|  |  |
| --- | --- |
| **字段摘要** | |
| static [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**ONE**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#ONE)            BigInteger 的常量 1。 |
| static [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**TEN**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#TEN)            BigInteger 的常量 10。 |
| static [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**ZERO**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#ZERO)            BigInteger 的常量 0。 |

|  |  |
| --- | --- |
| **方法摘要** | |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**abs**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#abs())()            返回其值是此 BigInteger 的绝对值的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**add**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#add(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回其值为 (this + val) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**and**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#and(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回其值为 (this & val) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**andNot**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#andNot(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) val)            返回其值为 (this & ~val) 的 BigInteger。 |
| int | [**bitCount**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#bitCount())()            返回此 BigInteger 的二进制补码表示形式中与符号不同的位的数量。 |
| int | [**bitLength**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#bitLength())()            返回此 BigInteger 的最小的二进制补码表示形式的位数，*不包括* 符号位。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**clearBit**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#clearBit(int))(int n)            返回其值与清除了指定位的此 BigInteger 等效的 BigInteger。 |
| int | [**compareTo**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#compareTo(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) val)            将此 BigInteger 与指定的 BigInteger 进行比较。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**divide**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#divide(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回其值为 (this / val) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类)[] | [**divideAndRemainder**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#divideAndRemainder(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) val)            返回包含 (this / val) 后跟 (this % val) 的两个 BigInteger 的数组。 |
| double | [**doubleValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#doubleValue())()            将此 BigInteger 转换为 double。 |
| boolean | [**equals**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#equals(java.lang.Object))([Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) x)            比较此 BigInteger 与指定的 Object 的相等性。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**flipBit**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#flipBit(int))(int n)            返回其值与对此 BigInteger 进行指定位翻转后的值等效的 BigInteger。 |
| float | [**floatValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#floatValue())()            将此 BigInteger 转换为 float。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**gcd**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#gcd(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) val)            返回一个 BigInteger，其值是 abs(this) 和 abs(val) 的最大公约数。 |
| int | [**getLowestSetBit**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#getLowestSetBit())()            返回此 BigInteger 最右端（最低位）1 比特的索引（即从此字节的右端开始到本字节中最右端 1 比特之间的 0 比特的位数）。 |
| int | [**hashCode**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#hashCode())()            返回此 BigInteger 的哈希码。 |
| int | [**intValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#intValue())()            将此 BigInteger 转换为 int。 |
| boolean | [**isProbablePrime**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#isProbablePrime(int))(int certainty)            如果此 BigInteger 可能为素数，则返回 true，如果它一定为合数，则返回 false。 |
| long | [**longValue**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#longValue())()            将此 BigInteger 转换为 long。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**max**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#max(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回此 BigInteger 和 val 的最大值。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**min**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#min(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回此 BigInteger 和 val 的最小值。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**mod**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#mod(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) m)            返回其值为 (this mod m) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**modInverse**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#modInverse(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) m)            返回其值为 (this-1 mod m) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**modPow**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#modPow(java.math.BigInteger,%20java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) exponent, [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) m)            返回其值为 (thisexponent mod m) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**multiply**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#multiply(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回其值为 (this \* val) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**negate**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#negate())()            返回其值是 (-this) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**nextProbablePrime**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#nextProbablePrime())()            返回大于此 BigInteger 的可能为素数的第一个整数。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**not**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#not())()            返回其值为 (~this) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**or**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#or(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回其值为 (this | val) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**pow**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#pow(int))(int exponent)            返回其值为 (thisexponent) 的 BigInteger。 |
| static [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**probablePrime**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#probablePrime(int,%20java.util.Random))(int bitLength, [Random](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/Random.html) rnd)            返回有可能是素数的、具有指定长度的正 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**remainder**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#remainder(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回其值为 (this % val) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**setBit**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#setBit(int))(int n)            返回其值与设置了指定位的此 BigInteger 等效的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**shiftLeft**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#shiftLeft(int))(int n)            返回其值为 (this << n) 的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**shiftRight**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#shiftRight(int))(int n)            返回其值为 (this >> n) 的 BigInteger。 |
| int | [**signum**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#signum())()            返回此 BigInteger 的正负号函数。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**subtract**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#subtract(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)            返回其值为 (this - val) 的 BigInteger。 |
| boolean | [**testBit**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#testBit(int))(int n)            当且仅当设置了指定的位时，返回 true。 |
| byte[] | [**toByteArray**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#toByteArray())()            返回一个 byte 数组，该数组包含此 BigInteger 的二进制补码表示形式。 |
| [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) | [**toString**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#toString())()            返回此 BigInteger 的十进制字符串表示形式。 |
| [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) | [**toString**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#toString(int))(int radix)            返回此 BigInteger 的给定基数的字符串表示形式。 |
| static [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**valueOf**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#valueOf(long))(long val)            返回其值等于指定 long 的值的 BigInteger。 |
| [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) | [**xor**](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#xor(java.math.BigInteger))([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) val)            返回其值为 (this ^ val) 的 BigInteger。 |

