Wyznaczmy A_1. Jeśli $e_2 = 0$, to $a_{2\uparrow n} < \varphi(m)$, więc $A_2 = a_{2\uparrow n}$ i wystarczy przyjąć $A_1 = a_1^{A_2} \bmod m$. Z kolei jeśli $e_2 = 1$, to $a_{2\uparrow n} \geqslant A_2 + \varphi(m) \geqslant \log_2 m \geqslant s$, czyli

$$a_{1\uparrow n} = a_1^{a_{2\uparrow n}} = a_1^{(\gamma-1)\varphi(m)+A_2+\varphi(m)} \overset{(***)}{\equiv} a_1^{A_2+\varphi(m)} \pmod m.$$

Zatem w obu przypadkach mamy $A_1 = a_1^{A_2+e_2\varphi(m)} \bmod m$. Z kolei wyznaczyć e_1 można następująco: jeśli $a_1 < 2$, to $e_1 = 0$, a w przeciwnym przypadku możemy wykonać potęgowanie $a_1^{A_2+e_2\varphi(m)}$, w każdej iteracji domnażając jedno a_1 i sprawdzając, czy wynik osiągnął już m (wykonamy co najwyżej $\log_2 m$ takich iteracji). Ostatecznie złożoność czasowa całego algorytmu nie zmienia się.

Pozostaje udowodnić dane wzorem $(***)$ uogólnienie twierdzenia Eulera. Zdefiniujmy ciąg d_i następująco:

$$d_0 = \operatorname{nwd}(a, m), \qquad d_i = \operatorname{nwd}\left(a, \frac{m}{d_0 \cdots d_{i-1}}\right) \quad \text{dla } i \geqslant 1,$$

oraz niech s będzie najmniejszą liczbą, taką że $d_s = 1$. Oznaczmy też $D = d_0 \cdots d_{s-1}$.

Liczba m/D jest liczbą powstałą po usunięciu z m wszystkich czynników pierwszych występujących w a, zatem liczby a i m/D są względnie pierwsze. Z twierdzenia Eulera wynika zatem, że

$$a^{k\cdot\varphi(m/D)} \equiv 1 \pmod{m/D}.$$

Ponadto każda z liczb a/d_i jest całkowita, więc liczba a^s/D również. Mnożąc powyższe równanie przez a^s/D, dostajemy

$$a^{s+k\cdot\varphi(m/D)}/D \equiv a^s/D \pmod{m/D}.$$

Korzystając z faktu, że kongruencja $x \equiv y \pmod w$ jest spełniona wtedy i tylko wtedy, gdy spełniona jest $xD \equiv yD \pmod{wD}$, możemy przemnożyć przez D obie strony i moduł powyższego równania:

$$a^{s+k\cdot\varphi(m/D)} \equiv a^s \pmod m.$$

Liczby D i m/D są względnie pierwsze, zatem z multiplikatywności funkcji φ dostajemy $\varphi(m) = \varphi(m/D)\varphi(D)$. Ponadto wykładniki w rozkładach na czynniki pierwsze liczb $m/(d_0 \cdots d_{i-1})$ zmniejszają się, więc $s \leqslant \log_2 m$. Zatem ostatecznie dostajemy tezę twierdzenia:

$$a^{s+k\cdot\varphi(m)} \equiv a^s \pmod m.$$

Tomasz IDZIASZEK

Odwracamy, obracamy...

Dana jest n-elementowa tablica $a[1\,..\,n]$, którą chcemy odwrócić, czyli spowodować, że jej elementy będą zapisane w kolejności $a[n], a[n-1], \ldots, a[1]$. Najłatwiej to zrobić *w miejscu* (czyli używając jedynie stałej liczby komórek pamięci dla zmiennych pomocniczych), korzystając z instrukcji $\operatorname{swap}(a[i], a[j])$ zamieniającej miejscami wartości dwóch elementów:

for $i := 1$ **to** $\lfloor n/2 \rfloor$ **do**
    $\operatorname{swap}(a[i],\ a[n+1-i])$;

Nietrudno się przekonać, że powyższy kod wywołuje instrukcję zamiany jedynie $\lfloor \frac{n}{2} \rfloor$ razy, co w przypadku odwrócenia tablicy jest wynikiem optymalnym.

A teraz trudniejsze zadanie: chcemy tę samą tablicę obrócić o k komórek w lewo, czyli ustawić jej elementy w kolejności $a[k+1], a[k+2], \ldots, a[n], a[1], \ldots, a[k]$. I tym razem spróbujmy to zrobić, używając jedynie instrukcji zamiany.

Można w tym celu trzykrotnie wywołać omówioną przed chwilą procedurę odwracania tablicy:

$$\operatorname{rev}(a[1\,..\,k]);\quad \operatorname{rev}(a[k+1\,..\,n]);\quad \operatorname{rev}(a[1\,..\,n]);$$

Ten kod wykona $\lfloor \frac{k}{2} \rfloor + \lfloor \frac{n-k}{2} \rfloor + \lfloor \frac{n}{2} \rfloor$ instrukcji zamiany, co w zależności od parzystości liczb n i k da nam n lub $n-1$ instrukcji. Pytanie: czy da się mniej?

Poniższy kod obraca tablicę, dzieląc ją na $\operatorname{nwd}(n,k)$ cykli o długości $n/\operatorname{nwd}(n,k)$ i wykonując obrót każdego z nich niezależnie, do czego potrzebuje $n/\operatorname{nwd}(n,k) - 1$ instrukcji zamiany:

for $i := 1$ **to** $\operatorname{nwd}(n,k)$ **do**
    for $j := 1$ **to** $n/\operatorname{nwd}(n,k) - 1$ **do**
        $\operatorname{swap}(a[i +_n (j-1)\cdot k],\ a[i +_n j \cdot k])$;

Operacja $i +_n j$ oznacza tu $(i + j - 1) \bmod n + 1$. Zatem w tym rozwiązaniu użyjemy w sumie $n - \operatorname{nwd}(n,k)$ instrukcji zamiany. A czy ten wynik da się poprawić?

T.I.