SAKARYA ÜNİVERSİTESİ

Ayrık İşlemsel Yapılar

Hafta 4

Yrd. Doç.Dr. Nilüfer YURTAY

Tamsayılar ,Algoritmalar

4.1 Tamsayılar

N × N kümesinde her (a, b), (c, d) ∈ N × N
için

$$(a, b) \sim (c, d) \iff a + d = b + c$$

şeklinde tanımlanan ~ bağıntısını ele alalım. Bu bağıntının bir denklik bağıntısı olduğu kolaylıkla gösterilebilir. O halde bu bağıntı N × N kümesini denklik sınıflarına ayırır. (a, b) elemanının denklik sınıfını

$\overline{(a,b)}$ ile gösterelim. Örneğin;

$\overline{(3,1)} = \{(x, y) \in N \times N : (x, y) \mid (3, 1)\} = \{(x, y) \in N \times N : x + 1 = y + 3\} = \{(2, 0), (3, 1), (4, 2), \ldots\}$.

$\overline{(0,4)} = \{(x, y) \in N \times N : (x, y) \mid (0, 4)\} = \{(x, y) \in N \times N : x + 4 = y\} = \{(0, 4), (1, 5), (2, 6), \ldots\}$.

(a, b) ∈ N × N olmak üzere (a, b)'nin ~ bağıntısına göre olan (a,b) denklik sınıfına tamsayı denir ve Z ile gösterilir.

Teorem

(a,b) ve (c,d) ∈ N × N olsun. N × N kümesinde

(a,b) ~(c,d) ⇔ a+d=b+c biçimden tanımlanan bağıntı bir denklik bağıntısıdır.

ispat:

N × N kümesinde tanımlanan ~ bağıntısının yansıyan, simetrik ve geçişli olduğunu göstermemiz yeterlidir.

∀(a,b)∈ N × N ⇒ a+b=b+a

⇒(a,b)~(a,b) olduğundan ~ bağıntısı yansıyandır.

∀(a,b),(c,d)∈ N × N olsun.
(a,b)~(c,d) ⇔ a+d= b+c
⇔ b+c= a+d
⇔ c+b= d+a
⇔ (c,d)~(a,b) olduğundan bağıntı simetriktir.

∀(a,b),(c,d),(e,f)∈ N × N olsun.

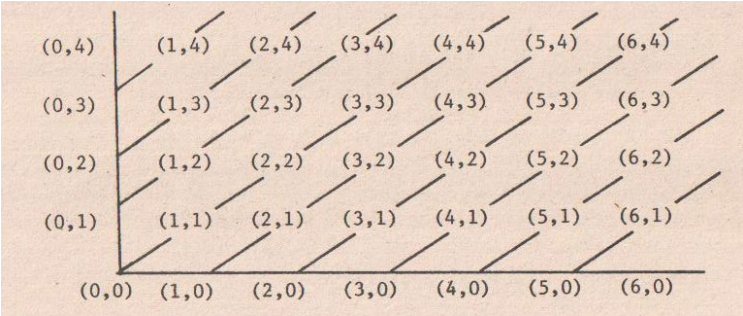
[(a,b)~(c,d) ve (c,d)~(e,f)] ⇔ [a+d=b+c ve c+f=d+e]
⇔ a+d+c+f=b+c+d+e
⇔ a+f=b+e
⇔ (a,b)~ (e,f) olduğundan ~bağıntısı geçişlidir.

(a,b) nin denklik sınıfı

$\overline{(a,b)} = \{(x,y) \mid (x,y) \in NXN, (x,y) \sim (a,b)]$ olarak

tanımlanır.

Aşağıdaki tabloyu inceleyiniz.



Tamsayılar kümesinde tanımlanan

+: ZXZ→Z

$+:(\overline{(a,b)},\overline{(c,d)}) \to \overline{(a+c,b+d)}$ işlemine $\overline{(a,b)}$ ve $\overline{(c,d)}$ tamsayılarının toplamı denir.

$\overline{(a,b)}+\overline{(c,d)}=\overline{(a+c,b+d)}$ olarak gösterilir.

Tamsayılar kümesinde tanımlanan

. : ZXZ→Z

$.:(\overline{(a,b)},\overline{(c,d)}) \to \overline{(ac+bd,ad+bc)}$ işlemine $\overline{(a,b)}$ ve $\overline{(c,d)}$ tamsayılarının çarpımı denir.

$\overline{(a,b)}.\overline{(c,d)}=\overline{(ac+bd,ad+bc)}$ olarak gösterilir.

