Приклад 7. При обчисленні 11^{13} спочатку знаходимо двійкові цифри числа 13: $(13)_{10} = (1101)_2$, тобто $13 = 2^3 + 2^2 + 2^0$. Потім послідовно обчислюємо $11^2 = 121$, $11^4 = 11^2 \cdot 11^2 = 14,641$, $11^8 = 11^4 \cdot 11^4 = 214,358,881$. Тепер

$$\begin{aligned} 11^{13} &= 11^{2^3} \times 11^{2^2} \times 11^{2^1} = 214,358,881 \times 14,641 \times 11 \\ &= 214,358,881 \times 161,051 \\ &= 34,522,712,143,931. \end{aligned}$$

Для чисел, розглянутих у прикладі, оптимізація обчислень не є суттєвою, але для тих чисел, які насправді необхідні в практичній роботі, вона може стати критично важливою.

5. Швидке піднесення до степеня за модулем

Для обчислення остачі $a^k \pmod{n}$ існує ефективний метод, принцип якого ми спочатку пояснюємо на простому числовому прикладі для $a = 7$, $k = 13$ та $n = 10$. Таким чином, ми обчислюємо $7^{13} \pmod{10}$.

Обчислити $7^{13} \pmod{10}$ нескладно й безпосередньо, помітивши, що послідовність остач 7^k при діленні на 10 є періодичною:

k	1	2	3	4	5	6	7	8	9	10	11	12	13
$7^k \pmod{10}$	7	9	3	1	7	9	3	1	7	9	3	1	7

Таким чином, $7 = 7^{13} \pmod{10}$.

Алгоритм, який ми збираємось продемонструвати, є універсальним. На першому його кроці необхідно знайти бінарне представления числа k. В данному випадку,

$$13 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \quad \text{або} \quad (13)_{10} = (1101)_2.$$

За допомогою цього представлення, будуємо послідовність обчислень, на кожному кроці якої виконується одна з двох наступних операцій:

1) $\uparrow$ (піднести до квадрату й обчислити $\bmod\ n$) або
2) $\uparrow\times$ (піднести до квадрату, помножити на a й обчислити $\bmod\ n$).

Послідовність операцій визначається наступним чином:

> якщо чергова двійкова цифра у двійковому запису числа n дорівнює 0, то застосовується операція ($\uparrow$), в іншому випадку — операція ($\uparrow\times$).

Аргументом кожної операції є результат обчислення попередньої; для першої операції аргументом є 1. Результат останньої операції дорівнює необхідній остачі. Алгоритм обчислення у загальному випадку представлено нижче. ㉒

Алгоритм 2. Процедура піднесення до степеня за модулем

Вхідні дані: натуральні числа a, k, n;

Вихідні дані: $a^k \pmod n$;

знайти двійкове представлення числа $k = (b_i b_{i-1} \dots b_0)_2$;

якщо $b_i = 0$, то $x_1 = 1^2 \pmod n$; інакше $x_1 = 1^2 \times a \pmod n$;

якщо $b_{i-1} = 0$, то $x_2 = x_1^2 \pmod n$; інакше $x_2 = x_1^2 \times a \pmod n$;

. . . . . .

якщо $b_0 = 0$, то $x_{i+1} = x_i^2 \pmod n$; інакше $x_{i+1} = x_i^2 \times a \pmod n$;

покласти $a^k \pmod n = x_{i+1}$.

Оскільки $(13)_{10} = (1101)_2$, то при обчисленні остачі від ділення 7^{13} на 10, послідовність операцій є такою:

$$\uparrow\times \qquad \uparrow\times \qquad \uparrow \qquad \uparrow\times$$

Тому алгоритм 2 здійснює такі обчислення:

1. $x_1 = (1^2 \times a) \pmod{n}$;
2. $x_2 = (x_1^2 \times a) \pmod{n}$;
3. $x_3 = x_2^2 \pmod{n}$;
4. $x_4 = (x_3^2 \times a) \pmod{n}$.

Якщо підставити числові значення, то отримаємо

1. $x_1 = 1^2 \times 7 \pmod{10} = 7$;
2. $x_2 = 7^2 \times 7 \pmod{10} = 343 \pmod{10} = 3$;
3. $x_3 = 3^2 \pmod{10} = 9$;
4. $x_4 = 9^2 \times 7 \pmod{10} = 567 \pmod{10} = 7$.

5.1. Бінарне представлення. Оскільки для швидкого обчислення a^k та $a^k \pmod{n}$ необхідним є знаходження двійкового запису числа, ми зупинимось на процедурах переводу чисел з десятичної системи у двійкову.

