

Problem Seti 3 Çözümler

Problem 3-1. Örüntü Eşleme (Pattern matching)

Müdür Skinner'ın bir problemi var: Skinner, Bart Simpson'ın, son kitap raporunda bazı metinleri aşırıldığından kesinlikle emin. Bart'ın cümlelerinden biri, kulağa tuhaf biçimde tanıdık geliyordu ama, Skinner bunun nereden alındığını tam olarak anlayamadı ve bazı bilgiç MIT öğrencilerinin ona yardım edip edemeyeceğini görmeye karar verdi.

Skinner size içinde Springfield halk kütüphanesinin tüm metinlerinin olduğu bir DVD verir. Veri; $T[i]$ 'in ya 0 ya da 1 olduğu, $T[1..n]$ dizilimi olarak gördüğümüz, $T[1], T[2], \dots, T[n]$, **ikili dizgisi** içinde depolanmıştır Aynı zamanda Skinner; Bart Simpson'ın kitap raporundan bir alıntıyı da size verir; $P[1..m]$ kısa ikili dizgisinde, yine her $P[i]$, ya 0 ya da 1 ve $m < n$ 'dir. $A[1..k]$ ikili dizgisinde, i, j tamsayıları için $1 \leq i \leq j \leq k$ olduğunda, $A[i], A[i+1], \dots, A[j]$ ikili dizgisi yerine, $A[i..j]$ simgelemine kullanırız ve buna, A 'nin **altdizgisi** deriz. Bu problemin amacı, P 'nin T 'nin bir altdizgisi olup olmadığını belirlemek, yani bazı i, j 'ler için $1 \leq i \leq j \leq n$ olduğunda $P = A[i..j]$ olup olmadığına karar vermektir.

Bu probleminde, $O(\log n)$ -bit tamsayılarını sabit zamanda işleyebileceğinizi farzedin. Örneğin, eğer $x \leq n^7$ ve $y \leq n^5$ olsa, $x + y$ 'yi sabit zamanda hesaplayabilirsiniz. Diğer yandan, m -bitlik tamsayıların sabit zamanda işlenebileceğini varsayamazsınız, çünkü m çok büyük olabilir. Örneğin, eğer $m = \Theta(\log^2 n)$ ve x ve y 'nin her biri m -bitlik tamsayılarıysa, $x + y$ 'yi sabit zamanda hesaplayamazsınız. (Genellikle, girdi boyutu logaritmik uzunlukta olan tamsayıları sabit zamanda işleyebileceğinizi varsaymak mantıklıdır, ancak daha büyük tamsayılar orantılı olarak fazla zaman gerektirir.)

- (a) Elimizde; bazı $A[1..k]$ ve bazı $1 \leq i \leq k - m + 1$ ikili dizgileri için, $x = A[i..(i + m - 1)]$ ikili dizgisinin m -bitlik kıyım değerini hesaplayan $h(x)$ kıyım fonksiyonu olduğunu farz edelim. Ayrıca, kıyım fonksiyonunun mükemmel olduğunu farz edelim. Eğer $x \neq y$ ise, $h(x) \neq h(y)$ 'dir. Kıyım fonksiyonunu $O(m)$ sürede hesaplayabileceğinizi farz edin. P 'nin, T 'nin altdizgisi olup olmadığını $O(mn)$ sürede nasıl belirleyeceğinizi gösterin.

Çözüm: Örüntü dizgisinin kıyımını hesaplarız ve A 'nın tüm olası m -uzunluklu kıyımlarıyla kıyaslarız; yani $1 \leq i < n - m + 1$ için, $h(P)$ 'yi, $h(A[i..(i + m - 1)])$ ile karşılaştırırız. Kıyım fonksiyon mükemmel olduğundan, $h(P) = h(A[i..(i + m - 1)])$ eğer ve sadece eğer $P = A[i..(i + m - 1)]$ olduğunda olabilir. Ortada $O(n)$ kıyım fonksiyonu hesaplaması ile kıyım değerlerinin $O(n)$ sayıda kıyaslaması vardır ve $O(mn)$ toplam koşma zamanı için her hesaplama ve kıyaslama, $O(m)$ süresi gerektirir.

Kıyım fonksiyonunun hesaplaması $O(m)$ süre aldığından, bu algoritma asimptotik olarak, sadece altdizgileri kıyaslamaktan daha iyi değildir. Bu bölüm problemin geri kalanı için motivasyonu arttırmak amacıyla tasarlandı.

