

Pratik Ara Sınav 1

Problem -1. Yinelemeler

Aşağıdaki yinelemeleri, sıkı Θ -simgelemi sınırlarını vererek çözün. Cevabınızı açıklamanıza gerek yok, ancak vereceğiniz bilgiler, gidiş yolundan puan almanızı sağlayabilir.

- (a) $T(n) = T(n/3) + T(n/6) + \Theta(n\sqrt{\log n})$
- (b) $T(n) = T(n/2) + T(\sqrt{n}) + n$
- (c) $T(n) = 3T(n/5) + \lg^2 n$
- (d) $T(n) = 2T(n/3) + n \lg n$
- (e) $T(n) = T(n/5) + \lg^2 n$
- (f) $T(n) = 8T(n/2) + n^3$
- (g) $T(n) = 7T(n/2) + n^3$
- (h) $T(n) = T(n-2) + \lg n$

Problem -2. Doğru veya Yanlış

Aşağıdaki ifadelerle ilgili **D** or **Y**'yi daire içine alarak, ifadelerin doğru mu yanlış mı olduğunu belirtin ve cevabınızı kısaca açıklayın. Anlatımınız ne kadar açık olursa, o kadar yüksek not alırsınız, ama kısa anlatmanız da önemli. Açıklama yapmadığınız takdirde, doğru cevaptan puan alamayacaksınız.

- (a) **D Y** Asimptotik olarak pozitif olan bütün $f(n)$ 'ler için; $f(n) + o(f(n)) = \Theta(f(n))$ 'dir.
- (b) **D Y** En kötü durumda ve beklenen durumdaki koşma süreleri, rastgele algoritmalar için, sabit faktörlere bağlıdır.
- (c) **D Y** $\{A, B, C, D\}$ evrenindeki anahtarları, aşağıdaki tabloya göre $\{0, 1, 2\}$ aralığına eşlemleyen, $H = \{h_1, h_2, h_3\}$ tanımlı 3 kıyım fonksiyonunun koleksiyonu evrenseldir.

x	$h_1(x)$	$h_2(x)$	$h_3(x)$
A	1	0	2
B	0	1	2
C	0	0	0
D	1	1	0

Problem -3. Kısa cevaplar

Aşağıdaki sorulara, kısa ama tam cevaplar verin.

- (a) Herhangi bir karşılaştırmaya dayalı sıralama algoritması, koşma süresini sabit bir çarpandan daha fazla değiştirmeyecek şekilde kararlı olabilir.
- (b) Kıyaslama modelinde, aşağıdaki özelliklerin ikisinin de olması durumunda Öncelikli Kuyruğun olamayacağını tartışın.

EXTRACT-MIN $\Theta(1)$ sürede koşacak.

BUILD-HEAP $\Theta(n)$ sürede koşacak.

- (c) $A[1]$ 'in en büyük anahtar olduğu $A[1,2,\dots,n]$ dizilimi içindeki bir yığın (en büyük yığın) için, aşağıdaki yordamı uygulayacak ve aynı zamanda da en büyük yığın olma özelliğini koruyacak sözde kodu yazın.

DECREASE-KEY(i, δ) - $A[i]$ anahtarının değerini δ kadar küçültür.
 δ 'nin 0'dan büyük veya eşit olduğunu varsayın.

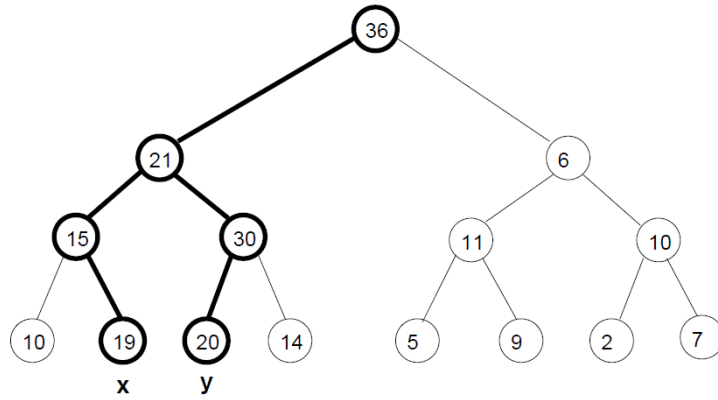
- (d) n tane farklı tamsayıdan (bazıları eksi değerli olabilir) oluşan sıralı A dizilimi için;

$1 \leq i \leq n$ ve $A[i]=i$ olan bir i dizilimini (varsa) bulmak için bir algoritma yazın. Eğer birden fazla buna benzer dizin varsa, algoritmanın herhangi birini vermesi yeterlidir .

Problem -4. h yüksekliğinde ve $n=2^h$ yapraklı bir ikili ağacın verildiğini düşünün.

Bu ağacın her düğümü ve yaprağına bir v değeri (rastgele gerçekte sayı) atanmış durumda. Eğer x bir yaprak ise, x 'i ve x 'in atalarının ve x 'in büyük kümesini $A(x)$ ile ifade ederiz. $A(x)$ köke kadar gider. Benzer şekilde x ve y farklı yapraklarsa,

$$A(x, y) = A(x) \cup A(y)$$



$A(x,y)$ koyu olarak gösterilmiştir.

$$f(x,y) = 19+15+21+36+20+30 = 141$$

$f(x,y)$ fonksiyonunu $A(x,y)$ 'deki düğümlerin değerlerinin toplamı olarak tanımlayın.

$f(x_0, y_0)$ 'ın en büyük olduğu x_0 ve y_0 yapraklarını bulmak için verimli bir algoritma (sözde kod) yazın. Algoritmanızın çalışma süresi nedir?

Problem -5. Küçük çoklu kümelerin sıralanması

Bu problem için, n uzunluğunda, k farklı anahtardan oluşan A diziliminde,

$$k < \sqrt{n}$$

Amacımız bu dizilimi, $\Omega(n \lg n)$ 'den daha hızlı sıralayabilmek. Bunu 2 aşamada yapacağız. Birinci aşamada, A 'daki k farklı anahtarı B *sıralı diziliminde* hesaplayacağız.

İkinci aşamada, B dizilimini kullanarak, A dizilimini sıralayacağız.

k 'nın bir sabit gibi çok küçük olabileceğini ve koşma sürenizin n kadar, k 'ya da bağımlı olabileceğine dikkat edin. n öğenin anahtarlara ek olarak uydu verileri de var.

Örnek:

$$A = \left[5, 10^{10}, \pi, \frac{128}{279}, 10^{10}, \pi, 5, 10^{10}, \pi, \frac{128}{279} \right] \text{ olsun.}$$

Burada $n = 10$ ve $k = 4$ olur.

İlk aşamada,

$$B = \left[\frac{128}{279}, \pi, 5, 10^{10} \right] \text{ hesaplarız.}$$

İkinci aşamadan sonra sıralı dizilimimiz ise,

$$\left[\frac{128}{279}, \frac{128}{279}, \pi, \pi, \pi, 5, 5, 10^{10}, 10^{10}, 10^{10} \right]$$

olur. Hedefiniz, bu iki aşama için de verimli algoritmalar tasarlamak ve bunları çözümlemek. Unutmayın, algoritmanız ne kadar verimli olursa, alacağınız not da o kadar yüksek olur !

- (a) Birinci aşama için, k farklı anahtarı olan k uzunluğundaki sıralı B dizilimini hesaplayacak bir algoritma tasarlayın. k 'nın değeri algoritmaya girdi olarak verilmemektedir.
- (b) (a) bölümündeki algoritmanın çözümlemesini yapın.
- (c) İkinci aşama için bir algoritma tasarlayın; yani bölüm (a)'da yarattığınız B dizilimini kullanarak verilen bir A dizilimini sıralayın. Öğelerin uydu verileri olduğundan, belirli anahtarları olan elemanları sayıp ve kopyalamanın yeterli olmayacağına dikkat edin.
İpucu: Sayma sıralamasını uyarlayın..
- (d) (c) bölümündeki algoritmayı çözümlayın.