

Ara Sınav 2

Bu sınav, her biri 25 puan değerinde 6 sorudan oluşmakta. Toplamda 150 puan toplayabilirsiniz. Her soru, ayrı bir 3-delikli kağıda cevaplanmalı.

Her sayfaya isminizi, 6.046J/18.410J başlığını, problem numarasını, etüt zamanınızı ve araştırma görevlinizi yazın. **Sınavınızı 2 Aralık saat 09:00 ile 11:00 arasında teslim edin.** Geç teslim edilen sınavlar, dekan onayı olmadan veya etüt öğretmeninizle öncelikli başka bir ayarlama yapılmadan kabul edilmeyecektir. Sınavı kendi elinizle teslim etmek zorundasınız.

Sınavın çözülmesi için 12 saat yeterlidir. Ancak sizin önünüzde 4 gününüz var. Zamanınızı akıllıca planlayın. Aşırı çalışmayın, uykunuzu alın. İyi yazılmış çözümler yüksek puan alacaktır. Tabii ki asimptotik sınırlarınız ne kadar iyi olursa o kadar yüksek puan alırsınız. Sıra dışı verimli ve seçkin cevaplara Bonus puanlar verilecektir.

Yazım:

Her problem ayrı bir kağıtta veya kağıtlarda çözülmeli. Her sayfanın üzerine adınızı, 6.046J/18.410J'yi, soru numarasını, etüt saatinizi ve araştırma görevlinizin adını yazmayı unutmayın. Herhangi bir soruya cevabınız, çözümünüzün özeti ile başlamalı. Bu özet, çözdüğünüz problemi açıklamalı, kullandığınız teknikleri, önemli varsayımları ve algoritmanızın koşma süresini içermelidir.

Cevaplarınızı bizim anlamamızı kolaylaştırmak için açıkça ve temiz yazın. Koşma süreleri ve algoritmalar ile ilgili kesin ifadeler kullanın. Mesela, sadece " n sayıyı sıralarız" demeyin. Onun yerine "Yığın sıralamasını kullanarak n sayıyı, en kötü durumda $O(n \lg n)$ sürede sıralarız"ı tercih edin. Bir algoritmayı açıklarken ana fikri Türkçe yazın. Sözde kodu sadece çözümünüzü daha açık hale getirmek gerekirse kullanın. Örnekler verin ve şekiller çizin. Çözümlerinizin doğruluğunu ispatlamak için az, öz ve ikna edici ifadeler kullanın. Sınıfta işlenen konuları defalarca tekrarlamayın. Sınıfta, ek derste veya kitapta geçen algoritma ve teoremlere referans vererek çözümünüzü basitleştirin.

Bu sınavın bir amacı da sizin mühendislik mantığınızı sınamaktır. Eğer bir soruyu şüpheli bulduysanız veya açık bulmadıysanız, mantıklı varsayımlar yapın. Ancak ne varsaydığınıza dikkat edin, çünkü eğer bir problemi kolaylaştıran çok güçlü bir varsayımda bulunursanız, o sorudan az puan alırsınız.

Hatalar, vs.: Eğer herhangi bir soruda bir hata bulduğunuzu düşünüyorsanız, bize e-posta atın. Düzeltmeler ve açıklamalar, bir e-posta ile ve ağ sayfası üzerinden sınıfa duyurulacaktır. E-postanızı ve ders sayfasını günlük olarak kontrol edin. Böylece önemli olabilecek açıklamaları kaçırmazsınız.

Akademik Dürüstlük İlkesi: Sınav Sınırlı Açık Kitap şeklindedir. Ders notlarınızı, ders kitabınızı, sözlükleri ve dersin sayfasındaki kitapçıkları kullanabilirsiniz. Ancak bunun haricindeki herhangi bir kaynağı kullanmanız yasaktır. Mesela, bu dersin daha önceki dönemlerden gelen ders notlarını veya benzer derslerin ders notlarını veya internetteki içeriği kullanmanız yasaktır. Sınavınızı teslim etmiş olsanız bile, dersin öğretim kadrosu hariç kişilerle sınav ile ilgili iletişim kurmanız, 2 Aralık'a kadar yasaktır.