- (b) Aşağıdaki h_p kırım fonksiyonları ailesini düşünün; bu fonksiyonlar $[2, cn^4]$ değer kümesindeki p asal sayısı tarafından, bir $c > 0$ sabiti için parametrelendirilmiş :

$$h_p(x) = x \pmod{p}.$$

Farz edelim ki $p, [2, cn^4]$ değer kümesindeki tüm asal sayılar içinden, tekdüze ama rastgele olarak seçildi.

Birkaç i 'yi $1 \leq i \leq n - m + 1$ arasında sabitleyelim ve $x = T[i..(i + m - 1)]$ olsun. Uygun bir c seçildiğinde, eğer $x \neq P$ ise,

$$\Pr_p \{h_p(x) = h_p(P)\} \leq \frac{1}{n}.$$

olur

İpucu: Sayılarla ilgili şu 2 teorik olguyu hatırlayalım: (1) bir x tamsayısının en fazla $\lg x$ asal faktörü vardır; (2) Asal Sayı Teoremi: $[2, x]$ değer kümesinde $\Theta(x / \lg x)$ asal sayı vardır.

Çözüm: Hem x hem de P , sadece $(x - P) = 0 \pmod{p}$ olduğunda, aynı kırım değerine sahiptir. yani, eğer $p, x - P$ 'nin bir faktörü ise. x ile P , m -bitlik sayılar olduklarından, $(x - P)$ en fazla $m + 1 \leq n$ bit'e sahiptir. $(x - P) \leq 2^n$ olduğundan $(x - P)$ 'nin en fazla $\lg 2^n = n$ asal faktörü olur.

Asal Sayı Teoremine göre; $[2, cn^4]$ aralığında, en az $\Omega\left(\frac{cn^4}{\lg cn^4}\right)$ asal vardır.

$$\Pr\{h_p(x) = h_p(P)\} \leq \frac{n}{\Omega\left(\frac{cn^4}{\lg cn^4}\right)} = O\left(\frac{n(\lg c + 4 \lg n)}{cn^4}\right).$$

Uygun seçilmiş sabitler için, bu olasılığın $O(1/n)$ olduğunu gösterebiliriz.

Daha resmi argüman şöyledir: Öyle n' ve d sabitleri vardır ki, tüm $n > n'$ değerleri için, $[2, cn^4]$ aralığındaki asalların sayısı en az $c' (cn^4 / \lg cn^4)$ olur. Bu nedenle, tüm $n > n'$ ler için, $(x - P)$ 'yi bölecek bir p seçme olasılığı en çok:

$$\Pr\{h_p(x) = h_p(P)\} \leq \frac{n}{c' \left(\frac{cn^4}{\lg cn^4}\right)} = \frac{n(\lg c + 4 \lg n)}{c' cn^4}.$$

$n \geq \max\{2 \lg c, 64\}$ ise, $\lg c + 4 \lg n < n$ olduğunun sağlanması kolaydır. Bu nedenle eğer $c > 1/c'$ ve $n \geq \max\{n', 2 \lg c, 64\}$ seçersek şunu elde ederiz:

$$\Pr\{h_p(x) = h_p(P)\} \leq \frac{1}{c' cn^2} \leq \frac{1}{n^2} \leq \frac{1}{n}.$$

- (c) Şık (b) de tanımlanan $h_p(x)$ 'i hesaplamak ne kadar süre alır? *İpucu:* x 'in m -bitlik bir tamsayı olduğuna ve artık sabit sürede işlenemeyeceğine dikkat edin.

Dağıtım 13: Problem Seti 3 Çözümler

Çözüm: $p \leq cn^4$ olduğundan, biliyoruz ki p 'nin $O(\lg n)$ bit'i vardır. $O(\lg n)$ -bitlik tamsayıları sabit sürede işlediğimizi varsayarsak, eğer w 'nin de $O(\lg n)$ biti varsa, $w \pmod{p}$ 'yi sabit zamanda hesaplamak mümkündür.

Eğer x , m -bit'li bir sayıysa ve $m = \omega(\lg n)$ ise hesaplama, giderek daha karmaşık hale gelir. Onun yerine, x ön-ekleri olan kıyım fonksiyonlarını adım adım hesaplayarak, $h_p(x)$ 'i $O(m)$ sürede hesaplarız.

