BÖLÜM 5

SAYILAR TEORİSİNE GİRİŞ

Bölenler

- **Tanım 5.1.1:** n ve d tamsayılar ve $d \neq 0$ olsun. Eğer $n=dq$ olacak şekilde bir q tamsayısı varsa d sayısı n sayısını ***böler*** denir.
- Burada q sayısına ***bölüm*** ve d sayısına da ***bölen*** denir.
- Eğer d sayısı n sayısını bölerse, bu $d|n$ şeklinde yazılır. Aksi halde $d \nmid n$ şeklinde gösterilir.

- **Teorem 5.1.2:** m,n ve d tamsayılar olsunlar.
  - Eğer d|m ve d|n ise,
    d|(m+n)
  - Eğer d|m ve d|n ise,
    d|(m-n)
  - Eğer d|m ise, d|mn

- **Tanım 5.1.3:** Tek pozitif bölenleri kendisi ve 1 olan 1 ‘den büyük olan sayılara asal sayı denir. Asal olmayan tamsayılara **birleşik** (composite) sayı denir.
- **Teorem 5.1.4:** 1 ‘den büyük bir n pozitif tamsayısının komposite olması için gerekli ve yeterli koşul n sayısının $2 \leq d \leq n^{1/2}$ olacak şekilde bir d bölenine sahip olmasıdır.

Algoritma 5.1.5

Bu algoritma n>1 sayısının asal sayı olup olmadığını hesaplar. Eğer n sayısı asalsa algoritma geriye 0 döndürür. Eğer n sayısı komposite sayıysa, algoritma geriye $2 \leq d \leq \sqrt{n}$ koşulunu sağlayan d bölenini döndürür.

Girdi: n

Çıktı: d

```
is_prime(n)  {
        for d=2 to ⌊√n⌋
                if (n mod d==0)
                        return d
        return 0
}
```

- **Teorem 5.1.6: (Aritmetiğin Temel Teoremi)**
  1 'den büyük her tamsayı asal çarpanlarının çarpımı olarak yazılabilir. Eğer asal çarpanlar artan sırada yazılırsa, bu çarpım gösterimi bir tekdir.
  Yani, p_k 'lar $p_1 \leq p_2 \leq \ldots \leq p_i$ koşulunu sağlayan asal çarpanlarsa

  $$n = p_1 p_2 \ldots p_i$$

  olur ve p_k' 'lar
  $p'_1 \leq p'_2 \leq \ldots \leq p'_i$ koşulunu sağlayan diğer asal çarpanlarsa ve

  $$n = p'_1 p'_2 \ldots p'_i$$

  ise, bu durumda i=j ve her k=1...i için

  $$p_k = p'_k$$

  olur.

- **Teorem 5.1.7:** Asal sayıların sayısı sonsuzdur.

- **Tanım 5.1.9:** Herikisi birden sıfır olamayan iki tamsayı m ve n olsunlar. Hem m ve hemde n sayısını bölen tamsayılara ***ortak bölen*** denir.
- Bu ortak bölenlerin en büyüğü *gcd(m,n)* ile gösterilir.

- **Teorem 5.1.10:** m ve n iki tamsayı ve m>1,n>1 olsun. Ayrıca bu iki sayının asal çarpımları

$$m = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$$

ve

$$n = p_1^{b_1} p_2^{b_2} \cdots p_k^{b_k}$$

olsunlar. Bu durumda

$$gcd(m,n) = p_1^{\min(a_1,b_1)} p_1^{\min(a_2,b_2)} \cdots p_1^{\min(a_k,b_k)}$$

ile verilir.

- **Tanım 5.1.11:** İki pozitif amsayı m ve n olsunlar. Hem m ve hemde n sayısı ile bölünebilen tamsayıya m ile n tamsayılarının ***ortak katı*** denir.
- m ile n tamsayılarının ortak katlarının en küçüğü *lcm(m,n)* ile gösterilir.

- **Teorem 5.1.12:** m ve n iki tamsayı ve m>1,n>1 olsun. Ayrıca bu iki sayının asal çarpımları

$$m = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$$

ve

$$n = p_1^{b_1} p_2^{b_2} \cdots p_k^{b_k}$$

olsunlar. Bu durumda

$$lcm(m,n) = p_1^{\max(a_1,b_1)} p_1^{\max(a_2,b_2)} \cdots p_1^{\max(a_k,b_k)}$$

ile verilir.

