



# **Izbrani algoritmi**

## **Algoritmi teorije števil**

Damjan Strnad

# Osnovni pojmi

- **cela števila** (integers)  $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
- **naravna števila** (naturals)  $\mathbb{N} = \{1, 2, 3, \dots\}$
- $a = k \cdot d$ , kjer  $k \in \mathbb{Z}$ :
  - $d$  je **delitelj** od  $a$
  - $a$  je **večkratnik** od  $d$
  - $d \mid a$  ( $d$  **deli**  $a$ ),  $d \nmid a$  ( $d$  **ne deli**  $a$ )
- vsako celo število deli 0

# Osnovni pojmi

- **trivialna delitelja** od  $a$  sta 1 in  $a$
- **faktorji** so netrivialni delitelji od  $a$ :
  - faktorji od 12 so 2, 3, 4 in 6
- **praštevilo** (prime) je naravno število, ki ima samo trivialna delitelja
  - primeri: 2, 3, 5, 7, 11, 13, ...
  - nasprotje praštevila je **sestavljeno število**
- 0, 1 (**enota**) in negativna števila niso niti praštevila niti sestavljena števila

# Izrek o deljenju

- Za vsako celo število  $a$  in neničelno celo število  $n$  obstajata celi števili  $q = \lfloor a / n \rfloor$  in  $r = a \bmod n$  taki, da velja  $0 \leq r < |n|$  in  $a = qn + r$ .
- $q$  imenujemo **kvocient**,  $r$  pa **ostanek** deljenja
- oznaka  $\bmod$  je kratica za **modulo**
- $n \mid a \Leftrightarrow a \bmod n = 0$

# Kongruenca

- če  $(a \bmod n) = (b \bmod n)$ , to zapišemo kot  $a \equiv b \pmod{n}$  in beremo kot:
  - $a$  je **ekvivalenten**  $b$ , modulo  $n$
  - $a$  in  $b$  sta **kongruentna** po modulu  $n$
- $n|a \Leftrightarrow a \equiv 0 \pmod{n}$
- relacija  $\equiv$  je refleksivna ( $a \equiv a$ ), simetrična ( $a \equiv b \pmod{n} \Leftrightarrow b \equiv a \pmod{n}$ ) in tranzitivna ( $a \equiv b \pmod{n} \wedge b \equiv c \pmod{n} \Rightarrow a \equiv c \pmod{n}$ )

# Lastnosti kongruence

- $a \equiv b \pmod{n} \Rightarrow a+c \equiv b+c \pmod{n}$  za  $\forall c \in \mathbb{Z}$
- $a \equiv b \pmod{n} \Rightarrow a \cdot c \equiv b \cdot c \pmod{n}$  za  $\forall c \in \mathbb{Z}$
- $a \equiv b \pmod{n} \Rightarrow a^k \equiv b^k \pmod{n}$  za  $\forall k \in \mathbb{N}$
- $a \equiv b \pmod{n} \wedge c \equiv d \pmod{n} \Rightarrow$   
 $a+c \equiv b+d \pmod{n} \wedge a \cdot c \equiv b \cdot d \pmod{n}$



# Skupni delitelji

- $d|a \wedge d|b \Rightarrow d$  je **skupni delitelj**  $a$  in  $b$
- $d|a \wedge d|b \Rightarrow d|(ax+by)$  za  $\forall x,y \in \mathbb{Z}$
- **največji skupni delitelj** števil  $a$  in  $b$  označimo z  $\gcd(a,b)$
- $\gcd(0,0) := 0$
- $a \neq 0 \wedge b \neq 0 \Rightarrow 1 \leq \gcd(a,b) \leq \min(|a|, |b|)$
- $d|a \wedge d|b \Rightarrow d|\gcd(a,b)$
- če  $\gcd(a,b)=1 \Rightarrow a$  in  $b$  sta **tuji števili** (coprimes)
- $\gcd(a,p)=1 \wedge \gcd(b,p)=1 \Rightarrow \gcd(ab,p)=1$

# Tuja števila

- izrek o enolični faktorizaciji:
  - Sestavljeno celo število  $a$  lahko zapišemo kot produkt oblike  $a = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$ , kjer so  $p_i$  praštevila (t.i. **prafaktorji**) in  $e_i \in \mathbb{Z}^+$
  - Primer:  $2535 = 3^1 \cdot 5^1 \cdot 13^2$
- uporabno za izračun  $\gcd(a, b)$ , če sta  $a$  in  $b$  že faktorizirana:
 
$$a = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r} \text{ in } b = p_1^{f_1} p_2^{f_2} \dots p_r^{f_r} \Rightarrow$$

$$\gcd(a, b) = p_1^{g_1} p_2^{g_2} \dots p_r^{g_r}, \text{ kjer } g_i = \min(e_i, f_i)$$
- sicer faktorizacija težji problem od gcd



# Evklidov algoritem

- Če sta  $a$  in  $b$  poljubni nenegativni celi števili, potem velja  $\gcd(a, b) = \gcd(b, a \bmod b)$ .
- Algoritem:

```
EUCLID( $a, b$ )  
  if  $b = 0$   
    return  $a$   
  else  
    return  $\text{EUCLID}(b, a \bmod b)$ 
```

# Evklidov algoritem - zgled

- Z Evklidovim algoritmom poiščimo  $\text{gcd}(1519, 469)$ .
- Vhodni parametri metode EUCLID po iteracijah:

iteracija	$a$	$b$
0	1519	469
1		
2		
3		
4		

# Evklidov algoritem - zgled

- Z Evklidovim algoritmom poiščimo  $\text{gcd}(1519, 469)$ .
- Vhodni parametri metode EUCLID po iteracijah:

iteracija	$a$	$b$
0	1519	469
1	469	112
2		
3		
4		

# Evklidov algoritem - zgled

- Z Evklidovim algoritmom poiščimo  $\text{gcd}(1519, 469)$ .
- Vhodni parametri metode EUCLID po iteracijah:

iteracija	$a$	$b$
0	1519	469
1	469	112
2	112	21
3		
4		

# Evklidov algoritem - zgled

- Z Evklidovim algoritmom poiščimo  $\text{gcd}(1519, 469)$ .
- Vhodni parametri metode EUCLID po iteracijah:

iteracija	$a$	$b$
0	1519	469
1	469	112
2	112	21
3	21	7
4		



# Evklidov algoritem - zgled

- Z Evklidovim algoritmom poiščimo  $\text{gcd}(1519, 469)$ .
- Vhodni parametri metode EUCLID po iteracijah:

