Algorytmy teorioliczbowe i RSA

wszelkie prawa zastrzeżone
zakaz kopiowania, publikowania i przechowywania
all rights reserved
no copying, publishing or storing

Maciej Hojda

1 Zadanie nr 1 – rozkład na czynniki pierwsze

Zaimplementuj funkcję, która zwróci listę czynników pierwszych zadanej liczby naturalnej n. Zrób to rekurencyjnie, sprawdzając podzielność liczby przez kolejne liczby naturalne (aż do $\lfloor\sqrt{n}\rfloor$) – rekurencja pojawia się, gdy liczba jest podzielna – wtedy uruchamiamy algorytm na jej dzielnikach.

Wejście: liczba naturalna n.
Wyjście: lista dzielników liczby n.

2 Zadanie nr 2 – sito Eratostenesa

Zaimplementuj sito Eratostenesa, aby wyznaczyć zbiór liczb pierwszych nie większych od zadanego p.

Algorytm 1 Sito Eratostenesa – `SERA`(p)

```
- wejscie: liczba naturalna p > 1.
1 niech x ≜ [x_n]_{n∈{2,3,...,p}} := [1]_{n∈{2,3,...,p}}
2 dla n := 2 do ⌊√p⌋
3    jeśli x_n = 1, to
4       dla j := 2 do ⌊p/n⌋
5          x_{n·j} := 0
6 zwróć x
- wyjście: wektor x dla którego, jeśli x_j = 1, to liczba j jest piewsza.
```

Wejście: liczba naturalna p.
Wyjście: lista liczb pierwszych nie większych od p.

3 Zadanie nr 3 – największy wspólny dzielnik

3.1 Wyszukiwanie

Zaimplementuj funkcję szukającą największego wspólnego dzielnika dwóch liczb. Zrób to na dwa sposoby.

- Z wykorzystaniem rozkładu na czynniki pierwsze `RNWD`(a, b).
- Z wykorzystaniem algorytmu Euklidesa `ENWD`(a, b).

Wejście: dwie liczby a, b.
Wyjście: $NWD(a, b)$ uzyskane na dwa sposoby.

3.2 Testy wydajności

Przygotuj procedurę testową do sprawdzenia czasu działania obu algorytmów.

Uruchamiaj `RNWD`(n, q) i `ENWD`(n, q) dla zadanej liczby n i dla kolejnych liczb naturalnych q do pewnego zadanego m. Czasy działania obu algorytmów wyświetl na jednym wykresie.

Wejście: dwie liczby n, m.
Wyjście: wykres czasu działania dwóch algorytmów.

4 Zadanie nr 4 – probabilistyczne testy piewszości

Zaimplementuj dwa algorytmy testowania pierwszości liczb

- test Fermata,
- test Millera-Rabina.

Wykorzystaj szybki algorytm potęgowania modulo.

Wejście: liczba potencjalnie pierwsza p.
Wyjście: rezultat testu pierwszości ww. algorytmami.

5 Zadanie nr 5 – RSA

Zaimplementuj algorytm RSA. Zaszyfruj nim (i odszyfruj) przykładowe teksty. W razie potrzeby, automatycznie dziel tekst na mniejsze, osobno szyfrowane części.

Wejście: tekst do zaszyfrowania/odszyfrowania.
Wyjście: zaszyfrowany/odszyfrowany tekst; klucz prywatny i klucz publiczny.