

1 送你一堆区间

首先对区间离散化, 再按左端点从小到大排序, 若左端点相同, 则按右端点从大到小排序.

考虑计算 $dp(i)$ 表示前 i 个关键点都完全覆盖的方案.

对于一个区间 $[l, r]$, 它可以将 $dp(l-1) \dots dp(r-1)$ 转移到 $dp(r)$ 上; 还可以将 $dp(r) \dots dp(m)$ 通乘上一个 2. 用线段树维护即可.

2 送你一个集合

$$\binom{12}{6} = 924 > 920$$

```
const int LIM = 12;
int type, N;
vector<int> code;

void init()
{
    for (int s = 0; s < (1 << LIM) && SZ(code) < N; ++s) {
        if (__builtin_popcount(s) == LIM / 2)
            code.push_back(s);
    }
}

int encode(int x, int y)
{
    return 32 - __builtin_clz((code[x - 1] ^ code[y - 1]) & code[x - 1]);
}

bool decode(int q, int h)
{
    return code[q - 1] >> (h - 1) & 1;
}
```

3 送你一朵圣诞树

显然, 对于权值最小的点, 选了它的父亲后一定马上就会选它.

这样我们可以把这两个点合并, 那么每个点现在都是一个二元组 (s_i, t_i) , s_i 表示权值之和, t_i 表示点的个数.

那么两个二元组 i, j 之间也是可以比较的. i 比 j 优的条件就是

$$t_i s_j > t_j s_i$$
$$\frac{s_i}{t_i} < \frac{s_j}{t_j}$$

于是点合并之后也可以用 $\frac{s_i}{t_i}$ 作为关键字比较大小. 并查集 + 堆即可.