Розглянемо два найпростіших способа отримання бінарного представлення десяткового числа. Принцип роботи кожного з них розглянемо на прикладі десяткового числа $x = 15$.

5.1.1. Почати обчислення з першої цифри. Позначимо $x_0 = x$. Спочатку знаходимо найбільший степінь m_0 двійки, для якого $2^{m_0} \le x_0$. Зрозуміло, що у нашому прикладі $m_0 = 3$, оскільки $2^3 \le 15 < 2^4$. Тепер повторимо цю ж процедуру, але для числа $x_1 \stackrel{\text{def}}{=} x_0 - 2^{m_0} = 15 - 2^3 = 7$. Отримуємо число $m_1 = 2$, оскільки $2^2 \le 7 < 2^3$. Тепер обчислюємо $x_2 = x_1 - 2^{m_1} = 7 - 2^2 = 3$. Далі діємо за тим же принципом: знаходимо $m_2 = 1$, оскільки $2^1 \le 3 < 2^2$. Після цього обчислюємо $x_3 = x_2 - 2^{m_2} = 2 - 2^1 = 1$ й знаходимо $m_3 = 0$, оскільки $2^0 \le 1 < 2^2$. Нарешті обчислюємо $x_4 \stackrel{\text{def}}{=} x_3 - 2^{m_3} = 0$. У загальному випадку алгоритм закінчує

роботу, коли чергове x стає рівним 0.

Для числа 15 алгоритм зупиняється після обчислення x_4.

Двійкове представлення числа 15 складається з m_0+1 позицій: позиції нумеруються зліва направо, починаючи з 0. В позиціях m_0, m_1, m_2, ... двійкового представлення записуємо одиниці, в інших позиціях — нулі. ㉓

У нашому прикладі необхідні $m_0 + 1 = 4$ позиції для двійкового запису десяткового числа 15, причому в позиціях $m_0 = 3$, $m_1 = 2$, $m_2 = 1$, $m_3 = 0$ необхідно записатти двійкові одиниці. Таким чином, $(15)_{10} = (1111)_2$.

Алгоритм у загальному випадку має такий вигляд.

Алгоритм 3. Двійкове представлення (починаємо ліворуч)

Вхідні дані: натуральне число k;

Вихідні дані: двійкове представлення $k = (b_i \dots b_0)_2$;

знайти m_0: $2^{m_0} \le k < 2^{m_0+1}$ та покласти $x_1 = k - 2^{m_0}$, $i = 1$;

якщо $x_1 = 0$, то виконати процедуру **WriteExpansion**. `STOP`.

якщо ж $x_1 > 0$, то знайти m_1: $2^{m_1} \le x_1 < 2^{m_1+1}$ та

покласти $x_2 = x_1 - 2^{m_1}$, $i = 2$;

якщо $x_2 = 0$, то виконати процедуру **WriteExpansion**. `STOP`.

якщо ж $x_2 > 0$, то знайти m_2: $2^{m_2} \le x_2 < 2^{m_2+1}$ та

покласти $x_3 = x_2 - 2^{m_2}$, $i = 3$;

. . . . . .

Процедура **WriteExpansion**

покласти $b_{m_0} = 1$, $b_{m_1} = 1$, ... , $b_{m_i} = 1$

та $b_j = 0$ для всіх інших $0 \le j \le m_0$.

У представленому алгоритмі переводу числа з десяткової

системи у двійкову для зручності використовується проста процедура **WriteExpansion**, яка за результатами обчислень показників $m_0, m_1, \ldots$ записує двійкове представлення числа k, а саме в позиціях двійкового представлення з номерами $m_0, m_1, \ldots$ вона записує одиниці, а в інших позиціях — нулі. У розглянутому вище прикладі з $k = 15$ ця процедура записала б чотири одиниці й жодного нуля.

Особливістю алгоритму 3, представленого нижче, є необхідність обчислювати степені двійки. Ці обчислення виконуються швидко за рекурсивною формулою $2^k = 2^{k-1} \cdot 2$.

5.1.2. Почати обчислення з останньої цифри. Існує ще один алгоритм переведення чисел з десяткової системи у двійкову. Його особливість у тому, що двійкові цифри обчислюються починаючи з молодших розрядів. Перевага цього алгоритму ㉔ у тому, що він не потребує попереднього обчислення ступенів двійки. Недоліком алгоритму 4 є те, що він використовує операцію ділення на двійку, яка виконується доволі "повільно".

Як і вище, дію цього алгоритму продемонструємо спочатку на прикладі запису числа 15 у двійковій системі.