iteracija	$a$	$b$
0	1519	469
1	469	112
2	112	21
3	21	7
4	7	0

# Evklidov algoritem - zgled

- Z Evklidovim algoritmom poiščimo  $\text{gcd}(1519, 469)$ .
- Vhodni parametri metode EUCLID po iteracijah:

iteracija	$a$	$b$
0	1519	469
1	469	112
2	112	21
3	21	7
4	7	0

$$\text{gcd}(1519, 469) = 7$$

# Razširjeni Evklidov alg.

- poleg gcd poišče tudi celoštevilski koeficienta  $x$  in  $y$ , za katera velja:  
$$d = \gcd(a, b) = ax + by$$
- če je  $\gcd(a, b) = 1$ , potem je  $x$  **multiplikativni inverz** od  $a$  (mod  $b$ ),  $y$  pa multiplikativni inverz od  $b$  (mod  $a$ )
- dve varianti algoritma:
  - ni zahteve  $a > b$
  - iterativna
  - rekurzivna

# Razširjeni Evklidov alg.

- poleg gcd poišče tudi celoštevilska koeficienta  $x$  in  $y$ , za katera velja:  
$$d = \gcd(a, b) = ax + by$$
- če je  $\gcd(a, b) = 1$ , potem je  $x$  **multiplikativni inverz** od  $a \pmod{b}$ ,  $y$  pa multiplikativni inverz od  $b \pmod{a}$
- dve varianti algoritma:
  - ni zahteve  $a > b$
  - **iterativna**
  - **rekurzivna**

```
EXTENDED_EUCLID(a, b)
  x ← 0, lastx ← 1, y ← 1, lasty ← 0
  while b ≠ 0
    q ← a div b
    temp ← b
    b ← a mod b
    a ← temp
    temp ← x
    x ← lastx - q · x
    lastx ← temp
    temp ← y
    y ← lasty - q · y
    lasty ← temp
  return (a, lastx, lasty)
```

# Razširjeni Evklidov alg.

- poleg gcd poišče tudi celoštevilska koeficienta  $x$  in  $y$ , za katera velja:  

$$d = \gcd(a, b) = ax + by$$
- če je  $\gcd(a, b) = 1$ , potem je  $x$  **multiplikativni inverz** od  $a \pmod{b}$ ,  $y$  pa multiplikativni inverz od  $b \pmod{a}$
- dve varianti algoritma:
  - ni zahteve  $a > b$
  - iterativna
  - **rekurzivna**

```

EXTENDED_EUCLID(a, b)
  if b=0
    return (a, 1, 0)
  (d', x', y') ← EXTENDED_EUCLID(b, a mod b)
  (d, x, y) ← (d', y', x' - (a div b) · y')
  return (d, x, y)

```



# Razširjeni Evklidov alg. - zgled

- $\gcd(13, 27)$ : ( $x=-2$ ,  $y=1$ , multiplikativni inverz od 13 (mod 27) je prvo pozitivno število oblike  $x+k\cdot b$ , kar je 25)

korak	kvocient	ostanek	substitucija	kombiniran izraz
1		13		$13=13\cdot 1+27\cdot 0$
2		27		$27=13\cdot 0+27\cdot 1$
3	0	$13=13-27\cdot 0$	$13=(13\cdot 1+27\cdot 0)-(13\cdot 0+27\cdot 1)\cdot 0$	$13=13\cdot 1+27\cdot 0$
4	2	$1=27-13\cdot 2$	$1=(13\cdot 0+27\cdot 1)-(13\cdot 1+27\cdot 0)\cdot 2$	$1=13\cdot (-2)+27\cdot 1$
5	13	0		

# Razširjeni Evklidov alg. - zgled

- $\gcd(13, 27)$ : ( $x=-2$ ,  $y=1$ , multiplikativni inverz od 13 (mod 27) je prvo pozitivno število oblike  $x+k \cdot b$ , kar je 25)

korak	kvocient	ostanek	substitucija	kombiniran izraz
1		13		$13=13 \cdot 1+27 \cdot 0$
2		27		$27=13 \cdot 0+27 \cdot 1$
3	0	$13=13-27 \cdot 0$	$13=(13 \cdot 1+27 \cdot 0)-(13 \cdot 0+27 \cdot 1) \cdot 0$	$13=13 \cdot 1+27 \cdot 0$
4	2	$1=27-13 \cdot 2$	$1=(13 \cdot 0+27 \cdot 1)-(13 \cdot 1+27 \cdot 0) \cdot 2$	$1=13 \cdot (-2)+27 \cdot 1$
5	13	0		$d$ $x$ $y$

# Razširjeni Evklidov alg. - zgled

- $\gcd(91, 70)$ : ( $x=-3$ ,  $y=4$ , multiplikativni inverz ne obstaja, ker  $\gcd(91, 70) \neq 1$ )

korak	kvocient	ostanek	substitucija	kombiniran izraz
1		91		$91 = 91 \cdot 1 + 70 \cdot 0$
2		70		$70 = 91 \cdot 0 + 70 \cdot 1$
3	1	$21 = 91 - 70 \cdot 1$	$21 = (91 \cdot 1 + 70 \cdot 0) - (91 \cdot 0 + 70 \cdot 1) \cdot 1$	$21 = 91 \cdot 1 + 70 \cdot (-1)$
4	3	$7 = 70 - 21 \cdot 3$	$7 = (91 \cdot 0 + 70 \cdot 1) - (91 \cdot 1 + 70 \cdot (-1)) \cdot 3$	$7 = 91 \cdot (-3) + 70 \cdot 4$
5	3	0		

# Modulske linearne enačbe

- enačbe oblike  $ax \equiv b \pmod{n}$ , kjer so  $a$ ,  $b$  in  $n$  podani
- pomembne za izračun kriptografskih ključev
- enačba je rešljiva natanko tedaj, ko  $\gcd(a,n) \mid b$
- rešitve enačbe so oblike  $x_i = x_0 + i \cdot (n/d)$ , kjer  $d = \gcd(a,n)$  in  $i = 0, 1, 2, \dots, d-1$
- za izračun gcd uporabimo razširjeni Evklidov algoritem
- časovna zahtevnost izračuna je  $O(\log(n) + \gcd(a,n))$

```

MOD_LIN_EQ_SOLVER(a, b, n)
  (d, x', y') ← EXTENDED_EUCLID(a, n)
  if d | b
    x0 ← (x' · (b div d)) mod n
    for i ← 0 to d-1
      xi ← (x0 + i · (n div d)) mod n
  else ni rešitve
  