- **Teorem 5.1.13:** Her m ve n pozitif tamsayıları için

  gcd(m,n).lcm(m,n)=m.n

  olur.
- **Teorem 5.1.14:** *m,n* ve *c* tamsayılar olsunlar

  (a) Eğer *c* sayısı *m* ve *n* sayılarının ortak böleni ise, bu durumda $c|(m+n)$ olur.

  (b) Eğer *c* sayısı *m* ve *n* sayılarının ortak böleni ise, bu durumda $c|(m-n)$ olur.

  (c) Eğer $c|m$ ise, bu durumda $c|mn$ olur.

TAMSAYILARIN TEMSİLİ
VE
TAMSAYI ALGORİTMALARI

Algoritma 5.2.1:

Bu algoritma b tabanında verilen $c_n c_{n-1} \ldots c_1 c_0$ tamsayısını ondalık tabana çevirir.

Girdi: c,n,b

Çıktı: dec_val

```
        base_b_to_dec (c,n,b) {
                dec_val=0
                power=1
                for i=0 to n {
                        dec_val =dec_val+c_i*power
                        power = power *b
                }
                return dec_val
        }
```

Algoritma 5.2.2:

Bu algoritma ondalık tabanda verilen bir m tamsayısını b tabanında verilen $c_n c_{n-1} \ldots c_1 c_0$ tamsayısına çevirir.

Girdi: m,b

Çıktı: c, n

```
dec_to_base_b (m,b) {
        n=-1
        while (m>0) {
                n=n+1
                c_n=m mod b
                m=⌊m/b⌋
        }
}
```

Algoritma 5.2.3:

Bu algoritma $b_n b_{n-1} \ldots b_1 b_0$ ve $b'_n b'_{n-1} \ldots b'_1 b'_0$ ikili tamsayılarını toplar ve toplamı $s_{n+1} s_n s_{n-1} \ldots s_1 s_0$ içinde tutar.

Girdi: b, b',n

Çıktı: s

binary_addition(b, b',n,s) {

carry=0

for i=0 **to** n **{**

$s_i = (b_i + b'_i + \text{carry}) \bmod 2$

carry = $\lfloor (b_i + b'_i + \text{carry})/2 \rfloor$

}

s_{n+1}=carry

}

x	n ‘in güncel değeri	n mod 2	sonuç	n sayısı 2 ile bölündüğünde bölüm
a	29	1	a	14
a^2	14	0	değişmedi	7
a^4	7	1	$a.a^4=a^5$	3
a^8	3	1	$a^5.a^8=a^{13}$	1
a^{16}	1	1	$a^{13}.a^{16}=a^{29}$	0

Algoritma 5.2.5:

Bu algoritma tekrarlı kareleme yöntemiyle a^n değerini hesaplar.

Girdi: a,n

Çıktı: a^n

```
exp_via_repeated_squaring(a,n) {
        result=1
        x=a
        while (n>0) {
                if(n mod 2 ==1)
                        result=result * x
                x=x*x
                n=⌊n/2⌋
        }
        return result
}
```

- **Teorem 5.2.6:** Eğer a, b ve z pozitif tamsayılarsa
  $ab \bmod z = [(a \bmod z)(b \bmod z)] \bmod z$
  dir.

Algoritma 5.2.8:

Bu algoritma tekrarlı kareleme yöntemiyle a^n mod z değerini hesaplar.

Girdi: a,n, z

Çıktı: a^n mod z

```
exp_mod_z_via_repeated_squaring(a,n,z) {
        result=1
        x=a mod z
        while (n>0) {
                if(n mod 2 ==1)
                        result=(result * x) mod z
                x=(x*x) mod z
                n=⌊n/2⌋
        }
        return result
}
```

EUCLIDEAN ALGORİTMASI

- **Teorem 5.3.1:** Eğer a is a negatif olmayan bir tamsayı, b is a pozitif tamsayı, ve $r=a$ **mod** b ise, bu durumda

  $$\gcd(a,b)=\gcd(b,r)$$

  olur.

