***Реферат на тему:***

*Розклад числа на прості множники*

**Означення.** ***Розкладом*** натурального числа *n* ***на прості множники*** (***факторизацією*** числа) називається представлення його у вигляді *n* = , де *pi* – взаємно прості числа, *ki* 1 .

Задача перевірки числа на простоту є простішою за задачу факторизації. Тому перед розкладанням числа на прості множники слід перевірити число на простоту.

**Означення.** ***Розбиттям*** числа називається задача представлення натурального числа n у вигляді *n* = *a* \* *b*, де *a*, *b* – натуральні числа, більші за 1 (не обов’язково прості).

### Метод Ферма

Нехай *n* – складене число, яке не є степенем простого числа. Метод Ферма намагається знати такі натуральні *x* та *y*, що *n* = *x*2 – *y*2. Після чого дільниками числа *n* будуть *a* = *x* – *y* та *b* = *x* + *y*: *n* = *a* \* *b* = (*x* – *y*)(*x* + *y*).

Якщо припустити що *n* = *a* \* *b*, то в якості *x* та *y* (таких що *n* = *x*2 – *y*2) можна обрати

, 

**Приклад.** Виберемо *n* = 143 = 11 \* 13.

Тоді *x* = (13 + 11) / 2 = 12, *y* = (13 – 11) / 2 = 1.

#### Перевірка: *x*2 – *y*2 = 122 – 11 = 143 = *n*.

**Теорема.** Якщо *n* = *x*2 – *y*2, то  < *x* < (*n* + 1) / 2.

**Доведення.** З рівності *n* = *x*2 – *y*2 випливає, що *n* < *x*2, тобто  < *x*.

Оскільки *a* = *n* / *b*, то . Максимальне значення *x* досягається при мінімальному *b*, тобто при *b* = 1. Звідси x =  < .

Отже для пошуку представлення *n* = *x*2 – *y*2 слід перебрати всі можливі значення *x* із проміжку [, (*n* + 1) / 2], перевіряючи при цьому чи є вираз *x*2 - *n* повним квадратом.

**Приклад.** Розкласти на множники *n* = 391 методом Ферма.  = 19.

202 – 391 = 9 = 32. Маємо рівність: 391 = 202 – 32.

Звідси 391 = (20 – 3)(20 + 3) = 17 \* 23.

### Алгоритм Полард - ро факторизації числа

У 1974 році Джон Полард запропонував алгоритм знаходження нетривіального дільника натурального числа *n*. Пр цьому алгоритм використовує лише операції додавання, множення та віднімання модулярної арифметики.

Ідея алгоритма Полард – ро полягає в ітеративному обчисленні деякої наперед заданої поліноміальної функції *f* з цілими коефіцієнтами. Побудуємо послідовність *xi* наступним чином: *x*0 оберемо довільним із Z*n*, а *xi*+1 = f(*xi*) mod *n*, *i* 0. Оскільки *xi* можуть приймати лише скінченний набір значень (цілі числа від 0 до *n*), то існують такі цілі *n*1 та *n*2 (*n*1 < *n*2), що = . Враховуючи поліноміальність *f*, для кожного натурального *k* маємо: =, тобто починаючи з індекса *i* = *n*1 послідовність {*xi* mod *n*} буде періодичною.

**Приклад.** Нехай *n* = 21, *x*0 = 1, *xi*+1 =  + 3 mod 21.

Тоді послідовність *xi* має вигляд: 1, 4, 19, 7, 10, 19, 7, 10, ... .

Таким чином *x*3 = *x*6, період послідовності дорівнює 3.

Послідовність *xi* можна відобразити у вигляді кола з хвостом: коло відповідає періодичній частині, а хвіст – доперіодичній. Картинка нагадує грецьку літеру , тому метод який застосовується в алгоритмі називається  – евристикою. Послідовність із попереднього прикладу можна зобразити так:

7

10

19

4

1

Ідея алгоритму полягає в обчисленні для кожного *i* > 0 значення *d* = НСД(*x*2*i* – *xi*, *n*). Якщо на деякому кроці *d* > 1, то це і є нетривіальний дільник числа *n*.

Побудуємо послідовність елементів *xi* наступним чином:

*x*0 = 2, *xi*+1 = f(*xi*) = ( + 1) mod *n*, *i* > 0

### Алгоритм

Вхід: натуральне число *n*, параметр *t*  1.

Вихід: нетривіальний дільник *d* числа *n*.

1. *a* =2, *b* =2;

**2.** for *i* 1 to *t* do

2.1. Обчислити *a* *a*2 + 1) mod *n*; *b* *b*2 + 1) mod *n*; *b* *b*2 + 1) mod *n*;

2.2. Обчислити *d* НСД(*a* - *b*, *n*);

2.3. if 1 < *d* < *n* return (*d*); // знайдено нетривіальний дільник

3. return (False); // дільника не знайдено

Вважаємо, що функція f(*x*) = (*x*2 + 1) mod *n* генерує випадкові числа. Тоді для знаходження дільника числа *n* необхідно виконати не більш ніж O() операцій модулярного множення.