Оскільки число $x_0 = 15$ є непарним, то $b_0 = 1$ (інакше треба покласти $b_0 = 0$). Нехай $x_1 \stackrel{\text{def}}{=} (x_0 - b_0)/2$, тобто $x_1 = 7$. Число $x_1 = 7$ є непарним, тому покладемо $b_1 = 1$ (якщо x_1 є парним, ми покладемо $b_1 = 0$). Продовжуємо аналогічно: $x_2 \stackrel{\text{def}}{=} (x_1 - b_1)/2 = 3$, $b_2 = 1$, $x_3 \stackrel{\text{def}}{=} (x_2 - b_2)/2 = 1$, $b_3 = 1$, $x_4 \stackrel{\text{def}}{=} (x_3 - b_3)/2 = 0$. Алгоритм закінчується, коли чергове x стає рівним 0. У нашому прикладі алгоритм закінчується обчисленням x_4, тому необхідні 4 позиції для того, щоб записати двійкове представлення числа 15: $(15)_{10} = (b_3 b_2 b_1 b_0)_2 = (1111)_2$.

Алгоритм 4. Двійкове представлення (починаємо праворуч)

Вхідні дані: натуральне число k

Вихідні дані: двійкове представлення $k = (b_0 b_1 \dots b_i)_2$;

позначимо $x_0 = k$;

якщо x_0 є непарним числом, то $b_0 = 1$; інакше $b_0 = 0$;

покладемо $x_1 = \dfrac{x_0 - b_0}{2}$; якщо $x_1 = 0$, то STOP.

якщо x_1 є непарним числом, то $b_1 = 1$; інакше $b_1 = 0$;

покладемо $x_2 = \dfrac{x_1 - b_1}{2}$; якщо $x_2 = 0$, то STOP.

якщо x_2 є непарним числом, то $b_2 = 1$; інакше $b_2 = 0$;

покладемо $x_3 = \dfrac{x_2 - b_2}{2}$; якщо $x_3 = 0$, то STOP.

. . . . . .

6. Контрольні питання

1. Чому $\mathcal{C}_{\mathtt{A}} = \mathcal{P}_{\mathtt{A}}$ для будь-якого експоненціального шифру $E_{k,n}$? (стор. 152).

2. Перевірити конгруенції (2). (стор. 153).

3. Пояснити чому комбінації 2709, 2008, 1307, 606 дають однаковий результат при використанні шифру $E_{k,701}$? (стор. 153).

4. Чому модуль мультиплікативного шифру $E_{k,n}$ повинен перевищувати 3333? (стор. 153).

5. Перевірити обчислення в (3). (стор. 154).

6. Довести формулу (4). (стор. 155).

7. Чому $\left(a^{\phi(n)}\right)^t \cdot a \pmod{n} = a \pmod{n}$? (стор. 157).

8. Чому дорівнює $\phi(p)$? (стор. 157).

9. Чому $(a, p) = p$, якщо $(a, p) \neq 1$? (стор. 157).

10. У якому результаті стверджується, що $k^{-1} \pmod{\phi(n)}$ існує, якщо k та $\phi(n)$ є взаємно простими? (стор. 158).

11. Перевірити, що $7 \pmod{30} = 13$. (стор. 158).
12. Згадайте, чому дорівнює $\phi(pq)$? (стор. 158).
13. Доведіть конгруенцію (11). (стор. 159).
14. Пояснити рівності $a^{kj} = up + a$ та $a^{kj} = vq + a$. (стор. 159).
15. Чому $q \mid u$, якщо $up = vq$? (стор. 159).
16. Пояснити правило 2. (стор. 159).
17. Перевірити, чи дійсно ми вже довели лему 4? (стор. 160).
18. Підрахувати $\phi(33)$. (стор. 160).
19. Перевірити рівність $27^{-1} \pmod{20} = 3$. (стор. 160).
20. Чому $7^{-1} \equiv 43 \pmod{60}$ та $11^{-1} \equiv 11 \pmod{60}$? (стор. 160).
21. Чому алгоритм 1 випливає з представлееня (12)? (стор. 161).
22. Чому алгоритм 2 завжди дає $a^k \pmod{n}$? (стор. 163).
23. Пояснити, чому алгоритм 3 є правильним у загальному випадку? (стор. 165).
24. Пояснити, чому алгоритм 4 є правильним у загальному випадку? (стор. 166).