Algoritma 5.3.3:

Bu algoritma negatif olmayan iki *a,b* tamsayısının en büyük ortak bölenini bulur.

Girdi: *a* ve *b* negatif olamayan tamsayılar

Çıktı: *a* ve *b* 'nin en büyük ortak bölen

```
1   gcd(a,b)  {
2       if (a<b)
3           swap(a,b)
4       while(b≠0)  {
5           r=a mod b
6           a=b
7           b=r
8       }
9       return a
10  }
```

a \ b	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	—	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	2	1	2	1	2	1	2	1	2	1	2
3	0	1	2	1	2	3	1	2	3	1	2	3	1	2
4	0	1	1	2	1	2	2	3	1	2	2	3	1	2
5	0	1	2	3	2	1	2	3	4	3	1	2	3	4
6	0	1	1	1	2	2	1	2	2	2	3	3	1	2
7	0	1	2	2	3	3	2	1	2	3	3	4	4	3
8	0	1	1	3	1	4	2	2	1	2	2	4	2	5
9	0	1	2	1	2	3	2	3	2	1	2	3	2	3
10	0	1	1	2	2	1	3	3	2	2	1	2	2	3
11	0	1	2	3	3	2	3	4	4	3	2	1	2	3
12	0	1	1	1	1	3	1	4	2	2	2	2	1	2
13	0	1	2	2	2	4	2	3	5	3	3	3	2	1

a	b	n (= number of modulus operations)
1	0	0
2	1	1
3	2	2
5	3	3
8	5	4
13	8	5

- **Teorem 5.3.4:** a,b, a>b çifti Euclidean algoritmasında girdi değerler olarak verilsinler ve $n \geq 1$ olsun.
  Bu durumda $\{f_n\}$ Fibonacci dizisini göstermek üzere

  $a \geq f_{n+2}$

  ve

  $b \geq f_{n+1}$

  olur.

- **Teorem 5.3.5:** Eğer a,b,a>b değerleri 0 ve m , m≥8 aralığında ise ve bunlar Euclidean algoritmasında girdi değerler iseler, bu durumda en fazla

$$\log_{3/2} \frac{2m}{3}$$

sayıda **mod** işlemine gerek vardır.

- **Teorem 5.3.6:** a ve b herikisi de sıfır olmayan iki negatif olmayan tamsayı ise, bu durumda
  gcd(a,b)=sa+tb
  olacak şekilde s ve t tamsayıları vardır.

Algoritma 5.3.7:

Bu algoritma negatif olamayan *a* ve *b* tamsayılarının en büyük ortak bölenini özyineli algoritma ile bulur.

