/\*

    集合论  --------->  位运算

    例如：二进制从高到低第i位为1表示i在集合中，为0表示i不在集合中，那么集合{0,2,3}可以使用二进制数1101来表示（包含非负整数的集合可以压缩成为一个数字）

    分类：

1. 集合与集合

集合 位运算

    交集 ： A ∩ B a&b

并集 ： A **∪** B a|b

对称差 ： A Δ B a^b

差 ： A \ B a&(~b)

差（子集）： A \ B a^b

（B⊆A）

包含于 ： A ⊆ B a & b = a

a | b = b

====================================================

1. 集合与元素

左移 (<<) 右移 (>>)

集合 位运算

空集 ： ∅ 0

单集合元素 : {i} 1 << i

全集： U = {0,1,……,n-1} (1 << n)-1

补集： CuS = U∖S ~s或者((1 << n)−1)^s

属于： i∈ S (s >> i) & 1 = 1

不属于： i/∈S (s >> i) & 1 = 0

添加元素： S∪{i} s | ( 1 << i)

删除元素： S∖{i} s & ∼(1 << i)

删除元素： S∖{i} && i∈ S s ^ (1 << i)

（一定在集合中）

删除最小元素： s & (s − 1)

使用内置函数：

集合大小： \_\_builtin\_popcount(s)

二进制长度（减一得到集合中的最大元素）： 32 - \_\_builtin\_clz(s)

集合中的最小元素： \_\_builtin\_ctz(s)

    ===================================================

1. 遍历集合：

设元素范围从0到n-1，挨个判断每个元素是否在集合中s中：

for (int i = 0; i < n; i++) {

if ((s >> i) & 1) { // i 在 s 中

// 处理 i 的逻辑

}

}

1. 枚举集合：

设元素范围从0到n-1，挨个判断每个元素是否在集合中s中：

for (int s = 0; s < (1 << n); s++) {

// 处理 s 的逻辑

}

设集合为s，从大到小枚举s的所有非空子集sub：

for (int sub = s; sub; sub = (sub - 1) & s) {

// 处理 sub 的逻辑

}

如果从到到小枚举s的所有子集sub（从s枚举到空集）

int sub = s;

do {

// 处理 sub 的逻辑

sub = (sub - 1) & s;

} while (sub != s);

\*/