Her hangi bir anda, bu ilkeyi ihlal ettiğinizi düşünürseniz, dersin eğitmenleriyle bir an önce irtibata geçin. Eğer bir kaynağın kullanılıp kullanılamayacağına dair sorunuz olursa bu soruyu sınavdan önce dersin eğitmenlerine e-posta ile bildirebilirsiniz.

Anket: Bu sınava ekli olarak bir anket bulacaksınız. Bu anket özellikle akademik dürüstlüğü ilişkin sınav tecrübelerinizle ilgili. Lütfen bu anketi sınav kağıtlarından ayırıp doldurun ve sınavla beraber teslim edin. Bu anket isimsiz olarak teslim edilecektir. Anketteki herhangi bir cevaptan ötürü kişilere bir sorumluluk yüklenmeyecektir. Bu bilgiler sınavın faydasını ölçmek için kullanılacak ve istatistiklerin özeti sınıfta duyurulacaktır.

**Lütfen Sınav Süresince yukarıdaki açıklamaları her gün tekrar okuyun.
İYİ ŞANSLAR, İYİ EĞLENCER!**

Problem 1. Yukarılar ve aşağılar

Akşam olunca, Prof. Silvermeadow Ulusal Teknoloji Üniversitesi'ndeki işinden uzaklaşıp, gece kulüplerinde sihir gösterileri yapmakta. Profesör, aşağıdaki kart numarasını geliştirmekte. 1'den n 'ye kadar, n tane yüzü yukarı bakan kart var. İzleyicilerden biri, $[i, j]$, arasında bir aralık vermekte. Profesör, $i \leq k \leq j$ olan her k kartını ters çevirmekte. Bu işlem sırasında izleyiciler profesörü hangi kartların yüzlerinin yukarı, hangilerinin aşağı olduğu konusunda sorguluyor ve işlem defalarca tekrarlanıyor. Aslında, gerçek kartlar yok. Profesör manipülasyonu kafasında yapmakta ve n oldukça büyük. Profesör manipülasyonu yapabilmek için izleyenlere belli etmeden bir hesaplama aracı kullanıyor ancak, mevcut sürüm çok yavaş ve gerçek zamanlı olarak çalışmıyor. Profesöre verimli bir veri yapısı tasarlayarak yardımcı olun. Bu veri yapısı aşağıdaki işlemleri yapmalı:

- FLIP(i, j): $[i, j]$ aralığındaki bütün kartları ters çevirsin.
- IS-FACE-UP (i): Eğer i kartı yukarı bakıyorsa DOĞRU, aşağı bakıyorsa YANLIŞ döndürsün.

Problem 2. Veri Merkezi

Dünyaca ünlü mimar Gary O'Frank, Veri Merkezi adında yeni bir bina tasarlaması için görevlendirilir. Gary mimari tasarım asistanından, Veri Merkezinin modelini tasarlarken, hassas kesim çubuklar kullanmasını, ama istenmeyen dik açılardan da kaçınmasını ister. Gary, $1, 2, \dots, n$ etiketli, i çubuğu x_i uzunluğunda, n tane çubuktan oluşan bir küme üretir. Gary, çubukları asistanına vermeden size gösterir ve bunlardan herhangi üçünü kullanarak bir dik üçgen oluşturup oluşturamayacağınızı sorar. a , b , c çubuklarıyla böyle bir üçgen oluşturulup oluşturulamayacağını belirleyen verimli bir algoritma bulun. Bu üçgenin köşe uzunlukları; x_a , x_b ve x_c olsun. (Dolayısıyla $x_a^2 + x_b^2 = x_c^2$ olmalı.)

Problem 3. Bir matrisi, pivot kullanarak negatiflikten kurtarma

Bir $M[1 \dots n, 1 \dots n]$ matrisi, $\mathbb{R} \cup (-\infty, \infty)$ 'dan alınan girdiler içerir. Her satır en fazla 10 adet sınırlı, ama bazıları negatif olabilecek değer içerir. Amacımız, pivot kullanarak M 'yi dönüştürerek, her girdisini negatif olmaktan kurtarmaktır.

```
PIVOT(M, i, x)
1  for j ← 1 to n
2      do M[i, j] ← M[i, j] + x
3      M[j, i] ← M[j, i] - x
```

i ve x 'in çeşitli değerlerinde bir dizi pivotlama işlemi kullanarak sonuçta $i, j = 1, 2, \dots, n$ için, $M[i, j] \geq 0$ olup olamayacağını belirlemek için verimli bir algoritma bulun.

