

Final Sınavı

- Bu sınav kitapçığını size söylenene kadar açmayın. Bu sayfadaki tüm direktifleri okuyun.
- Sınav başlayınca, bu sınav kitapçığının her sayfasına isminizi yazın.
- Bu sınavda 11 problem var ve bazılarının çoklu şıkları var. 180 puan almak için 180 dakikanız var.
- Bu sınav kitapçığının, bu sayfa dahil 20 sayfası var. 2 ek sayfa da müsvette kağıdı olarak sona eklenmiş. Sınav sonunda kitapçığı teslim ederken müsvelteleri lütfen ayırın.
- Bu sınav kapalı kitap yapılacak. Tuttuğunuz notlar ve formüller için el yazması 2 A4 kağıdı kullanabilirsiniz. Hesap makinesi yada programlanır cihazlara izin verilmez.
- Çözümlerinizi bırakılan boşluklara yazın. Daha fazla yere gerek duyarsanız, sorunun olduğu sayfanın arkasını kullanın. Yanıtlarınızın bir bölümünü, başka bir sorunun olduğu kağıdın arkasına yazmayın, çünkü not verme işleminde kitapçığın sayfaları ayrılabilir.
- İncelediğimiz sonuçları tekrar çıkarmak için zaman ve yer harcamayın. Bilinen sonuçları yazmak yeterlidir.
- Bir problemde çok fazla zaman harcamayın. Önce hepsini okuyun ve soruları en hızlı ilerleme kaydedeceğiniz sırayla ele alın.
- Yaptığınız işi gösterin, çünkü buna kısmi kredi verilecek. Sadece yanıtınızın doğruluğundan değil, onu açıklamadaki anlaşılabilirliğinizden de not alacaksınız. Düzenli olun
- İyi şanslar!

Not kağıdı: Aşağıdaki boşlukları boş bırakın



Prob	Başlık	Kısım	Puan	Not	Notveren
1	Yinelemeler	3	15		
2	Algoritmalar ve koşma süreleri	1	9		
3	Yerine koyma metodu	1	10		
4	Doğru- Yanlış ve Gerekçeleri	7	35		
5	Kırmızı-siyah ağaçlar	3	15		
6	Oynak (wiggly) dizilimler	1	10		
7	Fark kısıtları	2	12		
8	Amortize artış	1	12		
9	En az kapsayan ağaçlar	1	12		
10	Çok izlekli çizelgeleme	1	10		
11	Bir matrisi çaprazlama	4	40		
	<i>Toplam</i>		180		

Problem 1. Yinelemeler [15 puan] (3 bölüm)

Aşağıdaki her yinelemenin, sıkı asimptotik üst sınırını (O simgelemi) yazın. Cevabınızı açıklamanaız gerek yok.

(a) $T(n) = 2T(n/8) + \sqrt[3]{n}$.

(b) $T(n) = T(n/3) + T(n/4) + 5n$

(c) $T(n) = \begin{cases} 8T(n/2) + \Theta(1) & \text{if } n^2 > M, \\ M & \text{if } n^2 \leq M; \end{cases}$

M, n'den bağımsız bir değişken iken.

Problem 2. Algoritmalar ve koşma süreleri [9 puan]

Aşağıdaki algoritmaları en kötü koşma sürelerinin sıkı asemptotik üst sınırları ile eşleştirin. A'dan I'ya harfleri algoritmaların yanındaki kutulara yerleştirin. Sıralama algoritmaları için girdi sayısı olarak n kabul edilecektir. Matris algoritmaları için büyüklük $n \times n$ olarak kabul edilecektir. Grafik algoritmaları için, köşe sayısı olarak n , kenar sayısı olarak ise $\Theta(n)$ kabul edilecektir. Cevaplarınızı açıklamak zorunda değilsiniz. Bazı koşma süreleri birden fazla algoritma için kullanılabileceği gibi, bazıları da hiçbir algoritma ile eşleşmeyebilir. Yanlış cevaptan puan kırılacağı için tahmine dayalı yanıt vermemeniz tavsiye edilir.

☐

Araya yerleştirme sıralama

A: $O(\lg n)$ ☐

Yığın Sıralaması

B: $O(n)$ ☐

YIĞIN-İNŞASI

C: $O(n \lg n)$ ☐

Strassen

D: $O(n^2)$ ☐

Bellman-Ford

E: $O(n^2 \lg n)$ ☐

Derinliğine arama

F: $O(n^{2.5})$ ☐

Floyd-Warshall

G: $O(n^{\lg 7})$ ☐

Johnson

H: $O(n^3)$ ☐

Prim

I: $O(n^3 \lg n)$

Problem 3. Yerine Koyma Metodu [10 puan]

Aşağıdaki yinelemenin sıkı asemptotik alt sınırını (Ω -simgelemi) bulmak için yerine koyma metodunu kullanın.

$$T(n) = 4T(n/2) + n^2 .$$

Problem 4. Doğru veya Yanlış [35 puan] (7 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 özet olsun. Açıklamanız, D veya Y seçiminden daha fazla not getirecek.