 

|  |
| --- |
| **方法详细信息** |

**probablePrime**

public static [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **probablePrime**(int bitLength,

[Random](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/Random.html) rnd)

返回有可能是素数的、具有指定长度的正 BigInteger。此方法返回的 BigInteger 是合数的概率不超出 2 -100。

**参数：**

bitLength - 返回的 BigInteger 的 bitLength。

rnd - 随机比特源，用这些随机比特选择用来进行质数测试的候选数。

**返回：**

可能是素数的 bitLength 个位的 BigInteger

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - bitLength < 2。

**从以下版本开始：**

1.4

**另请参见：**

[bitLength()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#bitLength())

**nextProbablePrime**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **nextProbablePrime**()

返回大于此 BigInteger 的可能为素数的第一个整数。此方法返回的数是合数的概率不超出 2 -100。此方法在执行以下搜索时将始终不会跳过素数：如果它返回 p，则不存在 this < q < p 的素数 q。

**返回：**

返回大于此 BigInteger 的可能为素数的第一个整数。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - this < 0。

**从以下版本开始：**

1.5

**valueOf**

public static [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **valueOf**(long val)

返回其值等于指定 long 的值的 BigInteger。提供的此“静态工厂方法”优先于 ( long) 构造方法，因为前者允许重用经常使用的 BigInteger。

**参数：**

val - 要返回的 BigInteger 的值。

**返回：**

具有指定值的 BigInteger。

**add**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **add**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this + val) 的 BigInteger。

**参数：**

val - 将添加到此 BigInteger 中的值。

**返回：**

this + val

**subtract**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **subtract**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this - val) 的 BigInteger。

**参数：**

val - 从此 BigInteger 中减去的值。

**返回：**

this - val

**multiply**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **multiply**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this \* val) 的 BigInteger。

**参数：**

val - 要乘以此 BigInteger 的值。

**返回：**

this \* val

**divide**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **divide**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this / val) 的 BigInteger。

**参数：**

val - 此 BigInteger 要除以的值。

**返回：**

this / val

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - val==0

**divideAndRemainder**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html)[] **divideAndRemainder**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回包含 (this / val) 后跟 (this % val) 的两个 BigInteger 的数组。

**参数：**

val - 此 BigInteger 要除以的值和计算所得的余数。

**返回：**

两个 BigInteger 的数组：商 (this / val) 是初始元素，余数 (this % val) 是最终元素。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - val==0

**remainder**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **remainder**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this % val) 的 BigInteger。

**参数：**

val - 此 BigInteger 要除以的值和计算所得的余数。

**返回：**

this % val

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - val==0

**pow**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **pow**(int exponent)

返回其值为 (thisexponent) 的 BigInteger。注意， exponent 是一个整数而不是 BigInteger。

**参数：**

exponent - 此 BigInteger 的指数。

**返回：**

thisexponent

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - exponent 为负。（这会导致该运算产生一个非整数值。）

**gcd**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **gcd**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回一个 BigInteger，其值是 abs(this) 和 abs(val) 的最大公约数。如果 this==0 && val==0，则返回 0 。

**参数：**

val - 要一起计算最大公约数的值。

**返回：**

GCD(abs(this), abs(val))

