LICZBY

PIERWSZE

Jan Ciurej
Radosław Żak

klasa IV a
Katolicka Szkoła Podstawowa
im. Świętej Rodziny z Nazaretu w Krakowie
ul. Pędzichów 13, 31-152 Kraków

opiekun - mgr Urszula Zacharska
konsultacja informatyczna - mgr inż. Tomasz Łaska

Spis treści:

.

Wstęp

Nasza praca jest o liczbach pierwszych. Wybraliśmy ten temat, ponieważ jest bardzo interesujący. Jest on bardzo obszerny. Dlatego opisaliśmy tu tylko kilka zagadnień. Wyjaśniliśmy czym są te liczby i pokazaliśmy, że jest ich nieskończenie wiele. Zajęliśmy się ich wyszukiwaniem i zastosowaniem. Najpierw przedstawiliśmy wzory na wyliczanie tych liczb podane przez znanych matematyków. Następnie pokazaliśmy ich zastosowanie do szyfrowania numerów kart kredytowych. W naszej pracy liczby naturalne liczyliśmy od 0. Każdy z nas zajął się innym rozdziałem. Rozdziały 1 i 3 napisał Radosław Żak, a rozdziały 2 i 4 Jan Ciurej.

Zajmowanie się tematem liczb pierwszych sprawiło nam wiele przyjemności. Mamy nadzieję, że tą pracą zachęcimy innych uczniów do zainteresowania się tymi ciekawymi liczbami.

1. Co to jest liczba pierwsza?

Liczba pierwsza to liczba naturalna, która posiada dokładnie dwa dzielniki: 1 i samą siebie. Przykładami liczb pierwszych są.: 2, 3, 5, 7, 11, 13. Współtworzą one także inne liczby naturalne, nie będące liczbami pierwszymi, tzn. każdą liczbę naturalną większą od 1, która nie jest liczbą pierwszą można przedstawić w postaci iloczynu liczb pierwszych np.:

$$130 = 2 \cdot 5 \cdot 13$$

$$231 = 3 \cdot 7 \cdot 11$$

Liczbą pierwszą nie jest 1, ponieważ ma tylko jeden dzielnik oraz 0, które ma ich nieskończenie wiele.

Liczb pierwszych jest nieskończenie wiele, co zostało udowodnione przez żyjącego w IV w. p.n.e. Euklidesa. Założył on, że ilość liczb pierwszych jest skończona. Można je zatem pomnożyć i do wyniku dodać 1.

Ilość kolejnych liczb pierwszych	Wynik	Liczba pierwsza lub iloczyn liczb pierwszych
1	$2 + 1 = 3$	3
2	$2 \cdot 3 + 1 = 7$	7
3	$2 \cdot 3 \cdot 5 + 1 = 31$	31
4	$2 \cdot 3 \cdot 5 \cdot 7 + 1 = 211$	211
5	$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 + 1 = 2311$	2311
6	$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 + 1 = 30031$	$59 \cdot 509$

Żaden wynik nie będzie liczbą podzielną przez liczby pierwsze dotąd nam znane, ponieważ w wyniku dzielenia zawsze otrzymamy resztę 1, dlatego będzie podzielny przez nową liczbę pierwszą.

2. Próby wyznaczania liczb pierwszych.

Od dawna próbowano szukać liczb pierwszych za pomocą rożnych wzorów, niestety za każdym razem okazywało się, że wzór zawodzi. W tym rozdziale pokażemy tyko kilka takich prób. Symbol Lp będzie oznaczał liczbę pierwszą, n – liczbę naturalną.

1. **Leonhard E U L E R** próbował wyznaczać liczby pierwsze za pomocą wielu wzorów, m.in. poniższego:

   $$Lp = n^2 + n + 41$$

   Jednak dla n = 40 wzór okazał się nieprawidłowy, bo

   $$40^2 + 40 + 41 = 1681 = 41^2$$

2. **Adrian Marie L E G E N D R E** wskazał inny wzór:

   $$Lp = 2n^2 + 29$$

   który również okazał się niewłaściwy, bo dla n = 29 otrzymujemy

   $$1711 = 29 \cdot 59$$

3. **E S S C O T –** amerykański matematyk przekształcił wzór Eulera na taki:

   $$Lp = n^2 – 79n + 1601$$

   Wzór ten zawodzi przy wartości n = 80, bo

   $$80^2 – 79 \cdot 80 + 1601= 1681 = 41^2$$

4. **Jest także wzór:**

   $$Lp_m = (2^m + 1) / 3$$

   Daje on wartości liczb pierwszych gdy za m podstawiamy liczby nieparzyste, ale wzór zawodzi gdy za m podstawimy 37.

   $Lp_{37} = (2^{37} + 1) / 3 = 45812984491 = 1777 \cdot 25781083.$

5. **Ludzie nadal nie wynaleźli ogólnego wzoru na wyszukiwanie liczb pierwszych.** Największa liczba pierwsza jaką znaleziono do tej pory to $2^{57885161}$-1, która ma 17425170 cyfr.