Якщо алгоритм Поларда – ро не знаходить дільника за *t* ітерацій, то замість функції f(*x*) = (*x*2 + 1) mod *n* можна використовувати f(*x*) = (*x*2 + c) mod *n*, для деякого цілого c, c  0, -2.

**Приклад.** Нехай *n* = 19939.

Послідовність *xi*: 2, 5, 26, 677, 19672, 11473, 12391, 6582, 15217, 5483, 15217, 5483, 15217, ... .

|  |  |  |
| --- | --- | --- |
| *a* | *b* | *d* |
| 2 | 2 | 1 |
| 5 | 26 | 1 |
| 26 | 19672 | 1 |
| 677 | 12391 | 1 |
| 19672 | 15217 | 1 |
| 11473 | 15217 | 1 |
| 12391 | 15217 | 157 |

Знайдено розклад 19939 = 157 \* 127.

Нехай *n* = 143. Послідовність *xi*: 2, 5, 26, 105, 15, ... .

|  |  |  |
| --- | --- | --- |
| *a* | *b* | *d* |
| 2 | 2 | 1 |
| 5 | 26 | НСД(21, 143) = 1 |
| 26 | 15 | НСД(11, 143) = 11 |

Знайдено розклад 143 = 11 \* 13.

### Ймовірносний квадратичний алгоритм факторизації числа

**Твердження.** Нехай *x* та *y* – цілі числа, *x*2  *y*2 (mod *n*) та *x*  *y* (mod *n*). Тоді *x*2 – *y*2 ділиться на *n*, при чому жоден із виразів *x* + *y* та *x* – *y* не ділиться на *n*. Число *d* = НСД(*x*2 – *y*2, *n*) є нетривіальним дільником *n*.

**Теорема.** Якщо *n* – непарне складене число, яке не є степенем простого числа, то завжди існують такі *x* та *y*, що *x*2  *y*2 (mod *n*), при чому *x*   *y* (mod *n*).

**Доведення.** Нехай *n* = *n*1 \* *n*2 – добуток взаємно простих чисел. Оберемо таке *y*, що НСД(*y*, *n*) = 1. Далі розв’яжемо систему рівнянь:



Розв’язком системи будуть такі *x* та *y* за модулем *n* = НСК(*n*1, *n*2), що *x*2  *y*2 (mod *n*). Якщо при цьому припустити, що *x*  – *y* (mod *n*), то з другого рівняння системи маємо: *y*  – *y* (mod *n*2), або 2 \* *y* = 0 (mod *n*2). Оскільки було обрано НСД(*y*, *n*2) = 1, то з останньої рівності випливає що *n*2 ділиться на 2, тобто є парним. Це суперечить умові теореми про непарність *n*.

**Приклад.** Виберемо *n*1 = 11, *n*2 = 13 – взаємно прості числа. Тоді *n* = 11 \* 13 = 143. Покладемо *y* = 5, НСД(5, 143) = 1. Складемо систему порівнянь:

 або

Розв’язком системи буде x  60 (mod 143).

Має місце рівність 602  52 (mod 143) , при чому 60  5 (mod 143).

Тоді дільником числа *n* буде *d* = НСД(60 – 5, 143) = 11.

Формально ймовірносний квадратичний алгоритм факторизації будується на наступній ідеї:

Нехай *F* = {*p*0, *p*1, *p*2, …, *pt*} – множникова основа, *pi* – різні прості числа, при чому дозволяється обрати *p*0 = -1. Побудуємо множину порівнянь

**  *zi* ,

таку що значення *zi* є повіністю факторизованими у множині *F* :

,

та добуток деякої підмножини значень *zi* є повним квадратом:

*z* =  = *y*2, *y* Z, *fi* {0, 1}

Якщо множина порівнянь із вказаними властивостями побудована, то поклавши *x* =  і перевіривши виконання нерівності *x*   *y* (mod *n*), отри маємо *x*2  *y*2 (mod *n*). Число *d* = НСД(*x*2 – *y*2, *n*) є нетривіальним дільником *n*.

**Приклад.** Знайти дільник числа *n* = 143.

Обираємо випадково число *x*  [2, 142], обчислюємо x2 (mod 143) та розкладаємо результат на множники:

1. z1 = 192 (mod 143) = 75 = 3 \* 52.

2. z2 = 772 (mod 143) = 66 = 2 \* 3 \* 11.

3. z3 = 292 (mod 143) = 126 = 2 \* 32 \* 7.

4. z4 = 542 (mod 143) = 56 = 23 \* 7.

Можна помітити, що добуток z3 та z4 є повним квадратом:

z = z3 \* z4 = 24 \* 32 \* 72 = (22 \* 3 \* 7)2 = 842

Маємо рівність:

z3 \* z4 = 292 \* 542  842 (mod 143)

або враховуючи що 29 \* 54 (mod 143)  136, маємо:

1362 = 842 (mod 143), при чому 136  84 (mod 143)

Дільником числа *n* = 143 буде *d* = НСД(136 – 84, 143) = НСД(52, 143) = 13.