**abs**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **abs**()

返回其值是此 BigInteger 的绝对值的 BigInteger。

**返回：**

abs(this)

**negate**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **negate**()

返回其值是 (-this) 的 BigInteger。

**返回：**

-this

**signum**

public int **signum**()

返回此 BigInteger 的正负号函数。

**返回：**

当此 BigInteger 的值为负、零或正时，返回 -1、0 或 1。

**mod**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **mod**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) m)

返回其值为 (this mod m) 的 BigInteger。此方法不同于 remainder，因为它始终返回一个 *非负* BigInteger。

**参数：**

m - 模数。

**返回：**

this mod m

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - m <= 0

**另请参见：**

[remainder(java.math.BigInteger)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#remainder(java.math.BigInteger))

**modPow**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **modPow**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) exponent,

[BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) m)

返回其值为 (thisexponent mod m) 的 BigInteger。（与 pow 不同，此方法允许使用负指数。）

**参数：**

exponent - 该指数。

m - 模数。

**返回：**

thisexponent mod m

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - m <= 0

**另请参见：**

[modInverse(java.math.BigInteger)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#modInverse(java.math.BigInteger))

**modInverse**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **modInverse**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) m)

返回其值为 (this-1 mod m) 的 BigInteger。

**参数：**

m - 模数。

**返回：**

this-1 mod m.

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - m <= 0，或者此 BigInteger 没有乘法可逆元 mod m（即此 BigInteger 不是 m 的 *相对素数*）。

**shiftLeft**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **shiftLeft**(int n)

返回其值为 (this << n) 的 BigInteger。位移距离 n 可以为负，在此情况下，此方法执行右移操作。（计算 floor(this \* 2n)。）

**参数：**

n - 以位为单位的位移距离。

**返回：**

this << n

**另请参见：**

[shiftRight(int)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#shiftRight(int))

**shiftRight**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **shiftRight**(int n)

返回其值为 (this >> n) 的 BigInteger。执行符号扩展。位移距离 n 可以为负，在此情况下，此方法执行左移操作。（计算 floor(this / 2n)。）

**参数：**

n - 以位为单位的位移距离。

**返回：**

this >> n

**另请参见：**

[shiftLeft(int)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#shiftLeft(int))

**and**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **and**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this & val) 的 BigInteger。（当且仅当 this 和 val 同时为负时，此方法返回一个负 BigInteger。）

**参数：**

val - 要与此 BigInteger 执行 AND（与）运算的值。

**返回：**

this & val

**or**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **or**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this | val) 的 BigInteger。（当且仅当 this 和 val 之一为负时，此方法返回一个负 BigInteger。）

**参数：**

val - 要与此 BigInteger 执行或运算的值。

**返回：**

this | val

**xor**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **xor**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this ^ val) 的 BigInteger。（当且仅当 this 和 val 中只有一个为负时，此方法返回一个负 BigInteger。）

**参数：**

val - 要与此 BigInteger 执行异或运算的值。

**返回：**

this ^ val

**not**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **not**()

返回其值为 (~this) 的 BigInteger。（当且仅当此 BigInteger 为非负时，此方法返回一个负值。）

**返回：**

~this

**andNot**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **andNot**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回其值为 (this & ~val) 的 BigInteger。此方法等效于 and(val.not())，提供此方法是为了更方便地进行屏蔽操作。（当且仅当 this 为负且 val 为正时，此方法返回一个负 BigInteger。）

**参数：**

val - 要进行求补且与此 BigInteger 执行与运算的值。

**返回：**

this & ~val

**testBit**

public boolean **testBit**(int n)

当且仅当设置了指定的位时，返回 true。（计算 ((this & (1<<n)) != 0)。）

**参数：**

n - 要测试的位的索引。

**返回：**

当且仅当设置了指定的位时，返回 true。

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - n 为负。

**setBit**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **setBit**(int n)

返回其值与设置了指定位的此 BigInteger 等效的 BigInteger。（计算 (this | (1<<n))。）

**参数：**

n - 要设置的位的索引。

**返回：**

this | (1<<n)