3. Zastosowanie liczb pierwszych.

Istnieje wiele zastosowań liczb pierwszych. Tutaj opiszemy jedno z nich, lecz najpierw trzeba poznać arytmetykę modularną wynalezioną przez Carla Gaussa. Wybieramy jakąś liczbę i nazywamy ją modułem. Wynikiem działania modulo będzie reszta z dzielenia wyniku przez moduł. Arytmetykę modularną stosuje się powszechnie w obliczeniach zegarowych. Na przykład jeśli jest godzina 10.00 to za 17 godzin nie będzie 27.00, tylko 3.00, ponieważ
(10 + 17) : 24 = 1 reszty 3.

W 1977 roku odkryto kod RSA (nazwa pochodzi od pierwszych liter nazwisk twórców: Rona Rivesta, Adi Shamira i Leonarda Adlemana) oparty na liczbach pierwszych. Wykorzystuje się go między innymi do zakupów w sklepach internetowych. Działa on tak:

- Na stronie internetowej podane są dwie liczby: moduł obliczeń oznaczony literą N, który jest iloczynem dwóch liczb pierwszych p i q oraz kod szyfrujący oznaczony literą E.

- Karta kredytowa oznaczona jest literą C.
  Dla przykładu wybieramy małe liczby następująco:
  p = 17, q = 11, czyli N = 17 · 11 = 187, E = 9, C = 7.

- Komputer wykonuje następujące obliczenia:
  $$C^E \pmod{N} = F$$
  Czyli F jest resztą z dzielenia C^E przez N. Tym sposobem nr karty kredytowej został zaszyfrowany, dlatego teraz można wysyłać zaszyfrowaną wiadomość F.
  W naszym przykładzie $F = 7^9 \pmod{187} = 129$

- Aby odszyfrować wiadomość należy znaleźć liczbę deszyfrującą D. Znalezienie tej liczby wiąże się ze znalezieniem czynników pierwszych liczby N, ponieważ:

$$D \cdot E \pmod{(p-1)(q-1)} = 1$$

Wracając do przykładu – nasze D = 89, ponieważ $89 \cdot 9 \pmod{16 \cdot 10} = 1$

- Teraz można obliczyć C z poniższego wzoru:

$$C = F^D \pmod{N}$$

Podstawiając do wzoru otrzymujemy $129^{89} \pmod{187} = 7$

Kod RSA bardzo trudno złamać, ponieważ trzeba rozłożyć na czynniki pierwsze liczbę N. My mieliśmy ułatwione zadanie, bo znaliśmy p i q. Ale ktoś, kto nie zna tych liczb, nie będzie miał tak łatwo, tym bardziej, że w praktyce używa się bardzo dużych liczb (mających ponad 100 cyfr).

4. Sprawdzanie czy dana liczba jest liczbą pierwszą - program.

Oto prosty program w języku Python, który sprawdza czy dana liczba jest liczbą pierwszą. Program działa bardzo długo dla dużych liczb.

Funkcja **lop** musi być uruchomiona w języku Python.

W nawiasie należy podać zmienną **lpp**, która podlega sprawdzeniu.

Funkcja wraca True lub False. True - jeśli **lpp** jest liczbą pierwszą,

False - jeśli **lpp** nie jest liczbą pierwszą.

```
def lop(lpp):              #definiuje funkcję LOP
 if lpp==0 or lpp==1:      #jeżeli lpp jest równe 0 lub 1 to
     return False          #napisz False i nie rób dalej programu
 for i in range(lpp):      #rozpocznij pętlę (i przyjmuje wartości
                           od 0 do lpp-1)
   if i==0 or i==1:        #jeżeli i jest równe 0 lub 1 to przejdź
                           dalej
       continue            #czyli kontynuuj pętlę dla następnych i
    elif lpp%i==0:         #jeśli lpp mod i = 0, czyli lpp jest
                           podzielne przez i bez reszty (znak % to
                           dzielenie modulo czyli reszta z
                           dzielenia)
     return False          #to napisz False nie rób dalej programu
                           #pętla sprawdza podzielność lpp przez
                           wszystkie liczby od 2 do lpp-1, jeżeli
                           nie znajdzie to
return True                #napisz True nie rób dalej programu tzn.
                           lpp jest liczbą pierwszą
```

Literatura:

1. Szczepan Jeleński, *Śladami Pitagorasa*, Państwowe Zakłady Wydawnictw Szkolnych, Warszawa 1953.
2. Ian Stewart, *Gabinet matematycznych zagadek*, Wydawnictwo Literackie, Kraków 2011.
3. Marcus du Sautory, *Poker z Pitagorasem,* Carta blanca, Warszawa 2012.
4. http://primes.utm.edu/largest.html