Projekt Zespołowy

Etap projektu – projektowanie rozwiązania na zadaną architekturę

Autorzy:
Biernacka Kamila
Kania Dominik
Leśniak Mateusz
Maziarz Wojciech

kwiecień 2021

Streszczenie

Poniższe sprawozdanie jest wynikiem naszej pracy na drugim etapie projektu zespołowego z implementacji metody indeksu w architekturach GPU. Przedstawimy w nim przygotowane przez nas projekty i rysunki koncepcyjne wymaganych do zaimplementowania algorytmów.

Spis treści

1 Mnożenie modularne dużych liczb

2 Poszukiwanie relacji i faktoryzacja w bazie

2.1 Szybkie potęgowanie modularne

2.2 Fakoryzacja w bazie

2.3 Budowa relacji

3 Eliminacja Gaussa w pierścieniu $\mathbb{Z}_{p-1}$

3.1 Algorytm Euklidesa

Poniższy algorytm wykorzystuje algorytm Euklidesa do sprawdzenia, czy podane na wejściu dwie liczby są względnie pierwsze.

Algorithm 1: Algorytm Euklidesa

Input: a, b - liczby naturalne
Output: True, jeśli $\gcd(a, b) == 1$, False w przeciwnym przypadku.

```
while b ≠ 0 do
    temp := b
    b := a mod b
    a := temp
if a == 1 then
    output := True
else
    output := False
```

Result: output

3.2 Rozszerzony algorytm Euklidesa

Tożsamość Bezout mówi, że liczby a i p są względnie pierwsze i wtedy i tylko wtedy, gdy istnieją takie liczby s i t, że

$$ps + at = 1,$$

Wówczas po zredukowaniu tej równości modulo p otrzymujemy

$$at \equiv 1 \bmod p,$$

czyli t jest elementem odwrotnym a w pierścieniu Z_p.

Poniższy algorytm odnajduje element odwrotny do elementu a w pierścieniu Z_p.

Algorithm 2: Rozszerzony Algorytm Euklidesa

Input: a, p - liczby naturalne

Output: $x = a^{-1} \bmod p$ lub informacja, że taka liczba nie istnieje

```
1  u ← 1, w ← a, x ← 0, ← p
2  while w ≠ 0 do
3  |   if w < z then
4  |   |   swap(u, x)
5  |   |   swap(w, z)
6  |   q ← w/z − (w%z)
7  |   u ← u − q · x  w ← w − q · z
8  if z ≠ 1 then
9  |   Result None
10 if x < 0 then
11 |   x ← x + p
```

Result: x