Teorem: (Z,+,.) cebirsel yapısı değişmeli ve birikimli bir halkadır. (derste ispatlanacaktır!)

$\overline{(a,b)} \in Z$ olsun.

(a) a > b ise $\overline{(a,b)}=\overline{(a-b,0)}=a-b$ tamsayısına pozitif tamsayı denir.

Pozitif tamsayılar kümesi Z^+ ile gösterilir.

(b) a < b ise $\overline{(a,b)}=\overline{(0,b-a)}=-(b-a)$ tamsayısına negatif tamsayı denir.

Negatif tamsayılar kümesi Z^- ile gösterilir.

(c) a = b ise $\overline{(a,b)}$ tamsayısına sıfır denir.

x, y iki tamsayı ise x + (−y) tamsayısına x ile y nin farkı denir ve kısaca x − y ile gösterilir.

Teorem

x,y,z ∈Z olsun.
a) x-y=x+(-1)y
b)z(x-y)=(zx)-(zy) dir.

Ispat:
$x=\overline{(a,b)}$ **ve** $y=\overline{(c,d)}$ **olsun.** $-y=\overline{(d,c)}$ **dir.**

a) $x-y=x+(-y)=\overline{(a,b)}+\overline{(d,c)}=\overline{(a,b)}+\overline{(0,1)}.\overline{(c,d)}=x+(-1)y$

b) $z(x-y)=z[x+(-y)]=zx+z(-y)=zx+z(-1)y=zx+(-1)zy=zx-zy$

Örnek

x + 4 = 3 denkleminin çözümünün x = −1 olduğunu gösterelim. $x=\overline{(a,b)}$, $3=\overline{(3,0)}$ ve $4=\overline{(4,0)}$ alalım.

$$x+4=3\Rightarrow \overline{(a,b)}+\overline{(3,0)}=\overline{(4,0)}$$
$$\Rightarrow \overline{(a+3,b)}=\overline{(4,0)}$$
$$\Rightarrow (a+3,b)\sim(4,0)$$
$$\Rightarrow a+3=b+4\Rightarrow a+3=(b+1)+3\Rightarrow a=b+1$$
$$\Rightarrow x=\overline{(a,b)}=\overline{(a,a+1)}=\overline{(0,1)}=-1$$

a,b∈Z, $a=\overline{(m,n)}$ ve $b=\overline{(u,v)}$ olmak üzere
$a<b\Leftrightarrow m+v<n+u$ olarak tanımlanır.

Teorem
x, y, z, w, t ∈ Z olsun.

(a) x < y ⟺ x + z < y + z **(b) x < y, z < w ⇒ x + z < y + w**

(c) t > 0 ise xt > yt ⟺ x > y **(d) t < 0 ise xt > yt ⟺ x < y**

4.1.1 Tamsayılarda Aritmetik

a,b ∈ Z için ab=x olacak şekilde bir x tamsayısı varsa, a b yi böler denir ve a| b denir.

Aşağıdaki önermeler doğrudur.

1. ∀a∈Z için ±1| a ve ±a| a dır.
2. ∀a∈Z için a| ±1 ise a=±1 dir.
3. ∀a,b∈Z için a| b ve b| a ise a=±b dir.
4. ∀a,b∈Z için a| b ise ±a| ±b dir.
5. ∀a,b,c∈Z için a| b ve b| c ise a| c
6. ∀a,b,c∈Z için a| b ve a| c ise ∀x,y ∈ Z için a| xb+yc dir.
7. ∀a,b,c∈Z için a| b ise a| bc dir.

8. $\forall a,b,c \in Z$ için $c \neq 0$ ise $ca \mid cb \Leftrightarrow a \mid b$ dir.
9. $\forall a,b \in Z$ ve $b \neq 0$ için, $a \mid b$ ise $|a| \leq |b|$ dir. Eğer a has bir bölen ise $1 < |a| < |b|$ dir.

Z de birimin yani 1 tamsayısının bölenlerine aritmetik birimler denir. Bir tamsayının aritmetik birim olması için gerek ve yeter koşul bütün tamsayıları bölmesidir. Z'de aritmetik birimler ± 1 dir.

$a,b \in Z$ ve $b \neq 0$ olsun. $a=bq+r$ ve $0 \leq r < |b|$ olacak şekilde bir q tamsayısı ve r doğal sayısı bulunabiliyorsa a, b'ye kalanlı olarak bölünüyor denir. a'ya bölünen, b'ye bölen, q'ya bölüm, r'ye kalan denir.