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - n 为负。

**clearBit**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **clearBit**(int n)

返回其值与清除了指定位的此 BigInteger 等效的 BigInteger。（计算 (this & ~(1<<n))。）

**参数：**

n - 要清除的位的索引。

**返回：**

this & ~(1<<n)

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - n 为负。

**flipBit**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **flipBit**(int n)

返回其值与对此 BigInteger 进行指定位翻转后的值等效的 BigInteger。（计算 (this ^ (1<<n))。）

**参数：**

n - 要翻转的位的索引。

**返回：**

this ^ (1<<n)

**抛出：**

[ArithmeticException](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/ArithmeticException.html) - n 为负。

**getLowestSetBit**

public int **getLowestSetBit**()

返回此 BigInteger 最右端（最低位）1 比特的索引（即从此字节的右端开始到本字节中最右端 1 比特之间的 0 比特的位数）。如果此 BigInteger 不包含一位，则返回 -1。（计算 (this==0? -1 : log2(this & -this))。)

**返回：**

此 BigInteger 中最右端的一比特位的索引。

**bitLength**

public int **bitLength**()

返回此 BigInteger 的最小的二进制补码表示形式的位数， *不包括* 符号位。对于正 BigInteger，这等于常规二进制表示形式中的位数。（计算 (ceil(log2(this < 0 ? -this : this+1)))。)

**返回：**

返回此 BigInteger 的最小的二进制补码表示形式中的位数， *不包括* 符号位。

**bitCount**

public int **bitCount**()

返回此 BigInteger 的二进制补码表示形式中与符号不同的位的数量。此方法在实现 BigInteger 上的位向量样式设置时非常有用。

**返回：**

返回此 BigInteger 的二进制补码表示形式中与符号位不同的位的数量。

**isProbablePrime**

public boolean **isProbablePrime**(int certainty)

如果此 BigInteger 可能为素数，则返回 true，如果它一定为合数，则返回 false。如果 certainty <= 0，则返回 true。

**参数：**

certainty - 调用方允许的不确定性的度量。如果该调用返回 true，则此 BigInteger 是素数的概率超出 (1 - 1/2certainty)。此方法的执行时间与此参数的值是成比例的。

**返回：**

如果此 BigInteger 可能为素数，则返回 true，如果它一定为合数，则返回 false。

**compareTo**

public int **compareTo**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

将此 BigInteger 与指定的 BigInteger 进行比较。对于针对六个布尔比较运算符 (<, ==, >, >=, !=, <=) 中的每一个运算符的各个方法，优先提供此方法。执行这些比较的建议语句是： (x.compareTo(y) < *op*> 0)，其中 < *op*> 是六个比较运算符之一。

**指定者：**

接口 [Comparable](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Comparable.html)<[BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类)> 中的 [compareTo](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Comparable.html" \l "compareTo(T))

**参数：**

val - 将此 BigInteger 与之比较的 BigInteger。

**返回：**

当此 BigInteger 在数值上小于、等于或大于 val 时，返回 -1，0，或 1。

**equals**

public boolean **equals**([Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) x)

比较此 BigInteger 与指定的 Object 的相等性。

**覆盖：**

类 [Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) 中的 [equals](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html#equals(java.lang.Object))

**参数：**

x - 将此 BigInteger 与之比较的 Object。

**返回：**

当且仅当指定的 Object 是一个其值在数字上等于此 BigInteger 的 BigInteger 时，返回 true。

**另请参见：**

[Object.hashCode()](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html#hashCode()), [Hashtable](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/util/Hashtable.html)

**min**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **min**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回此 BigInteger 和 val 的最小值。

**参数：**

val - 要计算最小值的值。

**返回：**

其值为此 BigInteger 和 val 中的较小值的 BigInteger。如果它们相等，可能返回其中之一。

**max**

public [BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html) **max**([BigInteger](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html" \o "java.math 中的类) val)

返回此 BigInteger 和 val 的最大值。

**参数：**

val - 要计算最大值的值。

**返回：**

其值为此 BigInteger 和 val 中较大值的 BigInteger。如果它们相等，可能返回其中之一。

**hashCode**

public int **hashCode**()

返回此 BigInteger 的哈希码。

**覆盖：**

类 [Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) 中的 [hashCode](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html" \l "hashCode())