```

# Modulske lin. enačbe - zgled

- pozitiven primer:  $98x \equiv 7 \pmod{35}$ ,  $a=98$ ,  $b=7$ ,  $n=35$ 
  - $d=7$ ,  $d|b$  ✓

korak	kvocient	ostanek	substitucija	kombiniran izraz
1		98		$98=98 \cdot 1+35 \cdot 0$
2		35		$35=98 \cdot 0+35 \cdot 1$
3	2	$28=98-35 \cdot 2$	$28=(98 \cdot 1+35 \cdot 0)-(98 \cdot 0+35 \cdot 1) \cdot 2$	$28=98 \cdot 1+35 \cdot (-2)$
4	1	$7=35-28 \cdot 1$	$7=(98 \cdot 0+35 \cdot 1)-(98 \cdot 1+35 \cdot (-2)) \cdot 1$	$7=98 \cdot (-1)+35 \cdot 3$
5	4	0		

- $x_0 = (-1 \cdot (7 \operatorname{div} 7)) \bmod 35 = (-1) \bmod 35 = (-1+1 \cdot 35) \bmod 35 = 34$  (ostanek mora biti pozitiven!)
- $x_1 = (x_0 + 1 \cdot (35/7)) \bmod 35 = (34+5) \bmod 35 = 4$
- $x_2 = 9, x_3 = 14, x_4 = 19, x_5 = 24, x_6 = 29$



# Modulske lin. enačbe - zgled

- negativen primer:  $12x \equiv 4 \pmod{45}$ ,  $a=12$ ,  $b=4$ ,  $n=45$   
 –  $d=3$ ,  $d \nmid b$

korak	kvocient	ostanek	substitucija	kombiniran izraz
1		12		$12=12 \cdot 1+45 \cdot 0$
2		45		$45=12 \cdot 0+45 \cdot 1$
3	0	$12=12-45 \cdot 0$	$12=(12 \cdot 1+45 \cdot 0)-(12 \cdot 0+45 \cdot 1) \cdot 0$	$12=12 \cdot 1+45 \cdot 0$
4	3	$9=45-12 \cdot 3$	$9=(12 \cdot 0+45 \cdot 1)-(12 \cdot 1+45 \cdot 0) \cdot 3$	$9=12 \cdot (-3)+45 \cdot 1$
5	1	$3=12-9 \cdot 1$	$3=(12 \cdot 1+45 \cdot 0)-(12 \cdot (-3)+45 \cdot 1) \cdot 1$	$3=12 \cdot 4+45 \cdot (-1)$
6	3	0		

# Modulski multiplikativni inverz

- če sta  $a$  in  $n$  tuji števili, ima enačba  $ax \equiv b \pmod{n}$  enolično rešitev
- v posebnem primeru, ko je  $b=1$ , je rešitev enačbe  $ax \equiv 1 \pmod{n}$  multiplikativni inverz od  $a \pmod{n}$ 
  - če  $a$  in  $n$  nista tuji števili, enačba ni rešljiva
  - enačbo pogosto zapišemo tudi kot  $x \equiv a^{-1} \pmod{n}$ , vendar tukaj eksponent -1 ne pomeni potenciranja
- primeri:  
 $115x \equiv 1 \pmod{37} \Leftrightarrow x \equiv 115^{-1} \pmod{37} \Leftrightarrow x \equiv 4^{-1} \pmod{37}$   
 $140x \equiv 1 \pmod{33} \Leftrightarrow x \equiv 140^{-1} \pmod{33} \Leftrightarrow x \equiv 8^{-1} \pmod{33}$   
 $15x \equiv 1 \pmod{21} \Leftrightarrow x \not\equiv 15^{-1} \pmod{21}$ , ker 15 in 21 nista tuji

# Modulski multiplikativni inverz

- primer:  $4x \equiv 1 \pmod{37}$  oz.  $x \equiv 4^{-1} \pmod{37}$ ,
  - $a=4, n=37$
  - $d=1$  ✓
  - $x = (-9) \pmod{37} = (-9+1 \cdot 37) \pmod{37} = 28$  (želimo pozitivno vrednost)

korak	kvocient	ostanek	substitucija	kombiniran izraz
1		4		$4=4 \cdot 1+37 \cdot 0$
2		37		$37=4 \cdot 0+37 \cdot 1$
3	0	$4=4-37 \cdot 0$	$4=(4 \cdot 1+37 \cdot 0)-(4 \cdot 0+37 \cdot 1) \cdot 0$	$4=4 \cdot 1+37 \cdot 0$
4	9	$1=37-4 \cdot 9$	$1=(4 \cdot 0+37 \cdot 1)-(4 \cdot 1+37 \cdot 0) \cdot 9$	$1=4 \cdot (-9)+37 \cdot 1$
5	4	0		

# Eulerjeva funkcija $\varphi$

- **Eulerjeva ali totientna funkcija**  $\varphi(n)$  vrne število pozitivnih celih števil, ki so številu  $n$  tuja in ga ne presegajo (1 štejemo kot tuje število)
- zgled:  $\varphi(12) = 4$  (1,5,7,11)
- posebni primeri:
  - $n$  je praštevilo  $\Rightarrow \varphi(n)=n-1$
  - $n$  je  $m$ -ta potenca praštevila  $p$  ( $n=p^m$ )  $\Rightarrow \varphi(n)=\varphi(p^m)=p^{m-1} \cdot (p-1)$
  - $n$  je produkt tujih števil  $a$  in  $b$   $\Rightarrow \varphi(n)=\varphi(ab)=\varphi(a)\varphi(b)$
- splošen primer: 
$$\varphi(n) = n \prod_{p_i | n} \left(1 - \frac{1}{p_i}\right)$$

kjer so  $p_i$  prafaktorji od  $n$  (praštevila, ki delijo  $n$ )

# Eulerjeva funkcija $\varphi$

- **Eulerjeva ali totientna funkcija**  $\varphi(n)$  vrne število pozitivnih celih števil, ki so številu  $n$  tuja in ga ne presegajo (1 štejemo kot tuje število)
- zgled:  $\varphi(12) = 4$  (1,5,7,11)
- posebni primeri:
  - $n$  je praštevilo  $\Rightarrow \varphi(n)=n-1$
  - $n$  je  $m$ -ta potenca praštevila  $p$  ( $n=p^m$ )  $\Rightarrow \varphi(n)=\varphi(p^m)=p^{m-1} \cdot (p-1)$
  - $n$  je produkt tujih števil  $a$  in  $b \Rightarrow \varphi(n)=\varphi(ab)=\varphi(a)\varphi(b)$
- splošen primer:  $\varphi(n) = n \prod_{p_i|n} \left(1 - \frac{1}{p_i}\right)$

$$\varphi(4851) = \varphi(3^2 \cdot 7^2 \cdot 11^1) = 4851 \cdot \left(1 - \frac{1}{3}\right) \cdot \left(1 - \frac{1}{7}\right) \cdot \left(1 - \frac{1}{11}\right) = 2520$$