Herhangi bir a tamsayısı için a=0 ise, sıfırdan farklı her tamsayı a'nın bir bölenidir. $a \neq 0$ ise -1,+1,a,-a sayılarından biri a'nın bir bölenidir. A'nın bundan başka bölenleri de bulunabilir. Bir a tamsayısının bölenleri {B(a)} ile gösterilir. Örneğin

{B(8)}={-8,-4,-2,-1,1,2,4,8} dir.

Sıfırdan farklı iki a ve b tamasyısının her ikisini de bölen x tamsayısına bu sayıların ortak böleni denir. {OB(a,b)} ile gösterilir.

$\{OB(a,b)\}=\{B(a) \cap B(b)\}$

{B(6)}={-6,-3,-2,-1,1,2,3,6}
{B(8)}={-8,-4,-2,-1,1,2,4,8}
$\{OB(6,8)\}=\{B(6) \cap B(8)\}=\{-2,-1,1,2\}$

Teorem
$a,b \in Z$ ve $b \neq 0$ olsun.
$b \mid a \Rightarrow \{OB(a,b)\}=\{B(b)\}$ dir.

Teorem
$a,b \in Z$ ve $b \neq 0$ olsun. $a=bq+r$ ve $0 \leq r < |b|$ ise
$\{OB(a,b)\}= \{OB(b,r)\}$

En az biri sıfırdan farklı 2 tamsayı a ve b olsun. a ve b nin ortak bölenlerinin kümesinin en büyük elemanına OBEB denir.

Örneğin OBEB(36,28)=4 dür.

Teorem(Öklid Algoritması)

En az biri sıfırdan farklı 2 tamsayı a ve b olsun. OBEB(a,b)=r_n olacak biçimde bir ve yalnız bir tane r_n tamsayısı vardır. Bu sayı a ve b tamsayılarının lineer toplamı olarak yazılabilir. Yani, en az bir m,n tamsayıları için,

OBEB(a,b)= r_n=ma+nb dir.

Örneğin -118 ve 26 sayılarının OBEB'i 2 dir(gösteriniz!).

Teorem

En az biri sıfırdan farklı 2 tamsayı a ve b olsun. OBEB(a,b)= OBEB(b,a) dır.

Teorem

En az biri sıfırdan farklı 2 tamsayı a ve b olsun. $m \in Z^+$ ise OBEB(ma,mb)=m[OBEB(a,b)] dir.

En az biri sıfırdan farklı 2 tamsayı a ve b olsun. Bu sayıların ortak bölenlerinin en büyüğü 1 ise bu sayılara aralarında asal sayılar denir. (a,b)=1 ile gösterilir.

$(a,b)=1 \Leftrightarrow$ OBEB(a,b)=1

Örneğin OBEB(9,16)=1$\Leftrightarrow$(9,16)=1 dir.

Teorem

En az biri sıfırdan farklı 2 tamsayı a ve b olsun. a ve b nin aralarında asal olması için

ma+nb=1 olacak şekilde m,n tamsayılarının bulunması gerek ve yeter koşuldur.

Teorem

a,b,c tamsayılar olsun. (a,c)=1 ve (b,c)=1 ise (ab,c)=1 dir.

Teorem

a,b,c tamsayılar olsun. a| (bc) ve (a,b)=1 ise a| c dir.

Teorem

a,b,n tamsayılar olsun. a| n, b| n ve (a,b)=1 ise (ab)| n dir.

Sıfırdan farklı a tamsayısının bütün katlarının kümesi {K(a)} ile gösterilir. Her ikisi de sıfırdan farklı olan a ve b tamsayılarından her ikisinin katı olan bir tamsayıya bu sayıların ortak bir katı denir. {OK(a,b)} ile gösterilir.

$\{OK(a,b)\}=\{K(a) \cap K(b)\}$ dir.

Örneğin -2 ve 3 sayıları için $\{OK(-2,3)\}=\{K(6)\}$ dır.

$\{OK(a,b)\} \neq 0$ dır.

Teorem

$a,b \in Z-\{0\}$ olsun. OBEB(a,b)=d ise

{OK(a,b)}={K((ab):d]} dir.

Sıfırdan farklı a ve b tamsayılarının pozitif ortak katlarının en küçüğüne, a ve b nin ortak katlarının en küçüğü denir ve OKEK(a,b) ile gösterilir.

OKEK(a,b)= OKEK(a,-b)= OKEK(-a,b)= OKEK(-a,-b) dir.