Problem 4. Queueinator™ 'yi geliřtirmek

Profesör Uriah'ın řirketi, profesörün sıcak fizyon konusundaki arařtırmalarıyla geliřtirdiđi Queueinator™ 'yi üretmeye ve satmaya bařladı. Queueinator™ öncelikli bir kuyruk donanımı ve normal bir bilgisayara bađlanarak verimli řekilde öncelikli kuyruk iřlemleri olan INSERT(araya yerleřtirme) EXTRACT-MIN(en küçüđü çıkarma) iřlemlerini $O(1)$ sürede yapabiliyor. Profesörün řirketinin bir müřterisi var, ancak müřteri “iki uçlu” bir öncelikli kuyruk uygulaması donanımı istiyor. Sadece INSERT ve EXTRACT-MIN iřlemlerini deđil ayrıca EXTRACT-MAX iřlemini de yapan bir cihaz gerekmekte. Queueinator™'ın yeniden tasarımı profesörün řirketinin yaklařık 1 yılını alacak. Profesöre verimli bir “iki-uçlu öncelikli kuyruk” yazılımı ve bir ya da daha fazla Queueinator™ cihazı tasarlaması konusunda yardımcı olun.

Problem 5. Spam (mesaj yağanağı) dağıtımı

Professor Hormel spam dağıtımı için bir ağ tasarlamakta. Bu ağ, $T = (V, E)$ ile simgelenen ve kökü $r \in V$ olan bir ağaç yapısında; ağacın köşe ağırlık fonksiyonu $w: E \rightarrow \mathbb{R}$ ve negatif değil. Her $v \in V$ olan köşe bir milyon e-posta adresini barındıran bir sunucuyu temsil ediyor ve her $e \in E$ olan kenar ise maliyeti $w(e)$ dolar olan bir iletişim kanalını temsil ediyor. Kök r 'den v 'ye giden tüm yol satın alınırsa spam $v \in V$ sunucusuna ulaşıyor. Profesör, kök r 'den $k \leq |V|$ sunucularına (kök dahil) ucuza spam göndermek istiyor. Profesöre en az ağırlıklı ve bağlantılı, kök dahil k köşesi olan T alt ağacını bulacak bir algoritma tasarlayarak yardımcı olun. (Kısmi not için, problemi her $v \in V$ köşesinin T içinde 2 ardılının olması durumunda çözün.)

Problem 6. Günde bir domates

Profesör Kerry domateslere bayılır! Profesör, hergün bir domates yer. Çünkü profesör, domatesin içindeki antioksidan likopenin sağlığa faydaları hakkında takıntılıdır ve onları yemeyi çok sever. Domates fiyatları yıl içinde artar ve azalır ve doğal olarak fiyatlar düşükken, profesör alabildiği kadar çok domates almak ister. Ancak domatesin belli bir raf ömrü olduğundan (d gün), profesör, i gününde aldığı domatesi $i \leq j < i + d$ aralığında bir j gününde yemelidir veya domates boşa gider ve çöpe atılır. Profesör, herhangi bir gün ne kadar çok domates alırsa alsın, hergün sadece bir domates yediğinden, fiyat çok düşük olsa da çok fazla domates almamaya dikkat etmelidir.

Profesörün takıntısı, onu domatese çok para harcadığı konusunda bir endişeye itti. Profesör, geçmiş n günün domates fiyat verilerini edindi ve domatese ne kadar para harcadığını biliyor. Geçmiş fiyat verileri $C[i]$, i günündeki fiyatı temsil eden $C[1 \dots n]$ diziliminde saklanıyor. Profesör, geçmiş verilere bakarak, en az ne kadar para harcayarak günde bir domates yiyebileceğini öğrenmek istiyor. Bunu kendi yaptığı harcama ile karşılaştıracak.

Geçmiş verilere dayanarak en iyi çevrimdışı (20/20 sonradan belirlenimci) satınalma stratejisini belirleyebilmek için verimli bir algoritma yazın. d , n ve $C[1,2,\dots,n]$ verildiğinde, algoritmanızın çıktısı $B[1,\dots,n]$ 'yi vermeli ve $B[i]$, i 'inci günde satın alınacak domateslerin sayısı olmalı.