# Eulerjev izrek

- Za poljubni tuji števili  $a$  in  $n$  iz  $\mathbb{Z}^+$  velja  $a^{\varphi(n)} \equiv 1 \pmod{n}$
- Osnova šifrirnega sistema RSA, kjer je  $n$  produkt dveh velikih praštevil
- Omogoča zmanjšanje velikih potenc modulo  $n$
- Zgled:  $8^{975} \bmod 21 = ?$ 
  - 1) preverimo, da sta  $a=8$  in  $n=21$  tuji si števili ( $8=2^3$ ,  $21=3 \cdot 7$  ✓ )
  - 2) izračunamo  $\varphi(n)$ :  $\varphi(21)=\varphi(3 \cdot 7)=21 \cdot (1-1/3) \cdot (1-1/7)=12$
  - 3) zapišemo kongruenco po Eulerjevem izreku:  $8^{12} \equiv 1 \pmod{21}$
  - 4) uporabimo prejšnji korak za znižanje potence:  

$$8^{975} \equiv 8^{12 \cdot 81 + 3} \equiv (8^{12})^{81} \cdot 8^3 \equiv 1^{81} \cdot 8^3 \equiv 512 \pmod{21} = 8$$
- Ni vedno uporabno, npr.:  $12^{138} \bmod 203$ ,  $\varphi(203)=168$
- Mali Fermatov izrek je posebna oblika Eulerjevega izreka za primer, ko je  $n$  praštevilo:  $a^{n-1} \equiv 1 \pmod{n}$

# Modulsko potenciranje

- Problem: iščemo rešitev  $a^b \bmod n$  za cela števila  $a$ ,  $b$  in  $n$
- Pomembna operacija pri testiranju praštevil in v šifrirnih sistemih
- Učinkovit algoritem je metoda ponavljajočega kvadriranja:
  - deluje z binarno predstavitvijo eksponenta  $b$
  - časovna zahtevnost algoritma je  $O(\beta)$  aritmetičnih operacij in  $O(\beta^3)$  bitnih operacij, če so  $a$ ,  $b$  in  $n$   $\beta$ -bitna števila

```
MOD_EXP(a, b, n)
  d ← 1
   $\langle b_k, b_{k-1}, \dots, b_0 \rangle \leftarrow$  binarna predstavitev od b
  for i ← k downto 0
    d ← (d · d) mod n
    if  $b_i = 1$ 
      d ← (d · a) mod n
  return d
```

# Modulsko potenciranje


- Zgled:  $8^{975} \bmod 21 = ?$ 
  - pretvorimo  $b=975$  v dvojiško obliko:  
 $975_{[10]} = 1111001111_{[2]}$
  - izvedemo metodo s tabeliranjem vrednosti  $d$ :

$$\begin{aligned}
 975 &= 2 \cdot 487 + \mathbf{1} \\
 487 &= 2 \cdot 243 + \mathbf{1} \\
 243 &= 2 \cdot 121 + \mathbf{1} \\
 121 &= 2 \cdot 60 + \mathbf{1} \\
 60 &= 2 \cdot 30 + \mathbf{0} \\
 30 &= 2 \cdot 15 + \mathbf{0} \\
 15 &= 2 \cdot 7 + \mathbf{1} \\
 7 &= 2 \cdot 3 + \mathbf{1} \\
 3 &= 2 \cdot 1 + \mathbf{1} \\
 1 &= 2 \cdot 0 + \mathbf{1}
 \end{aligned}$$

$i$	9	8	7	6	5	4	3	2	1	0
$b_i$	1	1	1	1	0	0	1	1	1	1
$d$	8	8	8	8	1	1	8	8	8	<b>8</b>

# Modulsko potenciranje

- Zgled:  $12^{138} \bmod 203 = ?$ 
  - pretvorimo  $b=138$  v dvojiško obliko:  
 $138_{[10]} = 10001010_{[2]}$
  - izvedemo metodo s tabeliranjem vrednosti  $d$ :

$$\begin{aligned}
 138 &= 2 \cdot 69 + 0 \\
 69 &= 2 \cdot 34 + 1 \\
 34 &= 2 \cdot 17 + 0 \\
 17 &= 2 \cdot 8 + 1 \\
 8 &= 2 \cdot 4 + 0 \\
 4 &= 2 \cdot 2 + 0 \\
 2 &= 2 \cdot 1 + 0 \\
 1 &= 2 \cdot 0 + 1
 \end{aligned}$$


