29. Divide Two Integers

Running time of `recursiveDivide(N,M)` : $O((logN)^2)$.
Corner cases: $(-2^{31}, 1), (2^{31} - 1, 1)$...

```
public int divide(int dividend, int divisor) {
        if(divisor == 0){
            return Integer.MAX_VALUE;
        }
        if(dividend == 0){
            return 0;
        }

        int PosNeg = 0;
        if(divisor < 0){
            PosNeg = 1 - PosNeg;
        }
        if(dividend < 0){
            PosNeg = 1 - PosNeg;
        }
        long result = 0;
        long tmpDividend = Math.abs((long)dividend);
        long tmpDivisor = Math.abs((long)divisor);

        result = recursiveDivide(tmpDividend,tmpDivisor);

        if(result > Integer.MAX_VALUE){
            if(PosNeg == 1){
                result = Integer.MIN_VALUE;
            }
            else{
                result = Integer.MAX_VALUE;
            }
        }
        else{
            if(PosNeg == 1){
                result = -result;
            }
        }
        return (int)result;

    }
    private long recursiveDivide(long tmpDividend, long tmpDivisor){
        if(tmpDividend < tmpDivisor){
            return 0;
        }
        long expand = tmpDivisor;
        long result = 1;
        while(expand + expand <= tmpDividend){
            expand += expand;
            result += result;
        }
```

```
        return result + recursiveDivide(tmpDividend - expand, tmpDivisor);
    }
```