**返回：**

此 BigInteger 的哈希码。

**toString**

public [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) **toString**(int radix)

返回此 BigInteger 的给定基数的字符串表示形式。如果该基数超出从 [Character.MIN\_RADIX](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Character.html" \l "MIN_RADIX) 到 [Character.MAX\_RADIX](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Character.html" \l "MAX_RADIX)（包括）这一范围，则其默认值为 10（ Integer.toString 就是这种情况）。使用由 Character.forDigit 提供的从数字到字符的映射，并在需要时在前面加一个负号。（此表示形式与 [(String, int)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#BigInteger(java.lang.String,%20int)) 构造方法兼容。）

**参数：**

radix - 字符串表示形式的基数。

**返回：**

此 BigInteger 给定基数的字符串表示形式。

**toString**

public [String](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/String.html) **toString**()

返回此 BigInteger 的十进制字符串表示形式。使用由 Character.forDigit 提供的从数字到字符的映射，并在需要时在前面加一个负号。（此表示形式与 [(String)](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#BigInteger(java.lang.String)) 构造方法兼容，并允许使用 Java 的 + 运算符将字符串连接。）

**覆盖：**

类 [Object](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html) 中的 [toString](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Object.html" \l "toString())

**返回：**

此 BigInteger 的十进制字符串表示形式。

**toByteArray**

public byte[] **toByteArray**()

返回一个 byte 数组，该数组包含此 BigInteger 的二进制补码表示形式。该 byte 数组将为 *big-endian* 字节顺序：最高有效字节在第零个元素中。此数组将包含表示此 BigInteger 所需的最小数量的字节，至少包括一个符号位，即 (ceil((this.bitLength() + 1)/8))。（此表示形式与 [(byte[])](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/math/BigInteger.html#BigInteger(byte[])) 构造方法兼容。）

**返回：**

一个包含此 BigInteger 的二进制补码表示形式的 byte 数组。

**intValue**

public int **intValue**()

将此 BigInteger 转换为 int。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 long 到 int 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：如果此 BigInteger 太长而不适合用 int 表示，则仅返回 32 位的低位字节。注意，此转换会丢失关于该 BigInteger 值的总大小的信息，并返回带有相反符号的结果。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [intValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html" \l "intValue())

**返回：**

转换为 int 的此 BigInteger。

**longValue**

public long **longValue**()

将此 BigInteger 转换为 long。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 long 到 int 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：如果此 BigInteger 太长而不适合用 long 表示，则仅返回 64 位的低位字节。注意，此转换会丢失关于该 BigInteger 值的总大小的信息，并返回带有相反符号的结果。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [longValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html" \l "longValue())

**返回：**

转换为 long 的此 BigInteger。

**floatValue**

public float **floatValue**()

将此 BigInteger 转换为 float。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 double 到 float 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：如果此 BigInteger 的数量太大，不能表示为float，则将其适当地转换为 [Float.NEGATIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Float.html" \l "NEGATIVE_INFINITY) 或 [Float.POSITIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Float.html" \l "POSITIVE_INFINITY)。注意，即使在返回值是有限的情况下，此转换也可以丢失关于 BigInteger 值的精度的信息。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [floatValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html" \l "floatValue())

**返回：**

转换为 float 的此 BigInteger。

**doubleValue**

public double **doubleValue**()

将此 BigInteger 转换为 double。此转换类似于 [Java Language Specification](http://java.sun.com/docs/books/jls/html/) 中定义的从 double 到 float 的 [*基本收缩转换*](http://java.sun.com/docs/books/jls/second_edition/html/conversions.doc.html#25363)：如果此 BigInteger 的数量太大，不能表示为double，则将其适当地转换为 [Double.NEGATIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Double.html" \l "NEGATIVE_INFINITY) 或 [Double.POSITIVE\_INFINITY](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Double.html" \l "POSITIVE_INFINITY)。注意，即使在返回值是有限的情况下，此转换也可以丢失关于 BigInteger 值的精度的信息。

**指定者：**

类 [Number](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html) 中的 [doubleValue](http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/Number.html" \l "doubleValue())

**返回：**

转换为 double 的此 BigInteger。