SumMax Solution

If you see any bitwise operator, then it is benificial to think in binary than in base 10. Consider the i^{th} bit from LSB where $i < k$, that is, 2^i is its value in base 10. Let a_i, b_i, x_i denote i^{th} bit in a, b, x respectively. Let us greedily check if we can set each bit in max.

Consider the case where $a_i = b_i$. If a_i and b_i are set then x_i should be unset for the i^{th} bit to be set in max. If a_i and b_i are unset then x_i should be set for the i^{th} bit to be set in max. Let us call such a bit as $fixed\ bit$ as its value doesen't change for all x.

Now, consider the case where a_i and b_i are different. It doesn't matter if we set x_i or not, the bit cannot be set in max. Let us call such a bit $free\ bit$. These free bits give rise to different values of x. Let there be *cnt* free bits. Each free bit can be 0 or 1. So, there are 2^{cnt} different values of x.

Let *sum* denote the final answer.

Then, the fixed bits are the same in all x. If any fixed bit i is set, then add $2^i \cdot 2^{cnt} \bmod 10^9 + 7$ to the sum. For any free bit j add $2^j \cdot 2^{cnt-1} \bmod 10^9 + 7$ to the sum as each free bit is set in exactly half of all possible x. Notice how we don't need to calculate max to find the sum.

Time Complexity: $O(k)$ per test case.