7. Задачі для самостійної роботи

Задача 1. *Знайти два різні натуральні числа $a < 29$ та $b < 29$, для яких $a^2 \equiv b^2 \pmod{29}$. Пояснити, чому не варто використовувати шифр $E_{2,29}$?*

Задача 2. *Перевірити, що $9^k \equiv 0 \pmod{27}$ для будь-якого $k \geq 2$. Чому мультиплікативний шифр з модулем $n = 27$ не варто використовувати?*

Задача 3. *Оскільки $391 = 17 \times 23$, то*

$$\phi(391) = \phi(17)\,\phi(23) = 16 \cdot 22 = 352.$$

Пояснити, що означає рівність $\phi(391) = 352$ з точки зору

a) *теорії чисел,*
b) *мультплікативних шифрів,*
c) *експоненціальних шифрів.*

Задача 4. *Обчислити*

a) $31^{11} \pmod{59}$;
b) $11^{41} \pmod{521}$;
c) $19^{107} \pmod{1249}$.

Задача 5. *Використовуючи конгруенцію* $29 \equiv -2 \pmod{31}$, *обчислити*

a) $29^2 \pmod{31}$; b) $29^5 \pmod{31}$.

Задача 6. *Скільки операцій множення необхідно зробити, щоб обчислити*

a) a^{47}? b) a^{147}?

Задача 7. *Записати* $(2015)_{10}$ *у двійковій системі числення за допомогою*

a) *алгоритму* 3; b) *алгоритму* 4.

Задача 8. *Записати* $(988)_{10}$ *у двійковій системі числення за допомогою*

a) *алгоритму* 3; b) *алгоритму* 4.

Задача 9. *Записати позиції букв тексту* `КІНО` *у десятковій та двійковій системах числення.*

Задача 10. *Записати позиції букв тексту* `ШИФР` *у десятковій та двійковій системах числення.*

Задача 11. *Використовуючи алгоритм 2, зашифрувати текст* `КІНО` *за допомогою експоненціального шифру з параметрами* $k = 2015$ *та* $n = 1000$.

Задача 12. *Використовуючи алгоритм 2, зашифрувати текст* `ШИФР` *за допомогою експоненціального шифру з параметрами* $k = 988$ *та* $n = 51$.

Задача 13. *Знайти* j, *при якому* $a^{7j} \equiv a \pmod{34}$.

Задача 14. *Знайти j, при якому $a^{7j} \equiv a \pmod{523}$. Зважте на те, що 523 є простим числом.*

Задача 15. *За допомогою експоненціального шифру $E_{7,34}$ отримано зашифрований текст* `ЇАС`. *Дешифрувати його (використати обчислення, зроблені у задачі 13).*

Задача 16. *За допомогою експоненціального шифру $E_{7,523}$ отримано зашифрований текст* `131 95 1`. *Дешифрувати його (використати обчислення, зроблені у задачі 14).*

Задача 17. *Нехай $p \equiv 2 \pmod{3}$, де p — просте число. Показати, що $(3, \phi(p)) = 1$. Це означає, що $k = 3$ можна обрати для експоненціального шифру за модулем p. Показати, що показник степеня для дешифрації такого шифру дорівнює $j = \frac{2p-1}{3}$.*

Задача 18. *Нехай $p \equiv 2 \pmod{3}$ та $q \equiv 2 \pmod{3}$, де p та q — прості числа. Покладемо $n = pq$. Показати, що $(3, \phi(n)) = 1$. Це означає, що $k = 3$ можна обрати для експоненціального шифру за модулем n. Знайти показник степеня для дешифрації такого шифру.*

Задача 19. *Нехай $n = p_1 p_2 p_3$, де p_1, p_2, p_3 — три різних простих числа. Нехай k та j є взаємно оберненими за модулем $\phi(n)$. Довести, що $a^{kj} \equiv a \pmod{\phi(n)}$ для всіх цілих a.*

Задача 20. *Використовуючи задачу 19, визначити показник j кореня з k за модулем $\phi(n)$, якщо $k = 7$, $n = p_1 p_2 p_3$, $p_1 = 11$, $p_2 = 13$, $p_3 = 19$.*

Задача 21. *Нехай a та m — натуральні числа, причому $(a, m) = 1$. Довести, що послідовність $r_i = a^i \pmod{m}$, $i \geq 0$, є періодичною.*

Задача 22. *Згідно до задачі 21, послідовність $r_i = a^i \pmod{m}$, $i \geq 0$, є періодичною, якщо $(a, m) = 1$. Найменший період цієї послідовності назвемо порядком числа a за модулем m і позначатимемо $\mathrm{ord}_m(a)$. Нехай натуральне число u є таким, що $a^u \equiv 1 \pmod{m}$. Довести, що $\mathrm{ord}_m(a) \mid u$.*

Задача 23. *Позначення* $\mathrm{ord}_m(a)$ *для натуральних чисел* a *та* m *пояснено у задачі 22. Довести, що якщо* $(a,m)=1$, *то* $\mathrm{ord}_m(a) \mid \phi(m)$.

