Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

Лабораторна робота №1

Вибір та реалізація базових фреймворків та бібліотек

Виконали: студенти гр. ФІ-32мн,
Костюк К.М., Панасюк Є.С, Пелешенко Л.І.

Перевірив:
Кудін А.М

Київ – 2023

Мета лабораторної роботи:
Вибір базових бібліотек/сервісів для подальшої реалізації криптосистеми.
Постановка задачі:
Дослідження алгоритмів реалізації арифметичних операцій над великими (багато розрядними) числами над скінченими полями та групами з точки зору їх ефективності за часом та пам’яттю для різних програмно-апаратних середовищ. Дослідити бібліотеки багатослівної арифметики.
Підгрупа 1А. Бібліотеки багаторозрядної арифметики, вбудовані в програмні платформи C++/C# (BigInteger), Java (BigInt) та Python (обрати одну з них) для процесорів із 32-розрядною архітектурою та обсягомоперативної пам’ятідо 8 ГБ (робочі станції).

Огляд BigInt у Java для 32-розрядних систем

Основні функції:

1. Створення BigInt об'єктів:

```
import java.math.BigInteger;

BigInteger bigInt = new BigInteger("12345678901234567890123456789O");
```

   BigInt дозволяє створювати об'єкти для представлення великих цілих чисел, які не поміщаються в звичайні 32-бітні цілі числа.
2. Арифметичні операції:
   Додавання, віднімання, множення та ділення великих цілих чисел.

```
BigInteger result = bigInt1.add(bigInt2);
BigInteger result = bigInt1.subtract(bigInt2);
BigInteger result = bigInt1.multiply(bigInt2);
BigInteger result = bigInt1.divide(bigInt2);
```

3. Порівняння:
   Методи для порівняння BigInt об'єктів.

```
boolean isEqual = bigInt1.equals(bigInt2);
boolean isGreater = bigInt1.compareTo(bigInt2) > 0;
```

4. Степінь та модуль:
   Розрахунок степенів та взяття модуля.

```
BigInteger power = bigInt.pow(3);
BigInteger modResult = bigInt.mod(bigInt2);
```

Складність операцій:

1. Арифметичні операції:
   Складність додавання, віднімання та множення має порядок O(n), де n - кількість бітів у великих числах.
2. Складність ділення - O(n^2), де n - кількість бітів.
3. Порівняння:
   Складність порівняння - O(n), де n - кількість бітів.
4. Степінь та модуль:
   a. Складність обчислення степеня - O(k * log n), де k - поточний експонент, n - кількість біт в числі.
   b. Складність обчислення модуля - O(n^2), де n - кількість біт в числі.

Обмеження та оптимізація для 32-розрядних систем:

1. Максимальне значення:
   Хоча BigInt не обмежується обсягом пам'яті, максимальне значення може бути обмежене доступною кількістю бітів у 32-розрядній системі. Оптимально використовувати значення, які не перевищують максимального значення 32-бітних цілих чисел.
2. Ефективне використання пам'яті:
   BigInt може використовувати значну кількість пам'яті для збереження великих чисел. Розробники можуть оптимізувати використання пам'яті, обираючи оптимальний тип об'єкта та обмежуючи точність до необхідного рівня.
3. Кешування результатів:
   Для оптимізації використання обчислювальних ресурсів, можливе кешування результатів повторюваних обчислень.
4. Розподілений обчислення:
   У випадках великих обчислень може бути корисним розподілення обчислень на багатоядерних системах.
5. Забезпечення безпеки:
   Великі числа можуть іноді некоректно оброблятися через обмеженнями 32-розрядної архітектури, що може призвести до переповнень або некоректних результатів. Таке може трапитися, якщо число буде занадто велике і буде перевищувати 18446744073709551615 по кількості бітів

Заключення:

Бібліотека BigInt в Java надає потужні можливості для роботи з великими цілими числами в обмежених умовах 32-розрядної архітектури та обсягу оперативної пам'яті. Розробники повинні бути обережні при виборі та

використанні цієї бібліотеки для оптимальності та уникнення можливих обмежень.

Цей приклад демонструє основні арифметичні операції, порівняння, обчислення ступеня та модуля з використанням бібліотеки BigInteger в Java:

```
import java.math.BigInteger;

public class BigIntExample {
  public static void main(String[] args) {
    // Створення об'єктів BigInt
    BigInteger num1 = new BigInteger("12345678901234567890123456789O");
    BigInteger num2 = new BigInteger("98765432109876543210987654321O");

    // Арифметичні операції
    BigInteger sum = num1.add(num2);
    BigInteger difference = num1.subtract(num2);
    BigInteger product = num1.multiply(num2);
    BigInteger quotient = num1.divide(num2);

    // Порівняння
    boolean isEqual = num1.equals(num2);
    boolean isGreater = num1.compareTo(num2) > 0;

    // Степінь та модуль
    BigInteger power = num1.pow(3);
    BigInteger modResult = num1.mod(num2);

    // Вивід результатів
    System.out.println("Сума: " + sum);
    System.out.println("Різниця: " + difference);
    System.out.println("Добуток: " + product);
    System.out.println("Частка: " + quotient);
    System.out.println("Чи рівні: " + isEqual);
    System.out.println("Чи перше число більше: " + isGreater);
    System.out.println("Степінь: " + power);
    System.out.println("Модуль: " + modResult);
  }
}
```

Вивід програми:
Сума: 11111111101111111111011111111100
Різниця: -864197532086419753208641975320
Добуток:
121932631137021795226185032733622923332237463801111263526900
Частка: 0
Чи рівні: false
Чи перше число більше: false
Степінь:
188167637235365777254671604058964172625747722984940942620769379772
2198701224860897069000
Модуль: 1234567890123456789012345678 90