

Lab1 说明:

姓名: 程茜 学号 19302010084

1. bitAnd:

根据德摩根律有, $\text{非}(P \text{ 且 } Q) = (\text{非 } P) \text{ 或 } (\text{非 } Q)$, 所以 $P \text{ 且 } Q = \text{非}((\text{非 } P) \text{ 或 } (\text{非 } Q))$;

2. getByte:

只要把要取的第 n 个字节右移到最低位, 则其右边的字节会被清零, 而 n 个字节前面的字节可以通过取与 $0xff(255)$, 则可以清零 n 字节前面的字节而保留 n 字节本身。

3. logicalShift:

可以先进性算术右移, 在将补位全部换成 0, 这可以利用左边 n 位为 0, 右边 $32-n$ 位为 1 的掩码与原来的数算术右移后的数取 $\&$, 掩码可以将 1 左移 31 位, 再算术右移 $(n-1)$ 位, 由于不能使用-, 故可以先右移 n 位, 再左移 1 位;

4. bitCount:

采用自底向上的方法, 先求每两位中 1 的个数, 在求每 4 位中 1 的个数, 以此类推最后求 32 位中 1 的个数, 也是利用掩码。

5. Bang:

只需要判断该数的二进制位是否均为 0。因为只有 $0x00$ 和其补码取 '!' 之后最高位是 0, 其他数均为 1, 所以可以将原数与其补码取或 '!', 在右移 31 位, 再与 1 取并 $\&$, 之后再取反。

6. tmin:

最小的二进制补码应该只有最高位为 1, 其他为 0, 故可以直接将 $0x1$ 左移 31 位得到

7. fitsBits

x 左移 $(32-n)$ 再右移 $(32-n)$ 后如果和原数相等则表示可以被 n 位补码表示, 判断相等的结果再加一个! 则可以得到结果。

8. divpwr2

若 x 为正数, 则 $x/(2^n)=x>>n$, 若 $x<0$, $x/(2^n)=(x+(1<<n)-1)>>n$, 只需要再构造一个数, 使得 $x>0$, 该数为零, $x<0$, 该数为 $(1<<n)-1$, 可以取 $(x>>31)$ 与 $(1<<n)-1$ 的并。

9. Negate

将 x 先取反再加 1

10. isPositive

x 的正负至于最高位符号位有关, 直接将 x 右移 31 位, 再取与! x 的并。

11. isLessOrEqual

分为 x 和 y 同号和异号两种情况, 将 $y-x$ ($y+\sim x+1$), 当 y 、 x 同号时判断 $y-x$ 符号位即可,

当 y 、 x 异号时可能会发生溢出，需要看 y 、 x 的符号位。

12. `ilog2`

令 $\text{sign} = \text{!!}(x \gg 16)$ ，若 x 二进制数左边 16 位有 1，则 sign 为 1，反之为 0，再令 $\text{shift} = \text{sign} \ll 16$ ，则 shift 为 16 或 0，若 sign 为 1，则将 x 右移 16 位，丢掉右边 16 位；同理再令 $\text{sign} = \text{!!}(x \gg 8)$ 、 $\text{sign} = \text{!!}(x \gg 4)$ 、 $\text{sign} = \text{!!}(x \gg 2)$ 、 $\text{sign} = \text{!!}(x \gg 1)$ ，最后将每次的 shift 加起来即为所求值。