Teorem

Sıfırdan farklı a ve b tamsayıları için OKEK(a,b)=k ise

{OK(a,b)}={K(k)} dır.

Teorem

Pozitif bir k tamsayısının, a ve b tamsayılarının ortak katlarının en küçüğü olması için

OKEK(a,b)=k⇔((k:a),(k:b))=1 olmalıdır.

Pozitif bir n tamsayısını ele alalım. a tamsayısının n moduna göre b tamsayına denk olabilmesi için, $a \equiv b \pmod{n}$, n tamsayısının (a-b) tamsayısını bölmesi gerekmektedir.

$a \equiv b \pmod{n} \Leftrightarrow n \mid (a-b)$ dir.

Teorem

Sabit bir n için tanımlanan bir benzeşim tamsayılar kümesi üzerinde bir eşdeğer bağıntıdır, öyleki

a) $a \equiv a \pmod{n}$ her a tamsayısı için doğrudur.

b) Eğer $a \equiv b \pmod{n}$ ise, $b \equiv a \pmod{n}$ benzeşimi a ve b tamsayıları için sağlanır.

c) Eğer $a \equiv b \pmod{n}$ ve $b \equiv c \pmod{n}$ ise $a \equiv c \pmod{n}$ yazılabilir.

Teorem

$x,y,z \in Z$ ve $m \in Z^+$ olsun.

$x \equiv y \pmod{m} \Leftrightarrow$ r=s dir. (r ve s sırasıyla x ve y nin m ile bölünmesi sonucunda kalanlardır)

Teorem

$x,y,z,w,u,v \in Z$ ve $m \in Z^+$ olsun.

$x \equiv y \pmod{m}$ ve $z \equiv w \pmod{m}$ ise aşağıda önermeler doğrudur:

$x+u \equiv y+u \pmod{m}$

$xu \equiv yu \pmod{m}$

$x+z \equiv y+w \pmod{m}$

$x-z \equiv y-w \pmod{m}$

$xz \equiv yw \pmod{m}$

ux+vz≡ uy+vw (mod m)

4.2 Knapsack Problemi (Sırt Çantası)

Bir uzay mekiği bir uzay istasyonuna gönderilecektir. Bilim adamlarınca tasarlanan denemeler için 1400 Kg'lık bir yükleme sınırı vardır. Araştırmacılar deneyimlerine göre başvurmuşlar ve her deney için de yanlarına almaları gereken cihazların ağırlıklarını belirlemişlerdir. Daha sonra başvurular değerlendirilmiş ve her bir deneyin önemine göre 1 den 100 e kadar puan verilmiştir. Buna göre her bir deneyin gerektirdiği cihaz ağırlıkları ve önem puanları şöyledir;

Deney No	Ağırlık (Kg)	Puan
1	72	6
2	528	10
3	376	7
4	406	9
5	208	9
6	14	7
7	184	3
8	130	9
9	50	4
10	340	7
11	160	8
12	44	5

Deneylerin seçiminde, toplam puanını mümkün olduğunca fazla olması, ancak toplam ağırlığın 1400 kg geçmemesi istenmektedir. Bunun nasıl gerçekleşeceği çok açık değildir. Örneğin hemen liste başından başlayarak seçsek 1,2,3 ve 4 toplam 1382 olacak. 5. deney 208 kg olduğundan alınamayacak, 6. deney 14 kg olduğu için alınacaktır. Bu şekilde bir seçim yaparsak toplam puan ;

6+10+7+9+7 = 39 olmaktadır.

Şimdi soru bundan daha iyi bir çözüm olup olamadığıdır. Bir yol listeyi önem puanlarına göre sıralayıp, seçimi buna göre yapmak olabilir. Bir ihtimal bu öncekine göre çok daha iyi bir seçim olabilir. Sıralamada aynı puanda olan deneyden daha hafif olanı önce alırsak , sonuç liste ;

Deney No	Puan	Ağırlık	Alınsın mı ?	Toplam Ağırlık
2	10	528	Evet	528
8	9	130	Evet	658
5	9	208	Evet	866
4	9	406	Evet	1272
11	8	160	Hayır	1272
6	7	14	Evet	1286
10	7	340	Hayır	1286
3	7	376	Hayır	1286
1	6	72	Evet	1358
12	5	44	Hayır	1358
9	4	50	Hayır	1358
7	3	184	Hayır	1358

olacak ve toplam puan 10+9+9+9+7+6 =50 olacaktır. Bu bir öncekinden çok daha iyi bir çözüm oldu. Bir başka fikir deneyleri ağırlıklarına göre artan sırada listeleyip seçimi buna göre yapma olabilir. Bu durumda 6,12,9,1,8,11,7,5 ve 10. deneyler seçilecek ve toplam puan da 58 olacaktır. Fakat bunların hiçbiri optimum seçim değildir.

