ZAKRES MATERIAŁU

I. Testy pierwszości liczb i faktoryzacja
- **(A) Sito Eratostenesa**
- **(B) Algorytm Solovaya-Strassena - Probabilistyczny test na liczby pierwsze**

I. Testy pierwszości liczb i faktoryzacja

(A) Sito Eratostenesa

[TW01] Liczba złożona n, taka że n>1, ma dzielnik pierwszy p spełniający nierówność $p \le \sqrt{n}$.

[Algorytm] (Sito Eratostenesa) – służy do wyznaczania wszystkich liczb pierwszych mniejszych od liczby n

(1) Tworzymy tablicę t wartości int o rozmiarze n+1. W pozycje od 2 do n wpisujemy wartość 2.

(2) W kolejnych iteracjach dla i=2,..., $\lfloor \sqrt{n} \rfloor$,

- ♦ jeśli t[i]==1, to zerujemy wszystkie pozycje tablicy t o indeksach będących wielokrotnością i,

(3) tablica zawiera wartości równe 1 tylko na pozycjach o indeksach będących liczbami pierwszymi.

[Przykład 01] Wyznacz metodą sita Eratostenesa wszystkie liczby pierwsze mniejsze od 24.

```
t   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
    0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

$p = \lfloor \sqrt{24} \rfloor = \lfloor 4{,}898979486 \rfloor = 4$

```
i=2 0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
    0  0  1  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0  1  0
i=3 0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
    0  0  1  1  0  1  0  1  0  0  0  1  0  1  0  0  0  1  0  1  0  0  0  1  0
i=4 nie wykreślamy, ponieważ t[4]==0
```

Liczby pierwsze mniejsze lub równe od 24 to {2,3,5,7,11,13,17,19,23}

[Alorytm] (sito Eratostenesa)

```
//sprawdza, czy p jest nieparzystą liczbą pierwszą (sito Eratostenesa)
public boolean pierwsza(int p){
int[] t = new int[p+1];
for(int i=2; i<=p; i++)t[i]=1;
int a = (int)Math.floor(Math.sqrt(p));
for(int i=2; i<=a; i++){
  if(t[i]==0)continue;
  for(int j=2*i; j<=p; j+=i)t[j]=0;}String s="";
for(int i=2; i<=p; i++)if(t[i]==1)s+=i+",";
return (t[p]==1)&&(p!=2);}
```

(B) Algorytm Solovay'a-Strassena - Probabilistyczny test na liczby pierwsze

(♦) Podstawy matematyczne

[DEF 01] (problem decyzyjny)
Problem, w którym odpowiedzią na postawione pytanie jest „tak" albo „nie".

[DEF 02] (pozytywnie nastawiony algorytm Monte Carlo)
Algorytm probabilistyczny rozstrzygania problemu decyzyjnego, w którym odpowiedź „tak" jest zawsze poprawna, a odpowiedź „nie" może być błędna.

[DEF 03] (reszta kwadratowa mod p)

- ♦ x jest **resztą kwadratową (mod p)** wtw kongruencja $y^2 \equiv x (\mathrm{mod}\, p)$ ma rozwiązanie $y \in Z_p$.
- ♦ x jest **niereszta kwadratową (mod p)**, gdy $x \not\equiv 0 (\mathrm{mod}\, p)$ oraz x nie jest resztą kwadratową (mod p).

[Przykład 01] Resztami kwadratowymi (mod 17) są liczby 1, 2, 4, 8, 9, 13, 15, 16, ponieważ

$(\pm 1)^2 \equiv 1 (\mathrm{mod}\, 17)$, $(\pm 6)^2 \equiv 2 (\mathrm{mod}\, 17)$, $(\pm 2)^2 \equiv 4 (\mathrm{mod}\, 17)$, $(\pm 5)^2 \equiv 8 (\mathrm{mod}\, 17)$, $(\pm 3)^2 \equiv 9 (\mathrm{mod}\, 17)$,

$(\pm 8)^2 \equiv 13 (\mathrm{mod}\, 17)$, $(\pm 7)^2 \equiv 15 (\mathrm{mod}\, 17)$, $(\pm 4)^2 \equiv 16 (\mathrm{mod}\, 17)$

[TW 02] (kryterium Eulera)
Niech p – nieparzysta liczba pierwsza. Wówczas
x jest resztą kwadratową (mod p) wtw $x^{(p-1)/2} \equiv 1 (\mathrm{mod}\, p)$.

[DEF 04] (symbol Legendre'a)
Niech p – nieparzysta liczba pierwsza. Wówczas

Dla dowolnej liczby naturalnej $a \ge 0$ symbol Legendre'a $\left(\frac{a}{p}\right)$ definiujemy jako

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{gdy } a \equiv 0 (\mathrm{mod}\, p) \\ 1 & \text{gdy } a \text{ jest reszta kwadratowa } (\mathrm{mod}\, p) \\ -1 & \text{gdy } a \text{ jest niereszta kwadratowa } (\mathrm{mod}\, p) \end{cases}$$

[TW 02] Niech p – nieparzysta liczba pierwsza. Wówczas $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} (\mathrm{mod}\, p)$

[DEF 05] (symbol Jacobiego)
Niech p będzie nieparzystą dodatnią liczbą naturalną i niech $p_1^{e_1} \ldots p_k^{e_k}$ będzie rozkładem liczby n na czynniki pierwsze. Dla liczby naturalnej $a \ge 0$ definiujemy symbol Jacobiego $\left(\frac{a}{n}\right)$ następująco $\left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}$.

[Przykład 03] Policz symbol Jacobiego $\left(\frac{4369}{5355}\right)$.

$$\left(\frac{4369}{5355}\right) = \left(\frac{4369}{3}\right)^2 \left(\frac{4369}{5}\right)\left(\frac{4369}{7}\right)\left(\frac{4369}{17}\right) = \left(\frac{1}{3}\right)^2 \left(\frac{4}{5}\right)\left(\frac{1}{7}\right)\left(\frac{0}{17}\right) = 1^2 * 1 * 1 * 0 = 0$$

[Własności] (symbolu Jacobiego)
Niech n – nieparzysta liczba naturalna dodatnia (niekoniecznie pierwsza)

(1) $\left(\frac{a}{n}\right) = 0$ wtw $NWD(a,n) \neq 1$

(2) $\left(\frac{1}{n}\right) = 1$

(3) jeśli a jest nieparzystą liczbą naturalną, to $\left(\frac{a}{n}\right) = \begin{cases} -\left(\frac{n}{a}\right) & \text{gdy } m \equiv n \equiv 3 \pmod 4 \\ \left(\frac{n}{a}\right) & \text{w przeciwnym przypadku} \end{cases}$

(4) jeśli $a \equiv b \pmod n$, to $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$

(5) $\left(\frac{2}{n}\right) = \begin{cases} 1 & \text{gdy } n \equiv \pm 1 \pmod 8 \\ -1 & \text{gdy } n \equiv \pm 3 \pmod 8 \end{cases}$

(6) $\left(\frac{a_1 * a_2}{n}\right) = \left(\frac{a_1}{n}\right)\left(\frac{a_2}{n}\right)$

[Przykład 04] Korzystając z własności symbolu Jacobiego, policz $\left(\frac{5369}{9283}\right)$.