D Y A_1, A_2 ve A_3 , birbirinden farklı n doğal sayının sıralı dizilimleri olsun. Karşılaştırma modelinde, $A_1 \cup A_2 \cup A_3$ kümesinin dengeli bir ikili arama ağacına uyarlanması $\Omega(n \lg n)$ süre alır.

D Y T , n tane anahtarlı tam bir ikili ağaç olsun. T 'nin kökünden verilen bir $v \in T$ köşesine, enine arama kullanarak bir yol bulmak $O(\lg n)$ süreye mal olur.

D Y $A[1 \dots n]$, n tane tamsayılı sırasız bir dizilim veriliyor. A 'nın elemanlarından bir maks-yığın oluşturmak, kırmızı-siyah ağaç oluşturmaktan asimptotik olarak daha hızlıdır.

D Y h kıyım fonksiyonu kullanarak, n tane ayrışık anahtarı m uzunluğundaki T dizilimine yerleştirdiğimizi düşünelim. Basit kıyım kullanarak, çiftlerin beklenen çarpışma sayısı:

$$\Theta(n^2/m)$$

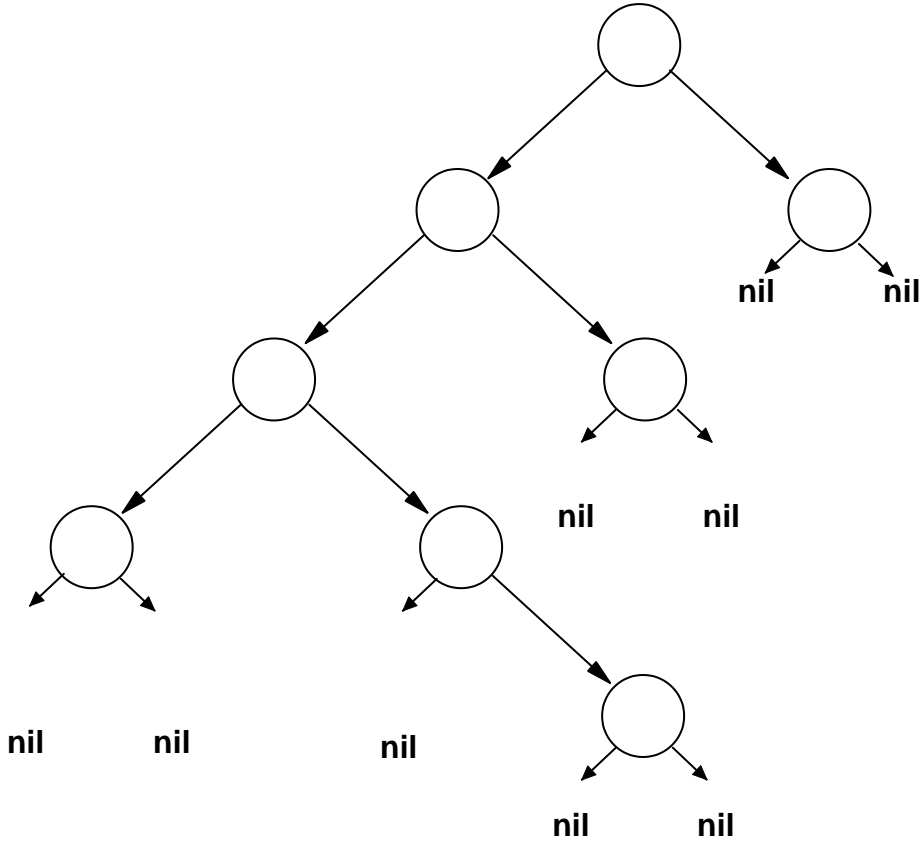
D Y n tane girdili bütün sıralama ağları $\Omega(\lg n)$ derinliğe sahiptir.

D Y Eğer dinamik bir programlama problemi en uygun altyapı özelliğini sağlıyor ise, yerel olarak en uygun çözüm, her yerde en uygundur.

- D Y** $G = (V, E)$, negatif ağırlıklı kenarları olan, ancak negatif ağırlık çevrimi olmayan yönlendirilmiş bir grafik olsun. Öyleyse, $s \in V$ kaynağından tüm $v \in V$ 'lere bütün en kısa yollar, yeniden ağırlıklandırma tekniğini kullanarak Bellman-Ford'dan daha hızlı hesaplanabilir.

