

Modelado y Programación 2017-1:

Práctica 11, Bitwise Tricks.

Cinthia Rodriguez Maya (cinthia.rodriguez@ciencias.unam.mx)
María del Carmen Sánchez Almanza (carmensanchez@ciencias.unam.mx)
Ricchy Alaín Pérez Chevanier (alain.chevanier@ciencias.unam.mx)

Fecha de Publicación: 07 de Noviembre de 2016
Fecha de Entrega: 13 de Noviembre de 2016

1 Introducción

Una operación bitwise opera sobre uno o más patrones de bits o números binarios al nivel de sus bits individuales. Es una operación rápida y simple soportada directamente por el procesador, y es utilizada para manipular valores para comparaciones y cálculos.

En procesadores muy simples/de bajo costo/viejos, típicamente las operaciones bitwise son substancialmente más rápidas que la división, varias veces más rápidas que la multiplicación y algunas veces significativamente más rápidas que la suma. Mientras que en los procesadores modernos usualmente realizan sumas y multiplicaciones tan rápido como las operaciones bitwise dado que tienen un pipeline de instrucciones más grande y otras elecciones de diseño en la arquitectura del procesador. Las operaciones bitwise por lo regular utilizan menos energía dado su reducido uso de recursos.

Algunos usos comunes en operaciones bitwise son los siguientes:

- Multiplicar o Dividir por potencias de 2

```
x<<n // x * 2^n
x>>n // x / 2^n
```

- Prueba de la paridad de un número

```
(x&1) == 1 // x es impar
(x&1) == 0 // x es par
```

- Ver el valor del n-ésimo bit

```
(x & (1<<n)) == 1 // Está prendido
(x & (1<<n)) == 0 // En caso contrario está apagado
```

- Alterar el valor del n-ésimo bit

```
y = x | (1<<n) // prende el bit
z = x & ~(1<<n) // apaga el bit
w = x ^ (1<<n) // invierte el valor del bit
```

- Equivalencia

```
x-1 == ~-x
```

- Operaciones sobre el último bit prendido

```
a = x & ~x // apaga el último bit prendido
b = x & -x // aísla el valor del último bit prendido
c = x | (x-1) // propaga el último bit prendido hacia la derecha
```

- Operaciones sobre el último bit apagado

```
a = x | (x+1) // Prende el último bit apagado
b = ~x & (x+1) // aísla el último bit apagado
```

2 Descripción del Problema

Dado un número $1 \leq N \leq 10^9$. Encuentra el número más pequeño mayor que N que tiene el mismo número de bits activos que él. (i.e. el mismo número de 1s en su presentación binaria).

Por ejemplo para

- $1 \rightarrow 2$
- $3 \rightarrow 5$
- $10 \rightarrow 12$
- $1260132 \rightarrow 1260136$
- $581369493 \rightarrow 581369494$
- $1000000000 \rightarrow 1000000512$
- $7 \rightarrow 11$
- $15 \rightarrow 23$
- $44 \rightarrow 49$
- $78 \rightarrow 83$

Entrada

- Línea 1: Un entero T que indica el número de casos.
- Línea 2..T+1: Un entero N que representa la entrada del problema.

Salida

- Línea 1...T: Un entero M que es la solución del problema.

Ejemplos

3 Entrega

Te recomendamos resolver este reto en algún lenguaje que soporte operaciones bitwise sobre el tipo integer como C, C++ o Java. Es necesario que en tu archivo README des una explicación breve de cual es la idea que utilizaste para resolver el reto.