$i$	7	6	5	4	3	2	1	0
$b_i$	1	0	0	0	1	0	1	0
$d$	12	144	30	88	157	86	41	57

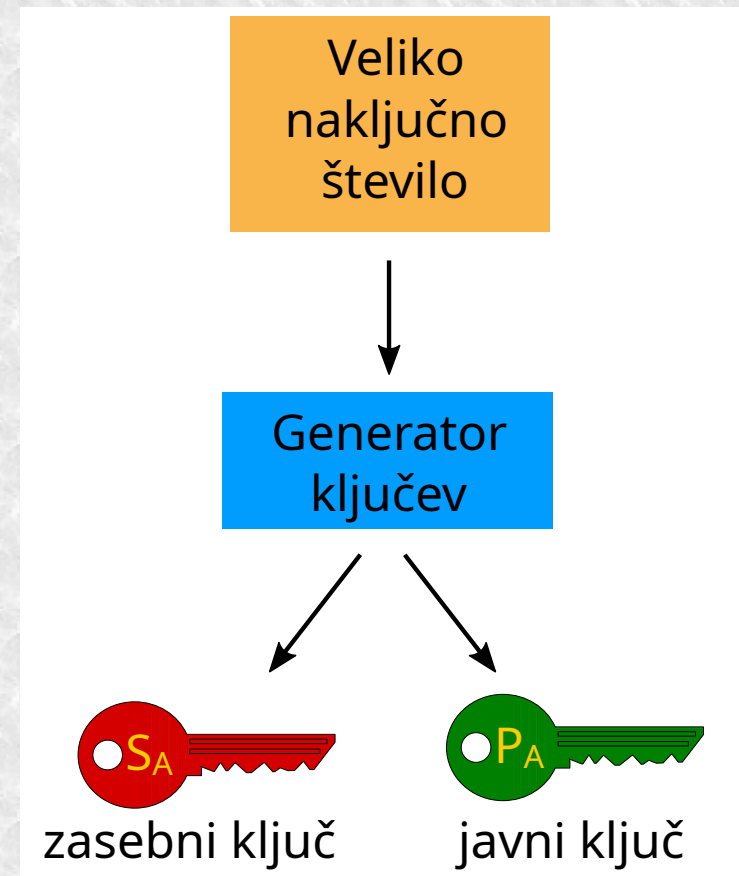
# Kriptografija

- skrivnopolisje, kriptologija – znanstvena veda, ki se ukvarja z odkrivanjem in preučevanjem računalniških algoritmov in protokolov za učinkovito zaščito informacij
- **čistopolis** (cleartext) – osnovno nezaščiteno sporočilo
- **šifropolis**, **tajnopolis** (cyphertext) – šifrirano sporočilo
- **simetrična kriptografija**:
  - uporablja isti ključ za šifriranje in dešifriranje (najbolj znan algoritem AES)
  - varen prenos ključa je problem sam po sebi
- **nesimetrična kriptografija** oz. **kriptografija javnega ključa** uporablja ločena ključa za šifriranje in dešifriranje



# Kriptografija javnega ključa

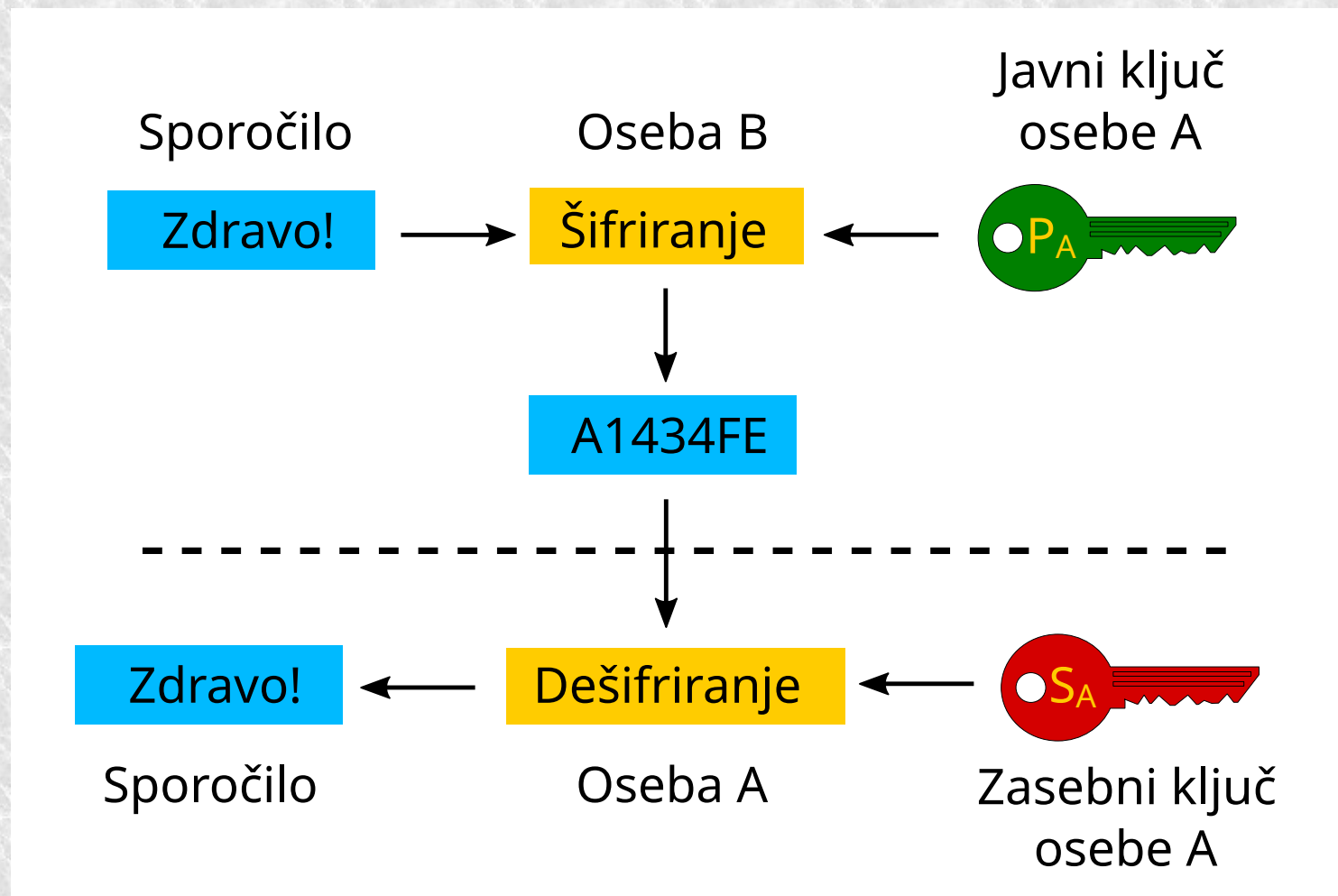
- vsak uporabnik generira svoj **javni ključ** (public key) in **zasebni** oz. **tajni ključ** (private key, secret key)
- tvorba para ključev je hitra, ugotovitev zasebnega ključa ob poznavanju javnega pa praktično neizvedljiva
- javni ključ se lahko objavi, zasebni ostane znan samo lastniku
- javni in zasebni ključ izvajata inverzni funkciji, ki omogočata šifriranje/dešifriranje sporočila  $M$ :
  - $M = S_A(P_A(M)) = P_A(S_A(M))$
  - $P_A$  – javni ključ osebe A
  - $S_A$  – zasebni ključ osebe A





