

3) (10 pts) DSN (Bitwise Operators)

It is getting harder and harder to stay green while using computers. You decided that you will reduce your carbon footprint by storing fewer bits for your integers. How you might ask? You will store your integer plus some power of 2. Your goal will be to use as few on-bits as possible in the resulting sum. This way you will have less bits on and just like turning off the lights when you leave a room you will be saving energy and the planet. Complete the function below so that it determines and returns the least number of bits that will be on after adding a positive power of 2 to your number. For full credit your function should take $O(b)$ time where b is the number of bits in an int. You are guaranteed that no positive power of 2 added to the original number will result in an overflow.

For example, the value $76 = 2^2 + 2^3 + 2^6$ can have 4 added to it to result in $80 = 2^4 + 2^6$, which requires only 2 on-bits. No better result can be achieved by adding a different power of 2.

```
int leastBitsOn(int x) {  
  
    int numOn = 0;  
    int cur = 0;  
    int longest = 0;  
  
    for (int i = 0; i < 8 * sizeof(int); i++) {  
  
        // Determine if the bit is on (3 pts)  
        if ((x & (1 << i)) != 0) {  
            cur++; // Update the bits on in the current run (1 pt)  
            numOn++; // Update the bits on the current number (1 pt)  
        }  
  
        else {  
            cur = 0; // Reset, 1 pt  
        }  
  
        // (+2 pts) Update the longest run of bits.  
        if (longest < cur) {  
            longest = cur;  
        }  
    }  
  
    // (+2 pts) Remove the longest run AND (+1 pt) add 1  
    return numOn + 1 - longest;  
}
```