

Ошибка с бесконечным нахождением НОД при аргументе INT.MIN_VALUE.

Краткое описание ошибки:

Ошибка была обнаружена при тестировании граничных значений. Насколько я понимаю, данная проблема происходит только при работе с Java. Дело в том, что Java не превращает INT.MIN_VALUE в положительное число, что в дальнейшем приводит к бесконечному циклу нахождения НОД.

Исходный код:

```
public class GCD
{
    public int gcd(int x, int y)
    {
        int t;

        if(x < 0) x = -x;
        if(y < 0) y = -y;

        while(y != 0)
        {
            if(y > x)
            {
                x = y-x;
                y = y-x;
                x = x+y;
            }

            if(y == 0) return x;

            t = y;
            y = x%y;
            x = t;
        }
        return x;
    }
}
```

Данные, на которых наблюдается некорректное поведение:

Ошибка возникает, если один из аргументов в методе gcd() является INT.MIN_VALUE.

Полученное значение, ожидаемое значение:

```
@Test
void gcdMaxMin() {
    assertEquals( expected: 1, getGcd(Integer.MIN_VALUE, Integer.MAX_VALUE));
}
```

Программа уходит в бесконечный цикл и ничего не возвращает. Ожидаемое значение – 1.

Код после исправления:

Насколько я понимаю - единственное решение проблемы – это конвертация в Long, а потом обратно в Int.

```
public class GCD {
    public int gcd(int x, int y) {
        long t;
        long _x = x;
        long _y = y;

        // if (x < 0) x = -x;
        // if (y < 0) y = -y;

        if (x < 0) { _x = -1L * x; }
        if (y < 0) { _y = -1L * y; }

        while (_y != 0) {
            if (_y > _x) {
                _x = _y - _x;
                _y = _y - _x;
                _x = _x + _y;
            }

            if (_y == 0) break;

            t = _y;
            _y = _x % _y;
            _x = t;
        }

        return (int) _x;
    }
}

void gcdMaxMin() {
    assertEquals( expected: 1, getGcd(Integer.MIN_VALUE, Integer.MAX_VALUE));
}
```

Tests passed: 1 of 1 test

"C:\Program Files\Java\jdk-11.0.10\bin\java.exe" ...