$y, x = A[1.. m]$ dizgisinin k en önemli bitleri olsun ve $h_p(y) = y \pmod{p}$ 'yi daha önceden hesaplamış olduğumuzu farzedelim. Ardından, sabit sayıda işlemle x 'in en önemli $(k + 1)$ bitleri olan z 'nin kıyım fonksiyonunu hesaplayabiliriz.

$z = A[1.. (k + 1)]$ dizgisini bir sayı olarak yorumlarsak, $z = 2y + A[k + 1]$ olur. Böylece $h_p(z)$ tam olarak:

$$h_p(z) = (2y + A[k + 1]) \pmod{p} = (2h_p(y) + A[k + 1]) \pmod{p} \text{ dir.}$$

$h_p(y)$ verilmiş ise $h_p(z)$ 'yi hesaplamak, sadece üç ilave tamsayı işlemi gerektirir: Bir sola kaydırma, toplama ve bir bölme. Böylece, x 'in her bit'i için, sabit miktarda iş yaparız.

Bu bölümün amacı; $h_p(x)$ 'in sabit sürede hesaplanamayacağını fark etmektir ve bu nedenle, Şık (d)'deki algoritma asimptotik bir gelişmeyi gösterir. $O(m)$ yanıtı kabul edilebilir. Bu hesaplamayı, x 'i her biri $c \lg n$ bit uzunluğunda, $m/(c \lg n)$ basamağa bölerek daha hızlı yapabileceğimize dikkat edin. Böylece önceki algoritma, bu şekilde değiştirilerek $h_p(x)$, $O(m / \lg n)$ sürede hesaplanabilir.

- (d) $1 \leq i \leq n - m$ için, Şık (b)'de gösterildiği gibi $h_p(A[i.. (i + m - 1)])$ 'in değerini önceden biliyorsak, sabit sürede, $h_p(A[(i + 1).. (i + m)])$ 'yi nasıl hesaplayacağınızı gösterin.

Çözüm: Dizgileri çift sayı olarak yorumladığımızda, şu denklem çıkar:

$$A[(i + 1) .. (i + m)] = 2 \left(A[i .. (i + m - 1)] - 2^{m-1} A[i] \right) + A[i + m].$$

Bu denklemin her iki tarafının modulo p 'sini aldığımızda şuna ulaşırız;

$$h_p(A[(i+1) .. (i+m)]) = \left(2 \left(h_p(A[i .. (i + m - 1)]) - 2^{m-1} A[i] \right) + A[i + m] \right) \pmod{p}.$$

$2^{m-1} \pmod{p}$ değerinin ön-hesaplamasını yapmış olduğumuzu ve bu değeri bir α değişkeninde depoladığımızı farz edelim. $h_p(A[(i + 1).. (i + m)])$ 'yi hesaplamak için, $\alpha A[i] \pmod{p}$ 'yi hesaplarız ve bu değeri, $h_p(A[i.. (i + m - 1)])$ 'dan çıkarırız. Ardından, 2 ile çarpmak için 1 bit sola kaydırırız, $A[i + m]$ bit'ini buna ekleriz ve kalan p modülüsünü hesaplarız. Kıyım değerlerimiz $O(\lg n)$ -bitlik tamsayılar olduğundan, tüm bu işlemler sabit sürede yapılabilir.

- (e) Şık (b)' deki kıyım fonksiyonları ailesini kullanarak, $O(n)$ beklenen zamanda, P ' nin T ' nin altdizgisi olup olmadığını belirlemek için bir algoritma tasarlayın

Çözüm: P kıyımını, A ' nin tüm m -uzunluğundaki alt-dizgi kıyım fonksiyonlarıyla karşılaştırıp, eşleşmeyi bulana veya A metnini bitirene kadar, Şık (a)' daki algoritmayı kullanabiliriz.

Şık (b)' de belirtilen kıyım ailesinden seçilmiş olan, h_p kıyım fonksiyonunu ve kıyım fonksiyonlarını adım adım hesaplamak için de Şık (d)' deki yöntemi kullanırız. h_p kusursuz bir kıyım fonksiyonu olmadığından, eğer kıyım değerlerinin eşleştiğini keşfedersek, eşit olup olmadıklarını görmek için, iki dizgiyi birbiriyle karşılaştırırız.