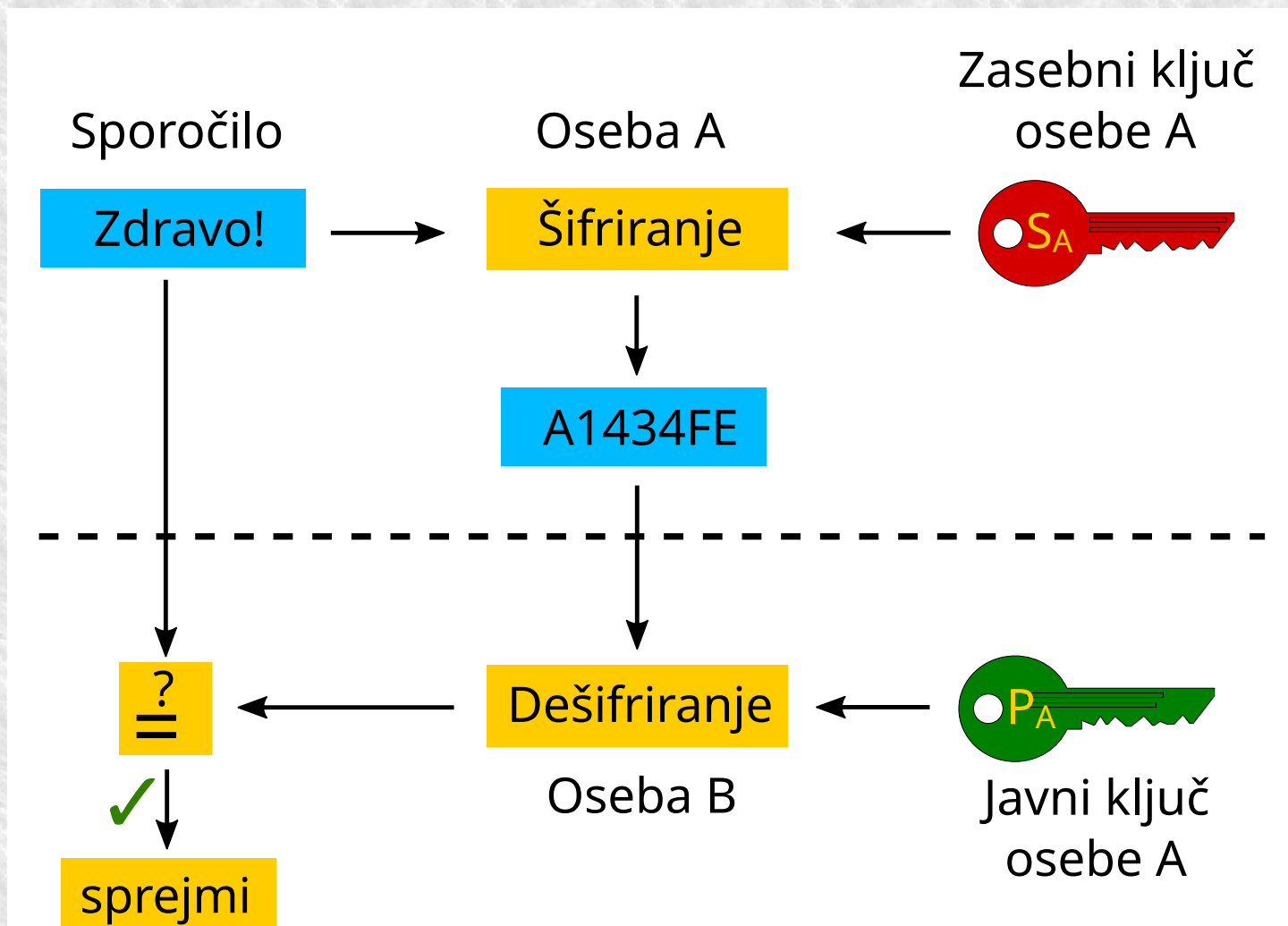
# Kriptografija javnega ključa

- Zgled: Šifriranje sporočila



# Kriptografija javnega ključa

- Zgled: Digitalni podpis



# Algoritem RSA

- algoritem za šifriranje z javnim ključem
- opisali Rivest, Shamir in Adleman leta 1977, patentiran do 2000
- tvorba ključev:
  - 1) izberi dve različni naključni veliki praštevili  $p$  in  $q$
  - 2) izračunaj  $n=p \cdot q$
  - 3) izberi majhno liho število  $e$ , ki je tuje številu  $\varphi(n)=(p-1) \cdot (q-1)$
  - 4) izračunaj multiplikativni inverz  $s$  od  $e$  (mod  $\varphi(n)$ ), t.j. reši enačbo  $s \cdot e \equiv 1 \pmod{\varphi(n)}$
  - 5) objavi  $P=(e,n)$  kot javni ključ RSA
  - 6) zadrži  $S=(s,n)$  kot tajni ključ RSA

# Algoritem RSA

Zgled: Tvorba ključev RSA pri  $p=31$  in  $q=19$ !

- 1)  $p = 31, q = 19$
- 2)  $n = p \cdot q = 589, \varphi(n) = (p-1) \cdot (q-1) = 540$
- 3)  $e = 11$  (izbrali smo prvo primerno vrednost  $> 10$ )
- 4)  $s = 491$  (prva pozitivna vrednost oblike  $-49+k \cdot 540$ )

korak	kvocient	ostanek	substitucija	kombiniran izraz
1		11		$11=11 \cdot 1+540 \cdot 0$
2		540		$540=11 \cdot 0+540 \cdot 1$
3	0	$11=11-540 \cdot 0$	$11=(11 \cdot 1+540 \cdot 0)-(11 \cdot 0+540 \cdot 1) \cdot 0$	$11=11 \cdot 1+540 \cdot 0$
4	49	$1=540-11 \cdot 49$	$1=(11 \cdot 0+540 \cdot 1)-(11 \cdot 1+540 \cdot 0) \cdot 49$	$1=11 \cdot (-49)+540 \cdot 1$
5	11	0		$d \checkmark \quad s$

- 5)  $P = (11, 589)$
- 6)  $S = (491, 589)$

# Šifriranje RSA

- šifriranje sporočila  $M$ :

$$C = P(M) = M^e \bmod n$$

- dešifriranje sporočila  $C$ :

$$M = S(C) = C^s \bmod n$$

- šifriranje in dešifriranje izvedemo z metodo ponavljajočega kvadriranja
- varnost algoritma RSA temelji na predpostavki, da je faktorizacija velikega celega števila  $n$  na  $p$  in  $q$  zahtevna
- trenutno priporočilo je, da naj bosta  $p$  in  $q$  dolga vsaj 512 bitov (RSA-768 (768-bitni  $n$  z natanko dvema faktorjema) je bil uspešno faktoriziran leta 2009)

# Šifriranje RSA

Zgled: Šifriranje sporočila  $M = 234$  s prejšnjima ključema!

