Problem Seti 3

Okumalar: 11.1-11. 3 kısımlar, Bölüm 11.5.

Hem egzersizler hem de problemler çözülecek, ama sadece problemler teslim edilecektir. Egzersizler ders materyalini hazmettirmek amacıyla hazırlanmıştır. Her ne kadar egzersiz çözümlerini teslim etmeyecek olsanız da, egzersizlerdeki konulardan sorumlu olacaksınız.

Sık sık bir problem için ‘’bir algoritma bulun’’ isteğiyle karşılaşacaksınız. Bu konudaki yanıtınız kısa bir makale şeklinde olmalıdır. Konu paragrafı, çözdüğünüz problem ve sonuçlarınızı özetleyecek şekilde düzenlenmelidir. Makalenizin ana yapısında aşağıdaki bilgiler verilmelidir:

1. Algoritmanın İngilizce açıklaması ve eğer faydalı olacaksa Sözde kodu..
2. Algoritmanızın nasıl çalıştığını gösteren en az bir işlenmiş örnek veya şekil.
3. Algoritmanın doğruluğunun kanıtı (veya göstergesi).
4. Algoritmanın koşma süresinin çözümlemesi.

Amacınız iletişim kurmaktır. Tam not sadece iyi açıklanan doğru yanıtlara verilecektir; net olmayan açıklamalar daha düşük notlandırılacaktır.

Egzersiz 3-1. Kitaptaki 11.1-1 nolu egzersiz (222. Sayfa)
Egzersiz 3-2. Kitaptaki 11.2-3 nolu egzersiz (229. Sayfa)
Egzersiz 3-3. Kitaptaki 11.3-3 nolu egzersiz (236 Sayfa)
Egzersiz 3-4. Kitaptaki 11-2 nolu egzersiz (250. Sayfa)

Problem 3-1. Örüntü Eşleme (Pattern matching)

Müdür Skinner'ın bir problemi var: Skinner, Bart Simpson'ın, son kitap raporunda bazı metinleri aşırdığından 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],..., *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 $1 \le i \le j \le k$ *olduğunda*, *A[i], A[i* + 1],..., *A[j]* ikili dizgisi yerine, *A[i. . j]* simgelemini kullanırız ve buna, *A*' nın ***altdizgisi*** deriz. Bu problemin amacı, *P* ' nin *T*'nin bir altdizgisi olup olmadığını belirlemek, yani bazı *i , j '* ler için $1 \le i \le j \le n$ olduğunda *P = A[i. . j]* olup olmadığına karar vermektir.

Bu probleminde, O(log *n)-bit* tamsayılarını sabit zamanda işleyebileceğinizi farzedin. Örneğin, eğer $x \le n^7$ ve $y \le n^5$ 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 $1 \le i \le k - m+1$ 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 $x \ne y$ ise, $h(x) \ne h(y)$'dir. Kıyım fonksiyonunu *O(m)* sürede hesaplayabileceğinizi farz edin. *P*'nin, *T*'nin altdizgisi olup olmadığını *O(mn)* sürede nasıl belirleyeceğinizi gösterin.

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

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

Farz edelim ki *p,* [2, cn^4] değer kümesindeki tüm asal sayılar içinden, tekdüze ama rastgele olarak seçildi.
Birkaç *i'* yi $1 \le i \le n - m + 1$ arasında sabitleyelim ve $x = T[i..\ (i + m - 1)]$ olsun.
Uygun bir c seçildiğinde, eğer $x \neq P$ ise,

$$\Pr_p \left\{ h_p(x) = h_p(P) \right\} \le \frac{1}{n} .$$

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.

(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.

(d) $1 \le i \le n - m$ 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.

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

Problem 3-2. 2-Evrensel Kıyımlama

H, her *h* Є *H'* nin *U* anahtarlar evrenini {0,1,... ,m — 1} ' ya eşlemlediği bir kıyım fonksiyonları grubu olsun . Eğer her sabit $\langle x, y \rangle$ anahtar-çifti için **x ≠ y** olursa, ve *h, H '*nin içinden birbiçimli ve rastgele seçilirse, $\langle h(x), h(y) \rangle$ çiftinin {0,1,..., *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.

(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 lUI ile ilgili örnekler var ve hepsi 4' den küçük.

(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 ≠ 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.

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