k = 1, 2 için, a_k, S_k'daki iki ardışık sayı arasındaki ortalama uzaklık olsun. n_k da S_k'daki eleman sayısı olsun. Bölüm (a)'daki sonucu kullanırsak

$$a_1 = \frac{1}{n_1 - 1}\left(\max_{z \in S_1} z - \min_{z \in S_1} z\right) = \frac{1}{n_1 - 1}\left(p - \min_{z \in S} z\right),$$

ve

$$a_2 = \frac{1}{n_2 - 1}\left(\max_{z \in S_2} z - \min_{z \in S_2} z\right) = \frac{1}{n_2 - 1}\left(\max_{z \in S} z - p\right).$$

elde ederiz. Sıralı olan S'deki iki ardışık sayı arasındaki uzaklık a ise;

$$\begin{aligned} a &= \frac{1}{n-1}\left(\max_{z \in S} z - \min_{z \in S} z\right) \\ &= \frac{1}{n-1}\left(p - \min_{z \in S} z\right) + \frac{1}{n-1}\left(\max_{z \in S} z - p\right) \\ &= \frac{n_1 - 1}{n-1} a_1 + \frac{n_2 - 1}{n-1} a_2. \end{aligned}$$

$n_1 + n_2 = n + 1$ olduğunu unutmayın. Çünkü, p hem S_1 hem de S_2'de yer almakta. Dolayısıyla a, a_1 ve a_2'nin ağırlıklı ortalamasıdır.

$\alpha = (n_2 - 1)/(n - 1)$ iken;

$$a = (1 - \alpha)a_1 + \alpha\, a_2,$$

$a_1 \leq a_2$ olduğunu düşünürsek ve x ve y S_1'de yakın sayılar ise;

$$|x - y| \leq a_1 = (1 - \alpha)a_1 + \alpha a_1 \leq (1 - \alpha)a_1 + \alpha a_2 = a.$$

Bu, S_1'deki her yakın sayı çiftinin aynı zamanda S'de de yakın olduğunu ifade eder. Benzer şekilde eğer $a_2 \leq a_1$ ise, S_2'deki her yakın sayı çifti S'de de yakındır.
S_k'daki en küçük veya en büyük sayı aranarak ortalama uzaklık olan a_k, O(n) süresinde hesaplanabilir. Aynı şekilde S_k altkümesinde de belirtilen özellik O(n) süresinde hesaplanabilir.

(c) S içindeki yakın sayı çiftlerini O(n)-sürede bulacak bir algoritma tanımlayın. Algoritmanızın neden doğru olduğunu ve koşma süresini kısaca açıklayın. (İpucu: Böl ve fethet'i kullanın.)

Çözüm: S'yi yakın bir çift bulana kadar yinelemeli olarak parçalarız.

1. S'nin, S_1 ve S_2'ye bölüntülemek için ortancasını buluruz.
2. Bölüm (b)'deki sonucu, S'nin bir yakın çiftini içeren S_k kümesini belirlemek için kullanırız.
3. S_k 'ya 2 eleman içerene kadar özyineleme uygularız.

Her bir özyineleme adımı, kümenin eleman sayısını yarıya indireceğinden, özyinelemenin sonlanacağı kesindir. Her özyinelemeden sonra geriye kalan küme, S'nin bir yakın sayı çiftini içerir.

Eğer belirleyici ortanca bulma algoritmasını kullanırsak, birinci adım, en kötü durumda, O(n) süresi alır. Bölüm (b)'de gösterdiğimiz gibi ikinci adım O(n) süre alır. Ayrıca Ana Teoreme gore,

$$T(n) = T(n/2) + O(n),$$

yineleme algoritmasının koşma süresi $T(n) = O(n)$'dir.

KARALAMA KAĞIDI – Lütfen, sınav sonunda kitapçıktan ayırın.

KARALAMA KAĞIDI – Lütfen, sınav sonunda kitapçıktan ayırın.