Hamming Distance

* Total Accepted: 62204
* Total Submissions: 88349
* Difficulty: Easy
* Contributors:[Samuri](https://leetcode.com/samuri/)

The [Hamming distance](https://en.wikipedia.org/wiki/Hamming_distance) between two integers is the number of positions at which the corresponding bits are different.

Given two integers x and y, calculate the Hamming distance.

**Note:**  
0 ≤ x, y < 231.

Solution is simple

1. Take XOR of two number, say *res*; this will give us the positions at which bits differ in these two numbers.
2. Now count the number of 1’s in the *res.* Now, we can count using for loop of 32 iterations. But this is not the most optimized way. We can optimize by pre-calculating the number of bits set for a given number, in an array. Now, all we need to do is to look up this array (a look-up table). Though this is quite helpful, we cannot have big array to fit all 32-bit space (i.e., 232 numbers). So, we can reduce it to probably 4-bits at a time (i.e., 16 entries). Thus, now we can calculate the number of bits set using 8 iterations instead of 32. This can be further optimized by having larger arrays like for 8-bit space (256 entries); Here, we need just 4 iterations. Example of such an array {0, 1, 1, 2, 1, 2, 2, 3}.

|  |  |  |
| --- | --- | --- |
| Number | Binary | Number of bits set. |
| 0 | 000 | 0 |
| 1 | 001 | 1 |
| 2 | 010 | 1 |
| 3 | 011 | 2 |
| 4 | 100 | 1 |
| 5 | 101 | 2 |
| 6 | 110 | 2 |
| 7 | 111 | 3 |