Problem 5. Kırmızı-siyah ağaçlar [15 puan] (3 bölüm)

- (a) 2, 3, 5, 7, 11, 13, 17, 19 anahtarlarını, aşağıdaki ikili arama ağacına, ikili arama ağacı özelliklerini karşılayacak şekilde yerleştirin.



- (b) Bu ikili arama ağacının kurallara uygun bir kırmızı-siyah ağaç olarak renklendirilememesinin nedenini açıklayın.

- (c)** Bir tane döndürme yapılarak ikili arama ağacı, bir kırmızı-siyah ağaca dönüştürülebilir. Bunu sağlayan kırmızı-siyah ağacı, her düğümü kırmızı veya siyah olarak etiketleyerek çizin. Bölüm (a)'daki anahtarları kullanın.

Problem 6. Wiggly (oynak) dizilimler [10 puan]

$A[1 \dots 2n + 1]$ dizilimi, eğer $A[1] \leq A[2] \geq A[3] \leq A[4] \geq \dots \leq A[2n] \geq A[2n + 1]$ ise wiggly yani oynaktır. Sıralı olmayan ve gerçekte sayılardan oluşan bir $B[1 \dots 2n + 1]$ dizilimi verildiğinde, B'nin bir oynak alt dizilimi olacak şekilde $A[1 \dots 2n + 1]$ permutasyonunu çıkaracak verimli bir algoritma tanımlayın.

Problem 7. Fark Kısıtları [12 puan] (2 bölüm)

Aşağıdaki fark kısıtlarının doğrusal programlama sistemini düşünün. (bir kısıtın eşitlik olduğuna dikkat edin):

$$x_1 - x_4 \leq -1$$

$$x_1 - x_5 \leq -4$$

$$x_2 - x_1 \leq -4$$

$$x_2 - x_3 = -9$$

$$x_3 - x_1 \leq 5$$

$$x_3 - x_5 \leq 2$$

$$x_4 - x_3 \leq -3$$

$$x_5 - x_1 \leq 5$$

$$x_5 - x_4 \leq 1$$

(a) Bu kısıtlar için, kısıt grafiğini çizin.

(b) x_1 , x_2 , x_3 , x_4 ve x_5 bilinmeyenleri için çözümü bulun veya neden çözüm olmadığını açıklayın.

Problem 8. Amortize edilmiş artım (increment) [12 puan]

Bitlerden oluşan $A[0 \dots k-1]$ dizilimi (her eleman 0 veya 1) bir ikili x sayısını saklamaktadır

$$x = \sum_{i=0}^{k-1} A[i] \cdot 2^i$$

(2^k modunda) 1'i x 'e eklemek için aşağıdaki yöntemi takip ederiz.

INCREMENT (A, k)

```

1   $i \leftarrow 0$ 
2  while  $i < k$  and  $A[i] = 1$ 
3      do  $A[i] \leftarrow 0$ 
4       $i \leftarrow i + 1$ 
5  if  $i < k$ 
6      then  $A[i] \leftarrow 1$ 
```

Verilen bir x sayısı için, potansiyel $\phi(x)$ 'i x 'in ikili yazımındaki 1'lerin sayısı şeklinde tanımlayın. Mesela, $\phi(19)=3$, çünkü $19 = 10011_2$. Potansiyel fonksiyon tartışmasını, bir artımın amortize maliyetinin, x 'in başlangıç değeri $x=0$ iken, $O(1)$ olduğunu ispatlamak için kullanın.

Problem 9. En az yayılan/kapsayan ağaçlar [12 puan]

$G = (V, E)$, $w : E \rightarrow \mathbb{R}$ kenar ağırlık fonksiyonu olan bağlantılı ve yönsüz bir grafiktir. Bu grafikte bütün kenarların ağırlıklarının farklı olduğunu varsayın. G 'deki $(v_1, v_2, \dots, v_k, v_{k+1})$ çevriminde $v_{k+1} = v_1$ olduğunu ve (v_i, v_{i+1}) kenarının çevrimdeki en ağır kenar olduğunu varsayın. (v_i, v_{i+1}) kenarının G 'nin en az yayılan ağacı T 'ye ait olmadığını ispatlayın.

Problem 10. Çok izlekli çizelgeleme [10 puan]

Aç gözlü bir çizelgeleyici, çok izlekli bir hesaplamayı, 4 işlemci ile 260 saniyede, 32 işlemci ile 90 saniyede bitiriyor. $T_1 = 1024$ ve kritik yol uzunluğu $T_\infty = 64$ iken hesaplamamanın çalışma ihtimali var mı? Cevabınızı açıklayın.