Problemle biraz daha uğraşırsak deneyleri ekleyip çıkartarak 58 puandan daha iyi bir sonuç elde edebiliriz. Fakat yine de bulduğumuz sonucun en iyisi olduğunu söyleyemeyiz. Görüldüğü gibi bu da bir başka optimizasyon problemidir. Bir önceki eşleme problemine benzer olarak , tüm olasılıkları deneme metoduna dönelim. Deneyleriniz 1 den 12 ye kadar kodlandığına göre problemi çok daha kompakt biçimde ele alabiliriz. Burada bize gerekli olan küme teorisi olacaktır. Küme teorisi hakkında ön bilgimiz olduğunu varsayarsak tüm deneyleri içeren kümeyi U={1,2,3,4,5,6,7,8,9,10,11,12} ile gösterebiliriz. Bu kümeden örneğin 1,2,3,4 ve 6 deneylerini seçmemiz bir T alt kümesi oluşturmak demektir. T={1,2,3,4,6} T kümesi , problemin çözümünde ele aldığımız ilk seçimdir. U'nun bazı alt kümeleri , toplam ağırlık 1400 kg'yi geçtiği için kabul edilmeyecektir. Örneğin {2,3,4,10} alt kümelerinin toplam ağırlığı 1650 kg'dir. O halde basit olarak U kümesinin tüm alt kümeleri için toplam ağırlığı hesaplayabiliriz. 1400 kg'yi geçmeyenlerin toplam puanlarını hesaplayıp , sonuçta hangi alt kümenin (ya da alt kümelerinin) max puanı olduğunu buluruz. Buna göre iki soru akla gelecektir ;

1. Kaç tane at küme vardır ?
2. Herhangi birini unutmadan bu alt kümeleri nasıl listeleyebiliriz ?

n elemanlı bir kümenin 2^n adet alt kümesi vardır. U kümesinin 12 elemanı olduğuna göre 2^{12} = 4096 dır. Görüldüğü gibi alt küme sayısı ancak bilgisayarla hesaplamaya uygundur. Eşleme probleminin aksine bu problemin çözümü için bilinen etkin bir yol yoktur

4.3 Algoritma Karmaşıklığı

Sırt çantası örneğinde gördük ki 12 deney için yaklaşık 4096 alt küme sınanmaktadır. Eğer deney sayısı 20 olsaydı , bu rakam 1000000 olacaktı. 250 kez daha fazla. Burada problem, n deney sayısı ile ölçülebilir. Her tür problem için çözümün dayandığı bilgi miktarına ölçü olabilecek bir n sayısı bulunabilir. Problemin çözümü için geliştirilecek algoritmalarda n aynı büyüklüğü göstermek üzere , buna bağlı olarak karşılaştırılabilir.

Ayrıca , problemin çözümünde yapılacak olan hesaplama yükünü de ölçmek isteriz.Muhakkak ki bu da n sayısına bağlıdır. Bir algoritmada n büyüdükçe bu yükün hızla artmaması istenilebilir. Bir algoritmik çözümün büyüklüğünü ölçmek için bir birime ihtiyacımız vardır. Uzay mekiği probleminde n deney için 2^n alt küme sınanmakta idi. Alt küme sınanmasından kasıt , küme elemanlarının ağırlıklarının toplanması, 1400 kg yi aşıp aşmadığını sınanması , aşmıyorsa toplam puanın hesaplanması ve bir önceki en iyi çözümle karşılaştırılmasıdır. Bir alt küme için gereken işlem yükü aynı zamanda o kümedeki eleman sayısına da bağlıdır.