```
(5369,9283)=
=(9283,5369),   z(3)
=(3914,5369),   z(4)
=(1957,5369),   z(5)
=(5369,1957),   z(3)
=(1455,1957),   z(4)
=(1957,1455),   z(3)
=(502,1455),    z(4)
=(251,1455),    z(5)
=-(1455,251),   z(3)
=-(200,251),    z(4)
=(100,251),     z(5)
=-(50,251),     z(5)
=(25,251),      z(5)
=(251,25),      z(3)
=(1,25),        z(4)
=1,             z(2)
```

[Algorytm] (wyznaczania symbolu Jacobiego na podstawie własności (1)-(5))

```
private void w(String s){
System.out.println(s);}

//oblicza symbol Jacobiego dla n nieparzystego
public int symbJacob(int a, int n){
boolean b=true;
while(b){
  b=false;
  if(a==0){w("=0, (def s. Legendre'a)\n"); return 0;}
  if(a%2==0&&n%2==0){w("=0, (1)\n"); return 0;}
  if(a==1){w("="+znak+", (2)\n"); return znak;}
  if(a%2==1&&n%2==1&&a<n){
     if(a%4==3&&n%4==3)znak*=-1;
     int pom=n; n=a; a=pom;
     w("="+znak+"("+a+"/"+n+") (3)\n");
     b=true;
  }else
  if(n%2==1&&a>n){a=a%n;w("="+znak+"("+a+"/"+n+"), (4)\n");b=true;}
  else
     if(n%2==1&&a%2==0){
       if(n%8==1||(n+2)%8==1)a=(int)a/2;
       else
         if(n%8==3||(n+6)%8==3){a=(int)a/2;znak*=-1;}
       w("="+znak+"("+a+"/"+n+"), (5)\n");
       b=true;}}
return 2;}
```

[Uwaga 01] Niech n>1 – nieparzysta liczba naturalna

- Jeśli n jest liczbą pierwszą, to $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod n$ dla dowolnej liczby a.
- Jeśli n jest liczbą złożoną, to może (ale nie musi) zachodzić własność $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod n$.
- Jeśli n jest liczbą złożoną i $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod n$, to n nazywamy **pseudopierwszą liczbą Eulera** przy podstawie a.

[Przykład 05] 91 jest pseudopierwszą liczbą Eulera przy podstawie a=10, ponieważ 91 jest liczbą złożoną i $\left(\frac{10}{91}\right) = -1 = 10^{45} \pmod{91}$

- Z Uwagi 01 wynika algorytm sprawdzania pierwszości liczb Solovay'a-Strassena.

(♦) <u>Algorytm Solovay'a-Strassena</u>

- pozytywnie nastawiony algorytm Monte Carlo dla problemu rozkładalności z prawdopodobieństwem błędu równym co najwyżej 1/2.
  - odpowiedź „tak" oznacza, że liczba n jest złożona,
  - jeśli liczba n jest złożona, algorytm odpowie „tak" z prawdopodobieństwem 1/2.

[Algorytm] (Solovaya-Strassena) sprawdzania pierwszości liczb

(1) wybierz losowo liczbę całkowitą a $(1 \le a \le n-1)$,

(2) if $\left(\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod n\right)$ odpowiedź „n jest pierwsza"; else odpowiedź „n jest złożona",

---

KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

[Przykład 06] Przeanalizuj algorytm Solovay'a-Starssena na poniższych przykładach.
(a) (10,91)
(b) (18,773)
---
odp.(a)

```
x=-1          // symbJacob()
e=45          // (n-1)/2
e=101101
   x=1 a=10
e0 x=10 a=9
e1 x=10 a=81
e2 x=82 a=9
e3 x=10 a=81
e4 x=10 a=9
e5 x=90 a=81
y=90          // 10^45(mod91)
true          // błędna odpowiedź
```

(b) (18,773)

```
x=-1
e=386
e=110000010
   x=1 a=18
e0 x=1 a=324
e1 x=324 a=621
e2 x=324 a=687
e3 x=324 a=439
e4 x=324 a=244
e5 x=324 a=15
e6 x=324 a=225
e7 x=238 a=380
e8 x=772 a=622
y=772
true          // poprawna odpowiedź
```


KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego