Left = 8, Right = 13.

Lft will always be less than right

	8	4	2	1
8 →	1	0	0	0
9 →	1	0	0	1
10 →	1	0	1	0
11 →	1	0	1	1
12 →	1	1	0	0
13 →	1	1	0	1
	1	0	0	0

There is only one position where 1 is common

And operation will have, 1 if all bit are 1, else it will be 0

0	0	0
0	1	0
1	0	0
1	1	1

Solution 1

Turn off the right bit of Largest element as long as it is equal to the smallest element

Keep a count of it, Now when smallest & Largest Number look the same, push the smallest Number to a position where common 1 was found, that will be the count

L: 1000
R: 1101

→ Count 1: 0100, 0110 → Count 2: 0010, 0010

} Now both the value are same

Push Lft by 2 position, this is where they had common 1

L << 2

0010 << 2 = 1000 = (8)

Solution 2 : Brian Kernighan's Algorithm.

Turn of the right bit is core concept of the algorithm.

Use the algorithm to turn of right bit of Largest num

```
L: 1000  -  1000  →  1000 ]
Y: 1101     1100     1000 ]
```

Now Largest & Smallest are Same

i.e the have a common bit at position of right

return 1000 = (8)