İlk olarak algoritmanın değerlendirilmesinde kullanılan alışılagelmiş yöntemi ele alalım. Bu yöntemde toplama , çıkarma , çarpma , bölme ve karşılaştırma gibi elementer işlemlerin toplam sayısı hesaplanır. Örneğin k adet

a_1, a_2, a_3..... a_k sayısını toplamak istersek k-1 adet toplama yaparız ; $a_1+ a_2$, $(a_1+ a_2)+ a_3$ .....$(a_1+$ .....$+a_{k-1})+a_k$

Elemanter işlemlerin toplam sayısına algoritmanın karmaşıklığı (complexity of algorithm) adını veriyoruz. Bu yolla karmaşıklık ölçmenin iki dezavantajı vardır:

1. Bu yöntem esas olarak algoritmanın gerçeklenmesi için geçecek toplam süreyi hesaplamaya yöneliktir. Burada her elamanter işlemin aynı sürede yapıldığı varsayılmaktadır. Ancak, bilgisayarlar bellekleri ile sınırlıdırlar. Algoritmanın bellek gereksinimi bilgisayarda mevcut bellekten fazla olabilir. Ya da daha yavaş ek bellek kullanımı gerekebilir ve işlem yavaşlayabilir. Saklama olayı gözönüne alınması gereken bir değerdir.

2. Elementer işlemler aynı sürede uygulanmazlar .Örneğin bölme işlemi, toplamadan uzundur.

Ayrıca elementer işlemin süresi, üzerinde işlem yapılan elemanların büyüklüğüne de bağlıdır. Büyük sayılar çok daha uzun süre olurlar.Toplu bir değişkene değer atamanın da bir süre alması gibi işlemler de bizim hesaplamamıza katılmamaktadır. Bu eleştirilere rağmen, algoritmanın karmaşıklığının ölçülmesinde önerilen yöntem kullanılabilir. Özel bir bilgisayarın iç işlemleri ile uğraşmamak ve basitlik için bu yöntemi kullanacağız.

Bazı algoritmaları ve karmaşıklıklarını ele alacağız. İlk olarak m bir reel sayı ve n bir tam sayı olmak üzere m^n in hesaplanmasına bakalım. $X^2=X*X$, $X^3=(X^2)*X...,X^n=X^{n-1}*X$ olduğuna göre toplam n-1 çarpma gerekecektir.Bu işlem için bir algoritma aşağıdaki gibidir.

Algoritma

Verilen bir m reel sayısı ve pozitif n sayısı için $P=X^n$ 'i hesaplayan algoritma

Adım1 (KOŞULLAMA)

P=X ve k=1

Adım 2 (SONRAKİ KUVVET)

While k< n

(a) P ← P.X

(b) k←k+1

EndWhile

Adım 3: ($P=X^n$ olur)

Print P.

Dikkat edilirse 2.adımda n-1 çarpma n-1 toplama ve k adet karşılaştırma (k=n olduğunda 2.adımdan çıkılıyor) yapılıyor.Buna göre X^n in hesaplanması için toplam (n-1)+(n-1)+n=3n-2 adet elemanter işlem yapılmaktadır.

Genellikle , bu işlem sayısında tam bir değer değil de bir mertebe söylememiz bizim için yeterlidir. Bu örnek için "işlem sayısı 3n civarındadır". Hatta “işlem sayısı n in bir sabitle çarpımı kadardır" bile diyebiliriz. Çoğunlukla n in büyümesi durumunda işlem sayısının hızla büyüyüp büyümediği bizi daha çok ilgilendirir.

Şimdi de n.dereceden polinomun hesaplanmasını ele alalım

$$P(X)=a_n x^n + a_{n-1}x^{n-1} + \ldots\ldots + a_1 x + a_a$$

$a_n \neq 0$ ve a_i (i=0,1,.....n) sabit

Algoritma

$P(X)=a_n x^n + \ldots\ldots\ldots + a_0$, n pozitif tam sayı ,x,$a_0$,..... a_n reel sayılar

Adım 1: (Koşullama)

$S= a_0$, k=1

Adım 2 : (Bir sonraki terimi ekle)

While $k \leq n$

(a) $S \leftarrow S+a_k x^k$

*(b) **k** $\leftarrow k+1$*

EndWhile

Adım 3: Print s

2.adımda $k \leq n$ kontrolünü k=1,2.....n+1 için n+1 kere yapıyoruz .Herhangi bir $k \leq n$ için 1 karşılaştırma, 2 toplama ,1 çarpma ve (3k-2) işlem x^k hesabı için yapıyoruz. O halde 3k-2+4=3k+2 işlem yapılmaktadır. k=1,2,...n için $5+8+11+\ldots\ldots+3n+2=(3n^2+7n)/2$ işlem yapılıyor. k=n+1için yapılan karşılaştırmayı da eklersek $((3n^2+7n)+1)/2$ işlem yapılmaktadır . Görüldüğü gibi algoritmanın karmaşıklılığı bir polinomdur , yani $1.5n^2+3.5n+0.5$ dir. Burada polinomun derecesi karmaşıklıkta bizim için çok dalış önemlidir. Bu örnekte n^2 karmaşıklılık etkin olacaktır. Genellikle karmaşıklılığı n^k ʹnın bir polinomu olan Algoritma n^k karmaşıklılığında kabul edilir.

Şimdi polinom hesaplama için çok daha etkin bir algoritmayı ele alalım. (Horner Yöntemi)

$$a_3x^3 + a_2x^2 + a_1 x + a_0 = x(a_3 x^2 + a_2 x + a_1) + a_0 = x(x(a_3 x + a_2) + a_1) + a_0 = x(x(a_3 (x) + a_2) + a_1) + a_0$$

Algoritma

Adım 1 *$S= a_n$ ve k=1*

Adım 2 *While $k \leq n$*

(a) $S \leftarrow XS + a_{n-k}$

(b) $k \leftarrow k+1$

EndWhile

Adım 3 *Print s.*

Adım 2 de her $k \leq n$ için 1 karşılaştırma, 1çarpma, 2 toplama, 1 çıkarma yapılmaktadır. n için 5n+1 işlem yapılmaktadır. Önceki algoritma ile karşılaştırırsak, örneğin n=10 için 51 işleme karşı 186 işlem olacaktır. n'in büyük değeri için fark çok daha artacaktır. Horner yönteminin karmaşıklılığı n olduğu için ilkine göre çok daha üstün bir algoritmadır.

Şimdiye kadar ele aldığımız algoritmalar, karmaşıklılığını tam olarak hesaplayabildiğimiz için yeterince basit algoritmalardır. Ancak, çoğunlukla tam işlem sayısı ,sadece n değerine bağlı değildir. Örneğin n adet sayıyı sıralama algoritması sayıların başlangıçtaki durumuna bağlı olarak daha az yada çok adım gerektirebilir.Böyle durumlarda işlem sayısını üstün halde hesaplarız ve gerçek işlem sayısının buna eşit yada daha az olduğunu söyleriz.

Alt Küme Üreten Algoritma

Uzay mekiği problemimizde gereken alt küme hesaplama için bir algoritma geliştirelim. n elemanlı ve elemanları $x_1,x_2,.....,x_n$ olan bir S kümesi ele alalım. Bu kümeyi 0 ve 1'lerden oluşmuş bir küme şeklinde modellersek,alt kümeleri de aynı biçimde üretebiliriz.S kümesindeki 0 elemanı k.sırada ise $x_k \notin S$ ve 1. sıradaki 1 değerinde $x_J \in S$ anlamındadır.Örneğin n=3 ise 2^3 =8 alt küme 3 elemanlı ikili katar şöyle temsil edilecektir. S:{x1,x2,x3}

000 {∅}

001 $\{x_3\}$

010 $\{x_2\}$

011 $\{x_2,x_3\}$

100 $\{x_1\}$

101 $\{x_1,x_3\}$

110 $\{x_1,x_2\}$

111 $\{x_1,x_2,x_3\}$

Bir sonraki alt katar algoritması

Verilen n pozitif tam sayısı için 0 ve 1 den oluşan $a_1.a_2 \ldots\ldots a_n$ katarını bir sonraki alt kümeye karşılıklı duran katarı hesaplayan algoritma.

Adım 1 (koşullama)

k=n

Adım 2 (en sağdaki sıfırı bul)

While k ≥1 and a_k =1

k ←k - 1

End While

Adım 3 (Sıfır var ise ,sonraki katarı oluştur.)

If k ≥ 1

Adım 3.1 (En sağdaki sıfırı bir ile değiştir.)

a_k ←1

Adım 3.2 (Arkasındaki birleri sıfır yap)

For j = k+1 To n

a_j = 0

End For

Adım 3.2 (Çıkış)

Print $a_1. a_2 \ldots\ldots a_n$

Other wise

Adım 3.3 (Alt küme yok)

Print ' Katarın tüm elemanları birdir.'