Girdi: Girdi: 0 'dan büyük ya da eşit bir *n* tamsayısı

Çıktı: *a* ve *b* 'nin en büyük ortak böleni

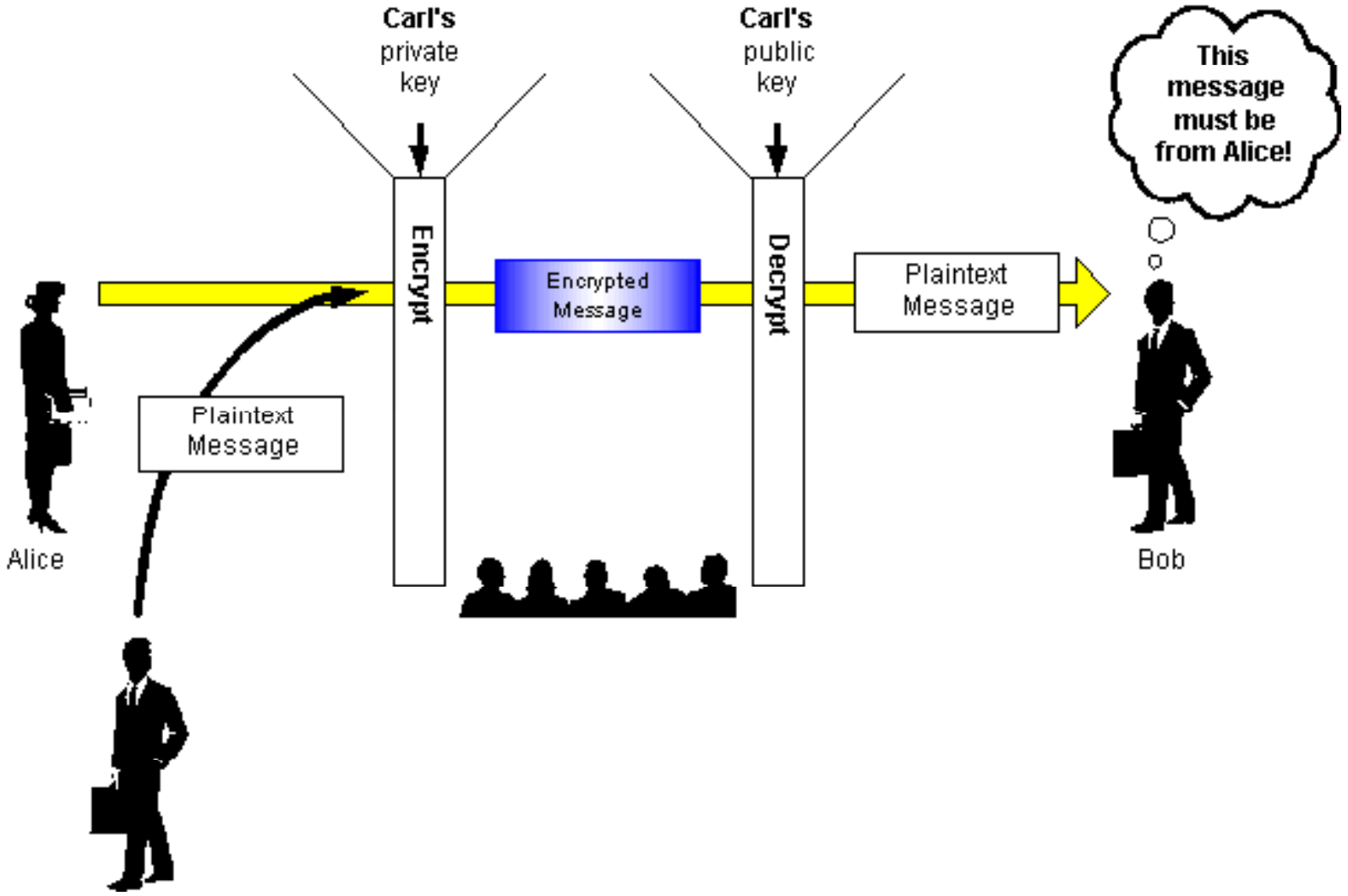
```
gcdr(a,b)  {
        if (a<b)
                swap(a,b)
        if (b==0)
                return a
        r = a mod b
        return gcdr(b,r)
}
```

RSA
Kamusal-Anahtar
Şifreleme Sistemi

- ABCDEFGHIJKLMNOPQRSTUVWXYZ
- EIJFUAXVHWPGSRKOBTQYDMLZNC

SEND MONEY
QARUESKRAN

SKRANEKRELIN
MONEY ON WAY


Carl's
private
key
Carl's
public
key
This
message
must be
from Alice!
Encrypt
Encrypted
Message
Decrypt
Plaintext
Message
Plaintext
Message
Alice
Bob

- Alıcı p ve q gibi iki asal sayı seçer ve z=pq hesabını yapar.
- Sonra alıcı

  $\phi=(p-1)(q-1)$

  sayısını hesaplarlar ve gcd(n, ϕ)=1 olacak şekilde bir n tamsayısı seçer.
- Pratikte n sayısı bir asal sayı olarak seçilir. Bu z, n sayı çifti kamusaldır. Sonra alıcı

  $ns \bmod \phi=1$

  olacak şekilde tek bir s, $o<s<\phi$ sayısını hesaplar. Bu sayı gizli tutulur ve mesajı şifrelemek için kullanılır.
- Gönderici a, $o \leq a \leq z-1$, tamsayısını göndermek isterse z ve n kamusal anahtarlarını kullanarak

  $c=a^n \bmod z$

  hesabını yapar ve c değerini gönderir.
- Mesajın deşifre edilmesi için alıcının

  $c^s \bmod z$

  hesabını yapması lazımdır.