1)  $P = (11, 589)$ ,  $S = (491, 589)$ ,  $M = 234$

2)  $C = P(M) = M^e \bmod n = 234^{11} \bmod 589 = 549$ ,  $b = 11_{[10]} = 1011_{[2]}$

$i$	3	2	1	0
$b_i$	1	0	1	1
$d$	234	568	119	549

3)  $M = S(C) = C^s \bmod n = 549^{491} \bmod 589 = 234$ ,  $b = 491_{[10]} = 111101011_{[2]}$

$i$	8	7	6	5	4	3	2	1	0
$b_i$	1	1	1	1	0	1	0	1	1
$d$	549	201	176	216	125	518	329	99	234

4)  $M = S(P(M))$  ✓



# Iskanje velikih praštevil

- prvi korak tvorbe ključev RSA:
  - če  $p$  in  $q$  nista praštevili, ne velja nujno  $P(S(M))=M$
- kako težko je najti praštevilo? **porazdelitvena funkcija gostote praštevil**  $\pi(n)$  določa število praštevil  $\leq n$
- primer:  $\pi(11)=5$  (2,3,5,7,11)
- izrek o praštevilu:  $\pi(n) \approx \frac{n}{\ln n}$
- primer:  $n=10^9$ ,  $\pi(n)=50847534$ ,  $n/\ln(n)=48254942$ ,  
 $\pi(n)-n/\ln(n)=2592592$ ,  $\pi(n)/(n/\ln(n))=1.054$ ,  $\pi(n)/n \approx 5\%$
- boljša aproksimacija (log. integral):  $\pi(n) \approx \text{li}(n) = \int_2^n \frac{1}{\ln t} dt$
- primer:  $n=10^9$ ,  $\pi(n)=50847534$ ,  $\text{li}(n)=50849235$ ,  $\Delta=1701$

# Iskanje velikih praštevil

- celo število  $n$  je praštevilo z verjetnostjo  $1/\ln(n)$  (potrebno je preveriti  $\ln(n)$  naključnih celih števil, da najdemo praštevilo enake dolžine kot  $n$ )
- primer: da najdemo 100-mestno praštevilo, moramo testirati približno 230 naključno izbranih 100-mestnih celih števil (pol manj, če izbiramo samo liha števila)
- sestavljeno število  $n$  je **psevdo praštevilo z bazo  $a$** , če velja  $a^{n-1} \equiv 1 \pmod{n}$

# Psevdo test praštevil

- Fermatov izrek: če je  $n$  praštevilo, velja  $a^{n-1} \equiv 1 \pmod{n}$  za  $\forall a \in \mathbb{Z}^+$ , ki je tuj  $n$
- če najdemo  $a$ , za katerega enačba ne velja, potem  $n$  gotovo ni praštevilo
- obrat prejšnje trditve **skoraj vedno** velja
- **Carmichaelova števila** – sestavljena števila, ki izpolnjujejo pogoj  $a^{n-1} \equiv 1 \pmod{n}$  za vsak  $a < n$ , ki je tuj  $n$

# Psevdo test praštevil

- želimo preveriti, ali je  $n$  praštevilo
- ideja: kongruenco  $a^{n-1} \equiv 1 \pmod{n}$  preverimo za  $a=2$  (uporabimo modulsko potenciranje)
  - če enačba ne velja, je  $n$  sestavljeno število
  - če enačba velja, je  $n$  praštevilo ali **psevdo praštevilo** z bazo 2

```
PSEUDOPRIME(n)
  d ← MOD_EXP(2, n-1, n)
  if d ≠ 1 (mod n)
    return false    // zagotovo
  else
    return true     // z veliko verjetnostjo
```



# Psevdo test praštevil

- želimo preveriti, ali je  $n$  praštevilo
- ideja: kongruenco  $a^{n-1} \equiv 1 \pmod{n}$  preverimo za  $a=2$  (uporabimo modulsko potenciranje)
  - če enačba ne velja, je  $n$  sestavljeno število
  - če enačba velja, je  $n$  praštevilo ali **psevdo praštevilo** z bazo 2
- verjetnost, da je naključno izbrano 512-bitno število psevdo praštevilo z bazo 2, je reda  $10^{-20}$
- za 1024-bitno število je ta verjetnost reda  $10^{-41}$
- za praktično uporabo je test dovolj zanesljiv

# Psevdo test praštevil - zgled

- ali je 2309 praštevilo?

–  $a=2$ ,  $b=2308_{[10]}=100100000100_{[2]}$ ,  $n=2309$

$i$	11	10	9	8	7	6	5	4	3	2	1	0
$b_i$	1	0	0	1	0	0	0	0	0	1	0	0
$d$	2	4	16	512	1227	61	1412	1077	811	1621	2308	1

- 2309 je praštevilo ali psevdo praštevilo z bazo 2, ker  $2^{2308} \equiv 1 \pmod{2309}$  (dejansko je praštevilo)



# Psevdo test praštevil - zgled

- ali je 1651 praštevilo?
  - $a=2$ ,  $b=1650_{[10]}=11001110010_{[2]}$ ,  $n=1651$

$i$	10	9	8	7	6	5	4	3	2	1	0
$b_i$	1	1	0	0	1	1	1	0	0	1	0
$d$	2	8	64	794	1159	385	921	1278	445	1461	1429

- 1651 ni praštevilo, ker  $2^{1650} \not\equiv 1 \pmod{1651}$

# Psevdo test praštevil - zgled

- ali je 1387 praštevilo?
  - $a=2$ ,  $b=1386_{[10]}=10101101010_{[2]}$ ,  $n=1387$

$i$	10	9	8	7	6	5	4	3	2	1	0
$b_i$	1	0	1	0	1	1	0	1	0	1	0
$d$	2	4	32	1024	8	128	1127	661	16	512	1

- 1387 je praštevilo ali psevdo praštevilo z bazo 2, ker  $2^{1386} \equiv 1 \pmod{1387}$  (dejansko ni praštevilo, ker je  $1387=73 \cdot 19$ )

# Miller-Rabinov test praštevil

- število  $x$  je **netrivialni koren od 1 (mod  $n$ )**, če velja  $x^2 \equiv 1 \pmod{n}$ ,  $x \not\equiv 1 \pmod{n}$  in  $x \not\equiv -1 \pmod{n}$
- pomožni izrek: če obstaja netrivialni koren od 1 (mod  $n$ ), potem je  $n$  sestavljeno število
- Miller-Rabinov test uvaja dve modifikaciji glede na psevdo test praštevil:
  - poskuša z več naključno izbranimi bazami namesto samo z  $a=2$
  - med modulskim potenciranjem preverja, ali se med kvadrati pojavi netrivialni koren od 1 (mod  $n$ )