### Квадратичний алгоритм факторизації

Серед усіх існуючих алгоритмів факторизації найшвидшим є квадратичний. Він ефективно застосовується для чисел, кількість цифр яких менша за 100 та які не мають малих простих дільників. Еврістичний аналіз, проведений Померансом [1] у 1981 році показав, що число *N* може бути розкладено на множники за час .

Нехай *n* – число, яке факторизується, *m* = . Розглянемо многочлен

q(*x*) = (*x* + *m*)2 - *n*

Квадратичний алгоритм обирає *ai* = *x* + *m* (*x* = 0, 1, 2, …), обчислює значення *bi* = (*x* + *m*)2 – *n* та перевіряє, чи факторизується *bi* у множниковій основі *F* = {*p*0, *p*1, *p*2, …, *pt*}.

Помітимо, що  = (*x* + *m*)2 – *n*  (*x* + *m*)2 (mod *n*)  *bi* (mod *n*).

### Алгоритм

Вхід: натуральне число *n*, яке не є степенм простого числа.

Вихід: нетривіальний дільник *d* числа *n*.

1. Обрати множникову основу *F* = {*p*0, *p*1, *p*2, …, *pt*}, де *p*0 = -1, *pi* – *i* - те просте число *p*, для якого *n* є квадратичним лишком за модулем p.

2. Обчислити *m* = [].

3. Знаходження *t* + 1 пари (*ai*, *bi*).

 Значення *x* перебираються у послідовності 0, 1, 2, … .

 Покласти *i*  1. Поки *i*  *t* + 1 робити:

3.1. Обчислити *b* = q(*x*) = (*x* + *m*)2 – *n* та перевірити, чи розкладається *b* у множниковій основі *F*. Якщо ні, обрати наступне *x* та повторити цей крок.

3.2. Нехай *b* = . Покласти *ai* = *x* + *m*, *bi* = *b*, *vi* = (*vi*1, *vi*2, …, *vit*), де *vij* = *eij* mod 2, 1 *j* *t*.

3.3. *i*  *i* + 1.

4. Знайти підмножину T {1, 2, …, t + 1} таку що  = 0.

5. Обчислити *x* =  mod *n*.

6. Для кожного j, 1 *j* *t*, обчислити *lj* = () / 2.

7. Обчислити *y* =  mod *n*.

8. Якщо *x*  *y* (mod *n*), знайти іншу підмножину T {1, 2, …, t + 1} таку що  = 0 та перейти до кроку 5.

9. Обчислити дільник *d* = НСД(*x* – *y*, *n*).

**Приклад.** Розкласти на множники *n* = 24961.

1. Побудуємо множникову основу: *F* = {-1, 2, 3, 5, 13, 23}

2. *m* = [] = 157.

3. Побудуємо наступну таблицю:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *i* | *x* | q(*x*) | факторизація q(*x*) | *ai* | *vi* |
| 1 | 0 | -312 | -23 \* 3 \* 13 | 157 | (1, 1, 1, 0, 1, 0) |
| 2 | 1 | 3 | 3 | 158 | (0, 0, 1, 0, 0, 0) |
| 3 | -1 | -625 | -54 | 156 | (1, 0, 0, 0, 0, 0) |
| 4 | 2 | 320 | 26 \* 5 | 159 | (0, 0, 0, 1, 0, 0) |
| 5 | -2 | -936 | -23 \* 32 \* 13 | 155 | (1, 1, 0, 0, 1, 0) |
| 6 | 4 | 960 | 26 \* 3 \* 5 | 161 | (0, 0, 1 ,1, 0, 0) |
| 7 | -6 | -2160 | -24 \* 33 \* 5 | 151 | (1, 0, 1, 1, 0, 0) |

4. Виберемо T = {1, 2, 5}, оскільки *v*1 + *v*2 + *v*5 = 0.

5. Обчислимо x = (*a*1*a*2*a*5) (mod *n*) = 936 = 26 \* 34 \* 132.

6. *l*1 = 1, *l*2 = 3, *l*3 = 2, *l*4 = 0, *l*5 = 1, *l*6 = 0.

7. *y* = -23 \* 32 \* 13 (mod *n*) = 24025.

8. Оскільки 936 –24025 (mod *n*), необхідно шукати іншу множину T.

9. Виберемо T = {3, 6, 7}, оскільки *v*3 + *v*6 + *v*7 = 0.

10. Обчислимо x = (*a*3*a*6*a*7) mod *n* = 23405 = 210 \* 34 \* 56.

11. *l*1 = 1, *l*2 = 5, *l*3 = 2, *l*4 = 3, *l*5 = 0, *l*6 = 0.

12. *y* = -25 \* 32 \* 53 (mod *n*) = 13922.

13. 23405  13922 (mod n).

*d* = НСД(*x* – *y*, *n*) = НСД(9483, 24961) = 109 – дільник.

Відповідь: 109 – дільник 24961.