Koşma süresini çözümlmek için, iki ayrı maliyeti analiz ederiz: her alt-dizgi için kıyım fonksiyonunu hesaplama maliyeti ve kıyım değerleri eşleştiğinde, alt-dizgileri karşılaştırma maliyeti.. Bu beklenen iki koşma süresinin de $O(n)$ olduğunu gösterir ve istenen sonucu alırız.

İlk $h_p(A[1.. m - 1])$ kıyım fonksiyonunu hesaplamak, $O(m)$ zamanı gerektirir. Öte yandan, Şık (d)' deki yöntemle tüm m -uzunluklu dizgilerinin kıyım değerlerini hesaplamak, toplam $O(m + n) = O(n)$ zamanı gerektirir.

Eğer kıyım değerleri eşleşirse , iki m -uzunluklu altdizginin karşılaştırma maliyeti $O(m)$ olur. i dizininde sahte pozitiflik varsa, X_i , değeri 1 olan bir göstergesel rastgele değişkeni olsun, yani, eğer

$$h_p(P) = h_p(A[i.. (i + m - 1)]), \text{ ama } P \neq A[i.. (i + m - 1)]. \quad \text{ise}$$

Şık (b)'ye göre $E[X_i] < 1/n$.

Öyleyse, sahte pozitiflik için alt-dizgileri karşılaştırmanın beklenen toplam maliyeti şu olur:

$$\sum_{i=1}^{n-m+1} E[X_i]O(m) = O(m) = O(n).$$

Eşleşme için, alt-dizgilerin karşılaştırma maliyeti $O(m) = O(n)$ ' dir, çünkü algoritma, bir eşleşme bulduktan sonra durur.

Problem 3-2. 2-Evrensel Kıyımlama

H , her $h \in H$ ' nin U anahtarlar evrenini $\{0,1,\dots, m-1\}$ 'ya eşlemediği bir kıyım fonksiyonları grubu olsun . Eğer her sabit $\langle x, y \rangle$ anahtar-çifti için $x \neq y$ olursa, ve h, H 'nin içinden birbiçimli ve rastgele seçilirse, $\langle h(x), h(y) \rangle$ çiftinin $\{0,1,\dots, m-1\}$ içindeki m^2 eleman çiftlerinden eşit olasılıkla biri olması durumunda, H **2-evrensel**dir deriz. (Olasılık kıyım fonksiyonunun sadece rastgele seçimi ile hesaplanır.)

- (a) H 'nin 2-evrensel olması durumunda, evrensel olacağını gösterin.

Çözüm: H , 2-evrensel ise, o halde, x ve y farklı anahtarlarının her çifti için ve tüm $i \in \{0, 1, \dots, m-1\}$ ler için;

$$\Pr_{h \in \mathcal{H}} [\langle h(x), h(y) \rangle = \langle i, i \rangle] = \frac{1}{m^2}$$

x ve y 'nin çarpışması için tam olarak m olası yol vardır; yani $i \in \{0, 1, \dots, m-1\}$ için, $h(x) = h(y) = i$ Böylece;

$$\Pr_{h \in \mathcal{H}} [h(x) = h(y)] = \sum_{i=0}^{m-1} \left(\Pr_{h \in \mathcal{H}} [\langle h(x), h(y) \rangle = \langle i, i \rangle] \right) = \frac{m}{m^2} = \frac{1}{m}.$$

Bu nedenle, tanımı gereği, H evrenseldir.

(b) H' ye evrensel olarak belirli bir aile inşa edin, ancak 2-evrensel olmasın ve cevabınızı gerekçeleyin. Aileyi, anahtar başına bir sütun ve fonksiyon başına bir satır olacak şekilde bir tablo olarak yazın. m , $|H|$, ve $|U|$ 'yu mümkün olduğunca küçük tutmaya çalışın.

İpucu: m , $|H|$, ve $|U|$ ile ilgili örnekler var ve hepsi 4' den küçük.

Çözüm: $m = |H| = |U| = 2$ ile örnek bulabiliriz.

Bir $U = \{x, y\}$ evreninde, aşağıdaki H ailesini düşünün:

	x	y
h_1	0	0
h_2	1	0

Eğer, H' den bir rastgele kıyım fonksiyonu seçersek, iki x ve y anahtarının çarpışma olasılığı, h_1 'in seçilme olasılığı veya $1/m = 1/2$ ' dir. Böylece H bir evrensel kıyım ailesidir.

Öte yandan, 2-evrensel olan bir kıyım ailesinden, rastgele seçilmiş bir h kıyım fonksiyonu için, $\langle h(x), h(y) \rangle$ 'nin tüm olası çiftleri, yani $\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle$ eşit olasılıklı olmak zorundadır. Bu örnekte, $\langle h(x), h(y) \rangle$ asla $\langle 0, 1 \rangle$ ya da $\langle 1, 1 \rangle$ ' e eşit olamaz; dolayısıyla H , 2-evrensel değildir.

(c) Bir rakibin; H kıyım ailesini bildiğini ve kıyımladığımız anahtarları denetlediğini ve çarpışmayı zorlamak istediğini farz edin. Problemin bu kısmında, H' nin evrensel olduğunu varsayın. Şu senaryo gerçekleşiyor: h kıyım fonksiyonunu H' den rastgele seçiyoruz, bunu rakipten saklıyoruz ve sonra rakip bir x anahtarını seçiyor ve $h(x)$ değerini öğreniyor. Şimdi rakip çarpışmayı zorlayabilir mi? Diğer bir deyişle, $h(x) = h(y)$ durumunu $1/m$ ' den daha büyük bir olasılıkla oluşturacak bir $y \neq x$ bulabilir mi?

Eğer bulabilecekse, Şık (b)' deki ile aynı formatta, belirli bir evrensel kıyım ailesi yazın ve rakibin bu senaryoda çarpışmayı nasıl zorlayabileceğini açıklayın. **Eğer bulamayacaksa**, rakibin çarpışmayı, $1/m$ 'den daha büyük bir olasılıkla zorlayamayacağını kanıtlayın.

Çözüm: Önceki örneğimize ek bir anahtar ekleyerek, rakibin çarpışmayı zorlayabildiğine dair bir senaryo oluşturabiliriz.

Bir U evreninde $U = \{x, y, z\}$, aşağıdaki H ailesini düşünün:

	x	y	z
h_1	0	0	1
h_2	1	0	1

H , hala bir evrensel kıyım ailesi: x ile y , $1/2$ olasılıkla çarpışıyor, x ile z , $1/2$ olasılıkla çarpışıyor ve y ile z , $0 < 1/2$ olasılıkla çarpışıyor.

Rakip, h_1 'i mi h_2 'yi mi seçmiş olduğumuzu, bize kıyımlama için x 'i vererek belirleyebilir. Eğer; $h(x) = 0$ ise, h_1 'i seçmişizdir ve bu durumda rakip bize y 'yi verir. Aksi takdirde, eğer $h(x) = 1$ ise, h_2 'yi seçmişizdir ve rakip bize z 'yi verir.

d) Şık (c)'deki soruyu cevaplayın ancak, bu kez H 'nin yalnızca evrensel değil, 2-evrensel olduğunu varsayın.

Çözüm: 2-evrensel bir kıyım ailesiyle rakip, çarpışmayı $1/m$ den daha iyi bir olasılıkla zorlayamaz. Aslında, $h(x)$ 'i bilmek, rakibe başka bir y anahtarı için $h(y)$ ile ilgili bir bilgi vermez.

Bunu resmi olarak koşullu olasılıkları kullanarak kanıtlayabiliriz. $h \in H$ rastgele kıyım fonksiyonunu seçtiğimizi ve sonra da rakibin bir x anahtarını kıyımlamamız için bizi zorladığını ve $h(x) = X$ değerini öğrendiğini varsayalım. Ardından rakip bize çarpışmaya yol açacağını umarak, herhangi bir $y \neq x$ anahtarını veriyor. 2-evrenselliğin tanımı gereği, $x \neq y$ olan herhangi bir x ve y için,

$$\Pr_{h \in \mathcal{H}} [h(y) = h(x) \mid h(x) = X] = \frac{\Pr_{h \in \mathcal{H}} [h(y) = h(x) \text{ and } h(x) = X]}{\Pr_{h \in \mathcal{H}} [h(x) = X]} = \frac{1/m^2}{1/m} = \frac{1}{m}.$$

Bu nedenle, rakibin ilk olarak hangi x 'i seçtiğinin ve hangi $h(x) = X$ değerini öğrendiğinin hiç önemi yoktur; herhangi bir y ile x 'in çarpışmasının olasılığı sadece $1/m$ 'dir.