# Miller-Rabinov test praštevil

- vhod je liho število  $n$ , ki ga testiramo, in število ponovitev  $s$  z različnimi naključno izbranimi bazami med 1 in  $n-1$
- pomožna funkcija WITNESS izvaja dejanski test pri izbrani bazi  $a$

```
MILLER_RABIN( $n, s$ )  
  for  $j \leftarrow 1$  to  $s$   
     $a \leftarrow \text{random}(1, n-1)$   
    if WITNESS( $a, n$ )  
      return false      // zagotovo  
  return true           // skoraj gotovo
```

# Miller-Rabinov test praštevil

- binarni zapis sodega števila  $n-1$  predstavimo kot binarni zapis lihega števila  $u$ , ki mu sledi  $t$  ničel:  $n-1=2^t \cdot u$
- velja  $a^{n-1} \equiv (a^u)^{2^t} \pmod{n} \Rightarrow$  izračunamo  $a^u \pmod{n}$  in rezultat  $t$ -krat kvadriramo

```

u ← n - 1
t ← 0
while u mod 2 = 0
    u ← u div 2
    t ← t + 1

```

```

WITNESS(a, n)
    določi lihi u in t ≥ 1, tako da n-1=2t·u
    x0 ← MOD_EXP(a, u, n)      // x0 ← au mod n
    for i ← 1 to t
        xi ← xi-12 mod n      // xi ≡ a2iu mod n
        if xi=1 and xi-1≠1 and xi-1≠n-1
            return true        // n je gotovo sestavljen
    if xt≠1
        return true            // n je gotovo sestavljen
    return false               // n je zelo verjetno praštevilo

```



# Miller-Rabinov test praštevil

- zgornja meja verjetnosti napake Miller-Rabinovega testa je enaka  $4^{-s}$  za vsak lihi  $n > 2$
- če je  $n$   $\beta$ -bitno število, je časovna zahtevnost M-R testa  $O(s\beta)$  aritmetičnih in  $O(s\beta^3)$  bitnih operacij
- zgled: testirajmo, ali je 2197 praštevilo, če je  $a=31$ !
  - $n = 2197$ ,  $a = 31$ ,  $n-1 = 2196_{[10]} = 100010010100_{[2]}$
  - $u = 1000100101_{[2]} = 549_{[10]}$ ,  $t = 2$
  - z modulskim potenciranjem izračunamo  $x_0 = a^u \bmod n = 31^{549} \bmod 2197 = 1643$
  - WITNESS:
 

$i$	0	1	2
$x_i$	1643	1533	1496
  - 2197 ni praštevilo, ker  $x_2 \neq 1$



# Miller-Rabinov test praštevil

- zgornja meja verjetnosti napake Miller-Rabinovega testa je enaka  $4^{-s}$  za vsak lihi  $n > 2$
- če je  $n$   $\beta$ -bitno število, je časovna zahtevnost M-R testa  $O(s\beta)$  aritmetičnih in  $O(s\beta^3)$  bitnih operacij
- zgled: testirajmo, ali je 2273 praštevilo, če je  $a=65$ !
  - $n = 2273$ ,  $a = 65$ ,  $n-1 = 2272_{[10]} = 100011100000_{[2]}$
  - $u = 1000111_{[2]} = 71_{[10]}$ ,  $t = 5$
  - z modulskim potenciranjem izračunamo  $x_0 = a^u \bmod n = 65^{71} \bmod 2273 = 1601$
  - WITNESS:
 

$i$	0	1	2	3	4	5
$x_i$	1601	1530	1983	2272	1	1
  - 2273 je praštevilo

# Miller-Rabinov test praštevil

- zgornja meja verjetnosti napake Miller-Rabinovega testa je enaka  $4^{-s}$  za vsak lihi  $n > 2$
- če je  $n$   $\beta$ -bitno število, je časovna zahtevnost M-R testa  $O(s\beta)$  aritmetičnih in  $O(s\beta^3)$  bitnih operacij
- zgled: testirajmo, ali je 2273 praštevilo, če je  $a=65$ !
  - $n = 2273$ ,  $a = 65$ ,  $n-1 = 2272_{[10]} = 100011100000_{[2]}$
  - $u = 1000111_{[2]} = 71_{[10]}$ ,  $t = 5$
  - z modulskim potenciranjem izračunamo  $x_0 = a^u \bmod n = 65^{71} \bmod 2273 = 1601$
  - WITNESS:
 

$i$	0	1	2	3	4	5
$x_i$	1601	1530	1983	2272	1	1
  - 2273 je praštevilo

ne zaključimo, ker je  $x_{i-1} = n-1$

# Miller-Rabinov test praštevil

- zgornja meja verjetnosti napake Miller-Rabinovega testa je enaka  $4^{-s}$  za vsak lihi  $n > 2$
- če je  $n$   $\beta$ -bitno število, je časovna zahtevnost M-R testa  $O(s\beta)$  aritmetičnih in  $O(s\beta^3)$  bitnih operacij
- zgled: testirajmo, ali je 1729 praštevilo, če je  $a=99$ !
  - $n = 1729$ ,  $a = 99$ ,  $n-1 = 1728_{[10]} = 11011000000_{[2]}$
  - $u = 11011_{[2]} = 27_{[10]}$ ,  $t = 6$
  - z modulskim potenciranjem izračunamo  $x_0 = a^u \bmod n = 99^{27} \bmod 1729 = 1331$
  - WITNESS:
 

$i$	0	1	2
$x_i$	1331	1065	1
  - 1729 ni praštevilo ( $=7 \cdot 13 \cdot 19$ )

# Miller-Rabinov test praštevil

- zgornja meja verjetnosti napake Miller-Rabinovega testa je enaka  $4^{-s}$  za vsak lihi  $n > 2$
- če je  $n$   $\beta$ -bitno število, je časovna zahtevnost M-R testa  $O(s\beta)$  aritmetičnih in  $O(s\beta^3)$  bitnih operacij
- zgled: testirajmo, ali je 1729 praštevilo, če je  $a=99$ !
  - $n = 1729$ ,  $a = 99$ ,  $n-1 = 1728_{[10]} = 11011000000_{[2]}$
  - $u = 11011_{[2]} = 27_{[10]}$ ,  $t = 6$
  - z modulskim potenciranjem izračunamo  $x_0 = a^u \bmod n = 99^{27} \bmod 1729 = 1331$
  - WITNESS:
 

$i$	0	1	2
$x_i$	1331	1065	1
  - 1729 ni praštevilo ( $=7 \cdot 13 \cdot 19$ )

zaključimo, ker  $x_{i-1} \neq 1$  in  $x_{i-1} \neq n-1$