Problem 11. Bir matrisin ters çevrilmesi [40 puan] (4 bölüm)

X , $N \times N$ boyutlu ve N 'nin 2'nin tam kuvvetinde olduğu bir matris olsun. Aşağıdaki kod $Y = X^T$ 'yi hesaplar:

```

TRANS(X, Y, N)
1  for i ← 1 to N
2      do for j ← 1 to N
3          do Y[j, i] ← X[i, j]

```

Önbelleği hesaba katmayan 2 seviyeli hafıza modelinde, ön belleğin M tane eleman ve B tane blok elemana sahip olduğunu düşünün. X ve Y matrislerinin ikisinin de satır öncelikli düzende düzenlendiğini, X 'in doğrusal düzeninin bellekte, $X[1, 1], X[1, 2], \dots, X[1, N], X[2, 1], X[2, 2], \dots, X[2, N], \dots, X[N, 1], X[N, 2], \dots, X[N, N]$, olduğunu ve aynı şeyin Y için de geçerli olduğunu varsayın.

(a) TRANS'daki, $N \gg M$ iken meydana gelen hafızaya aktarım sayısı $MT(N)$ 'yi çözümleyin.

Şimdi, ters çevirmeyi yapabilmek için böl ve fethet yöntemini ele alalım:

R-TRANS(X, Y, N)

```

1  if  $N = 1$ 
2    then  $Y[1, 1] \leftarrow X[1, 1]$ 
3    else  $X$ 'i 4 parçaya bölüntüle  $(N/2) \times (N/2)$  altmatrisler  $X_{11}, X_{12}, X_{21}$ , ve  $X_{22}$ .
4         $Y$ 'i 4 parçaya bölüntüle  $(N/2) \times (N/2)$  altmatrisler  $Y_{11}, Y_{12}, Y_{21}$ , ve  $Y_{22}$ .
5        R-TRANS( $X_{11}, Y_{11}, N/2$ )
6        R-TRANS( $X_{12}, Y_{21}, N/2$ )
7        R-TRANS( $X_{21}, Y_{12}, N/2$ )
8        R-TRANS( $X_{22}, Y_{22}, N/2$ )

```

Bölüntülemenin maliyetinin $O(1)$ olduğunu varsayın.

(b) R-TRANS'daki, $N \gg M$ iken oluşan $MT(N)$ hafızaya taşıma sayısının yinelenmesini verin ve çözün.

Aşağıdaki çokizlekli algoritma, matris ters çevirmesini paralel olarak hesaplar.

```

P-TRANS(X, Y, N)
1  if N = 1
2    then Y[1, 1] ← X[1, 1]
3  else X'i 4'e bölüntüle (N/2) × (N/2) altmatrisler X11, X12, X21, and X22.
4      Y'i 4'e bölüntüle (N/2) × (N/2) altmatrisler Y11, Y12, Y21, and Y22.
5      spawn P-TRANS(X11, Y11, N/2)
6      spawn P-TRANS(X12, Y21, N/2)
7      spawn P-TRANS(X21, Y12, N/2)
8      spawn P-TRANS(X22, Y22, N/2)
9      sync

```

- (c) $T_1(N)$ için yinelemeler verin ve çözün ve P-TRANS'ın kritik yol uzunluğu $T_\infty(N)$ 'yi bulun. Algoritmanın asimptotik paralelliği nedir?

Professor Kellogg koduna dikkatsizce, aşağıdaki gibi 2 yeni **sync** komutu ekliyor.

K-TRANS(X, Y, N)

```
1  if N = 1
2    then Y[1, 1] ← X[1, 1]
3  else X'i 4'e bölüntüle (N/2) × (N/2) altmatrisler X11, X12, X21, and X22.
4      Y'i 4'e bölüntüle (N/2) × (N/2) altmatrisler Y11, Y12, Y21, and Y22.
5      spawn K-TRANS(X11, Y11, N/2)
6      sync
7      spawn K-TRANS(X12, Y21, N/2)
8      sync
9      spawn K-TRANS(X21, Y12, N/2)
10     spawn K-TRANS(X22, Y22, N/2)
11     sync
```

(d) $T_1(N)$ 'nin işini yapacak yinelemeyi verin ve çözün ve K-TRANS'ın kritik yol uzunluğu $T_\infty(N)$ 'yi bulun. Algoritmanın asimptotik paralelliği nedir?

KARALAMA KAĞIDI – Lütfen, sınav sonunda kitapçıktan ayırın.