Задача 24. *Довести, що якщо* $ab \equiv 1 \pmod{m}$, *то* $\mathrm{ord}_m(a) = \mathrm{ord}_m(b)$.

Задача 25. *Позначимо* $e = \mathrm{ord}_m(a)$. *Нехай* k *— натуральне число. Довести, що*

$$\mathrm{ord}_m(a^k) = \frac{e}{(e,k)}.$$

Задача 26. *Натуральне число* α *називається примітивним коренем для модуля* m, *якщо* $(\alpha,m)=1$ *та* $\mathrm{ord}_m(\alpha) = \phi(m)$. *Перевірити, що*

a) 3 *та* 5 *є примітивними коренями для модуля* 7;
b) 2 *є примітивними коренями для модуля* 9.

Довести, що не існує жодного примітивного кореня для модуля 12.

Задача 27. *Нехай* $\mathrm{ord}_m(a) = e$. *Довести, що* $a^i \equiv a^j \pmod{m}$ *тоді і тільки тоді, коли* $i \equiv j \pmod{e}$.

Задача 28. *Нехай* α *— це примітивний корінь для модуля* m (*ми також кажемо, що* m *має примітивний корінь* α). *Позначимо*

$$r_k = \alpha^k \pmod{m}, \qquad 1 \le k \le \phi(m).$$

Довести, що $r_1, \ldots, r_{\phi(m)}$ *— це перестановка чисел, які не перевищують* m *та є взаємно простими з ним.*

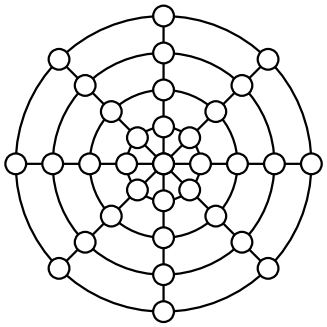
Задача 29. *Довести, що якщо число* m *має примітивний корінь, то воно має* $\phi(\phi(m))$ *примітивних коренів. Зокрема, якщо* m *є простим числом, то воно має* $\phi(m-1)$ *примітивних коренів.*

Задача 30. *Довести, що якщо просте число* $p > 3$ *має примітивний корінь, то воно має парну кількість примітивних коренів.*

Задача 31. *Нехай натуральне число* m *є таким, що* $(a,m)=1$ *та* $\mathrm{ord}_m(a) = m-1$. *Довести, що* m *є простим числом.*

Задача 32. *Довести, що* 2 *не є примітивним коренем для жодного з чисел Ферма* $F_n = 2^{2^n} + 1$, $n \geq 2$.

Задача 33. *На малюнку, наведеному нижче, розташувати числа 1, 2, . . . , 33 в маленьких колах так, щоб суми чисел на усіх більших колах та на діаметрах були б однаковими.*



Цю задачу запропонував китайський математик Янг Ху у книзі "Розвиток стародавніх математичних методів для з'ясування дивних властивостей чисел", виданій у 1275 році. Він розглядав задачі про магічні квадрати, а наведений рисунок був однією з ілюстрацій до його відкриттів. Хоча, звичайно, коло зовсім не квадрат, але магія чисел з зазначеними властивостями присутня!

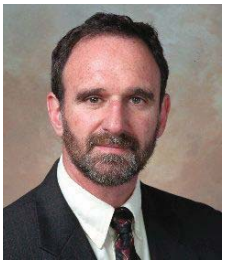
8. БІОГРАФІЇ

Поліґ, *Стефен* (нар. 1953 р.), американський інженер-електрик, працює в Масачусетському технологічному інституті. В середині 1970-х років був аспірантом Мартіна Хеллмана в Стенфордському університеті. Саме тоді він брав участь у розробці експоненціального шифру. Отримані ним формули використовуються також і для обчислення дискретних логарифмів.

С. Поліґ

В 1978 році разом з М. Хеллманом отримав патент "*Метод для експоненціального шифрування*".

Хеллман, *Мартін* (нар. 2.10.1945), американський криптограф. Здобув популярність головним чином завдяки розробці першої асиметричної криптосистеми у співавторстві з Уітфілдом Діффі і Ральфом Меркле у 1976 році.



М. Хеллман

Проте його спільна робота з С. Поліґом стосовно експоненціальних шифрів також добре відома спеціалістам.