Trie树

别称：前缀树；字典树

421. 数组中两个数的最大异或值

思路1:

暴力

```
private int forceSolution(int[] nums){
    // 暴力解决方案
     int max = 0;

    for ( int i = 0; i < nums.length; i++){
        for ( int j = i; j < nums.length; j++){
            max = Math.max(nums[i] ^ nums[j], max);
        }
    }

    return max;
}
```

思路2:

1. 将二进制串倒序拼接为31位放入Trie树; 离树根近的,是高位;
2. DFS
3. DFS时,不断剪枝; 如果该层有1,则不需要异或为0的;
4. 因为该层只有0,1; 那么可以直接用二叉树
5. 因为全部补齐为31位,所以,该树是完全二叉树

```
private int trieSolution(int[] nums){
    TreeNode root = new TreeNode();
    for(int num : nums){
        addToTrieTree(root, num);
    }
    return dfs(root, root, MAX);
}

private int dfs(TreeNode left, TreeNode right, int level){
    if(level == -1 || left == null || right == null){
        return 0;
    }
    int max = 0;
    if (left.left != null && right.right != null){
        max = (1 << level) + dfs(left.left, right.right, level -1);
    }
    if (left.right != null && right.left != null){
```

```
            max = Math.max((1 << level) + dfs(left.right, right.left, level -1), max);
        }

        if (max != 0){
            return max;
        }

        if (left.left != null && right.left != null){
            max = dfs(left.left, right.left, level -1);
        }

        return Math.max(max, dfs(left.right, right.right, level -1));

    }
```

思路3:

掩码

思路4:

哈希表

思路5:

Trie树

208. 实现 Trie (前缀树)
720. 词典中最长的单词
648. 单词替换
677. 键值映射