**Chapter 5 Bit Manipulation**

An operation like x & (~0 « n) clears the n rightmost bits of x. The value ~0 is simply a sequence of 1 s, so by shifting it left by n, we have a bunch of ones followed by n zeros. By doing an AND with x, we clear the rightmost n bits of x.

((n & (n-1)) == 0) checks if n is a power of 2 (or if n is 0).

**Clear Bit**

This method operates in almost the reverse of setBit. First, we create a number like

11101111 by creating the reverse of it (00010000) and negating it. Then, we perform

an AND with num. This will clear the ith bit and leave the remainder unchanged.

int clearBit(int num, int i) {

int mask = ~(1 « i);

return num & mask;

}

int clearBitsMSBthroughI(int num, int i) {

int mask = (1 « i) - 1;

return num & mask;

}

int clearBitsIthrough0(int num, int i) {

int mask = ~((l « (i+1)) - 1);

return num & mask;

}