End If

Şimdi de bu algoritmayı sırt çantası problemine nasıl uygulayacağımızı düşünelim. W_i , i . deneyin ağırlığı olsun. Bu durumda a_1. a_2 .... a_{12} katarı ile temsil ettiğimiz alt küme için $a_1W_1+ a_2 W_2+....+ a_{12}W_{12} \leq$ 1400 karşılaştırmasını yapacağız. Burada a_i =1 i. deney var, a_i =0 i. deney yok anlamında yorumlanacaktır. Bu hesap için karşılaştırma ve indeks hesapları hariç n çarpma ve n-1 toplama olamak üzere 2n-1 işlem vardır. n deney için 2^n alt küme vardır. Buna göre bu yöntemin karmaşıklığı C bir sabit olmak üzere Cn 2^n olacaktır. Aşağıdaki tabloda saniyede 10^6 işlem yapabilen bir bilgisayarda n çeşitli değerleri için topla işlem süresi ve bir karşılaştırma için 1000 n^2 'nin alacağı süre verilmiştir.

n	10	20	30	40	50
n. 2^n	0.001s	21 s	9 s	1.4 yıl	1785 yıl
$1000n^2$	0.1 s	0.4 s	0.9 s	1.6 s	2.5 s

Genel olarak , bir algoritmada eğer n sayısı birden büyük bir sayının kuvveti olarak geliyorsa, n 'in bir polinoma göre çok daha hızlı artacaktır. n! ise çok daha hızlı artacaktır. Aşağıdaki tablo çeşitli n değerleri için farklı n fonksiyonlarında yine saniyede 10^6 işlem yapan bilgisayarın toplam sürelerini vermektedir.

n	10	30	60
$\log_2 n$	0.0000033 s	0.0000049 s	0.0000059 s
$n^{1/2}$	0.0000032 s	0.0000055 s	0.0000077 s
n	0.00001s	0.00003 s	0.00006 s
n^2	0.0001 s	0.0009 s	0.0036 s
n^2 +10n	0.0002 s	0.0012 s	0.0042 s
n^{10}	2.8 saat	19 yıl	19174 yıl
2^n	0.001 s	18 ay	36559 yıl
n!	3.6 s	8.4 10^{18} yıl	2.610^{68} yıl

Ödev

1. $\forall a,b,c \in Z$ için $a \mid b$ ve $a \mid c$ ise $\forall x,y \in Z$ için $a \mid xb+yc$ dir. İspatlayınız.
2. $x,y \in Z$ ve $m,n \in Z^+$ olsun.
   $x \equiv y \pmod m \Rightarrow x^n \equiv y^n \pmod m$ olduğunu ispatlayınız.
3. Öklid algoritmasını yazınız.

Kaynaklar

F.Selçuk,N.Yurtay,N.Yumuşak,Ayrık İşlemsel Yapılar, Sakarya Kitabevi,2005.
İ.Kara, Olasılık, Bilim Teknik Yayınevi, Eskişehir, 2000.

"Soyut Matematik", S.Aktaş,H.Hacısalihoğlu,Z.Özel,A.Sabuncuoğlu, Gazi Ünv.Yayınları,1984,Ankara.

"Applied Combinatorics", Alan Tucker, John Wiley&Sons Inc, 1994.

"Applications of Discrete Mathematics", John G. Michaels, Kenneth H. Rosen, McGraw-Hill International Edition, 1991.

"Discrete Mathematics", Paul F. Dierker and William L.Voxman, Harcourt Brace Jovanovich International Edition, 1986.

"Discrete Mathematic and Its Applications", Kenneth H. Rosen, McGraw-Hill International Editions, 5th Edition, 1999.

"Discrete Mathematics", Richard Johnson Baugh, Prentice Hall, Fifth Edition, 2001.

"Discrete Mathematics with Graph Theory" , Edgar G. Goodaire, Michael M. Parmenter, Prentice Hall, 2nd Edition, 2001.

"Discrete Mathematics Using a Computer", Cordelia Hall and John O'Donnell, Springer, 2000.

"Discrete Mathematics with Combinatorics", James A. Anderson, Prentice Hall, 2000.

"Discrete and Combinatorial Mathematics", Ralph P. Grimaldi, Addison-Wesley, 1998.

"Discrete Mathematics", John A. Dossey, Albert D. Otto, Lawrence E. Spence, C. Vanden Eynden, Pearson Addison Wesley; 4th edition 2001.

"Essence of Discrete Mathematics", Neville Dean, Prentice Hall PTR, 1st Edition, 1996.

"Mathematics:A Discrete Introduction", Edvard R. Schneiderman, Brooks Cole; 1st edition, 2000.

"Mathematics for Computer Science", A.Arnold and I.Guessarian, Prentice Hall, 1996.

"Theory and Problems of Discrete Mathematics", Seymour Lipschuts, Marc. L. Lipson, Shaum's Outline Series, McGraw-Hill Book Company, 1997.

"2000 Solved Problems in Discrete Mathematics", Seymour Lipschuts, McGraw- Hill Trade, 1991.