```
struct BitTrie {
    int id = 0, N;
    std::vector<std::array<int, 2>> trie {std::array<int, 2>()};
    std::vector<int> cnt {0};
    BitTrie() {}
    BitTrie(int _) {
        init(_);
    }
    void init(int _) {
        N = _;
        id = 0;
        cnt.assign(1, {});
        trie.assign(1, std::array<int, 2>());
    }
    void insert(int x, int v) {
        int p = 0;
        for (int i = N - 1; i >= 0; --i) {
            int it = x >> i & 1;
            if (!trie[p][it]) {
                trie[p][it] = ++id;
                trie.push_back(std::array<int, 2>());
                cnt.push_back(0);
            }
            p = trie[p][it];
            cnt[p]++;
        }
    }
    int find(int u, int x) {
        int p = 0, res = 0;
        for (int i = N - 1; i >= 0; --i) {
            int it = u >> i & 1, bx = x >> i & 1;
            if (!bx) {
                if (trie[p][it ^ 1]) {
                    res += cnt[trie[p][it ^ 1]];
                }
                p = trie[p][it];
            } else {
                p = trie[p][it ^ 1];
            }
            if (!i) {
                res += cnt[p];
            }
            if (!p) {
                break;
            }
        }
        return res;
    }
};
```