Architektura komputerów 2 - projekt etap 2

Adam Karczewski, Michał Skrzęta
prowadzący: dr inż. Dominik Żelazny

16 maja 2018

1 Zaimplementowane algorytmy - etap 1

1.1 Sito Eratostenesa

Użytkownik programu był proszony o wprowadzenie przedziału liczb, który musiał się zawierać w przedziale [10, 2560]. Program sprawdza poprawność wczytanych danych. Algorytm sita Eratostenesa polega na wykreślaniu liczb do momentu, gdy jakaś liczba z sita (której wielokrotności wykreślamy) będzie większa niż pierwiastek z wprowadzonej liczby (koniec przedziału liczbowego). W celu wyliczenia tego pierwiastka została użyta funkcja z języka C. W kodzie programu znajduje się funkcja o nazwie sito. W niej początkowo został wypełniony odpowiednio bufor danych służący za sito. W algorytmie sita Eratostenesa wybierana jest najmniejsza liczba z przedziału (2), a potem są wykreślane jej wszystkie wielokrotności. Kolejno wybierana jest następna liczba niewykreślona i jej wielokrotności są również wykreślane. W programie wielokrotności liczb w buforze były zastępowane zerami. Potem program przechodził przez sito, jeśli natknął się na liczbę, która nie była zerem - oznaczało to, że jest pierwsza. Liczby pierwsze z sita były zapisywane do pliku tekstowego, po odpowiedniej konwersji na kod ASCII i dodaniu znaku spacji. Jeśli liczby zostały zapisane prawidłowo do pliku, był wyświetlany odpowiedni komunikat, a program potem kończył swoje działanie.

1.2 Test Fermata

Test pierwszości Fermata to probabilistyczny test umożliwiający sprawdzenie, czy dana liczba jest złożona, czy prawdopodobnie pierwsza. Jest jednym z najprostszych testów pierwszości i pomimo swoich wad jest wykorzystywany w algorytmach szyfrowania PGP.

Test opiera się na małym twierdzeniu Fermata, które ma postać:

Jeżeli liczba p jest liczbą pierwszą i $1 \leqslant$ a $<$ p to,

$$a^{p-1} \equiv 1(\text{mod p})$$

Aby stwierdzić, czy p jest pierwsza, można wybrać kilka losowych wartości a i sprawdzić, czy ta równość jest dla nich spełniona. Jeśli dla którejkolwiek z nich nie jest, to na pewno p jest liczbą złożoną. Jeśli wszystkie ją spełniają, p jest prawdopodobnie liczbą pierwszą albo pseudopierwszą.

Zaimplementowany algorytm pobiera od użytkownika liczbę do testów i ilość iteracji. Do losowania wartości użyto funkcji rand(). Potęgowanie modularne jest realizowane przy użyciu funkcji szybkiego potęgowania modularnego.

1.2.1 Szybkie potęgowanie modularne

Korzystając z własności kongruencji oraz postaci binarnej liczby możliwe jest szybkie obliczenie wartości wyrażenia a^p mod k. Wykorzystanie tego algorytmu jest konieczne, jeżeli chcemy operować na dużych liczbach- klasycznie potęgowanie zajmuje wtedy bardzo dużo czasu oraz występuje problem z przepełnieniem rozmiaru zmiennych.

2 Zaimplementowane algorytmy - etap 2

2.1 Test Millera - Rabina

Został opracowany w roku 1975 przez Michaela O. Rabina na podstawie prac Gary'ego L. Millera. Udostępnia on szybką metodę sprawdzania pierwszości liczby z możliwością kontrolowania poziomu prawdopodobieństwa popełnienia błędu – jest to zatem metoda probabilistyczna. Test Millera-Rabina oparty jest na następującym twierdzeniu:

Niech p będzie nieparzystą liczbą pierwszą zapisaną jako p = 1 + 2^sd, gdzie d jest nieparzyste. Wtedy dla dowolnej liczby naturalnej $a \in < 2, p-2 >$ ciąg Millera-Rabina:

$$a^d, a^{2d}, a^{4d}, ..., a^{2^{s-1}d}, a^{2^s d} (\text{mod p})$$

kończy się liczbą 1. Co więcej, jeśli a^d nie przystaje modulo p do 1, to wyraz ciągu Millera-Rabina bezpośrednio poprzedzający 1 jest równy p - 1.

Jeśli liczba p przejdzie test, to jest albo pierwsza, albo silnie pseudopierwsza przy podstawie a. Test Millera-Rabina daje złe wyniki (p złożona) dla co najwyżej 1/4 baz a <p. Zatem dla jednego przebiegu prawdopodobieństwo błędu wynosi 1/4.

W algorytmie testu Millera-Rabina wykorzystujemy procedury mnożenia i potęgowania modulo. Test ten wykorzystują obecnie prawie wszystkie systemy kryptografii publicznej do testowania pierwszości dużych liczb potrzebnych przy generacji kluczy szyfrujących/deszyfrujących.

2.2 Test Solovay - Strassena

Test pierwszości opracowany przez Roberta M. Solovaya i Volkera Strassena. Jest to test probabilistyczny, który określa czy dana liczba jest liczbą złożoną, czy prawdopodobnie pierwszą. W większości zastosowań test ten został wyparty przez test Millera-Rabina, lecz ma wysoki historyczny wkład w pokazaniu praktycznego wykorzystania RSA. Test korzysta z poniższego twierdzenia:

a jest świadkiem Eulera dla złożoności liczby n jeśli:

$$a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) (\text{mod n})$$

gdzie $\left(\frac{a}{n}\right)$ to symbol Jacobiego

Należy wybierać wartości a losowo i sprawdzać czy liczba ta jest świadkiem Eulera dla n. Jeśli zostanie znaleziony taki świadek Eulera, czyli takie a, które nie spełnia kongruencji, to oznacza, że n nie jest liczbą pierwszą. Użyteczność tego testu wynika z faktu, że dla każdej nieparzystej liczby złożonej n przynajmniej połowa ze wszystkich a jest świadkiem Eulera.