

# Solution

北京大学  
周思源

# jump

- 100%：每次瞬移一定是到一个尚未被跳过的格子上
- 每次找到一个未被跳过的格子开始跳跃
- 跳跃的过程相当于一次 bfs，把所有能遍历到的格子都遍历一遍

# love

- 100%：区间  $[L, R]$  的答案可以用  $[1, R]$  的答案减去  $[1, L]$  的答案
- 设  $R$  的二进制表示为  $x$ ，对于  $[1, R]$  的答案，我们可以枚举二进制的长度和哪一位为 0。假设当前枚举长度为  $l$ ，则另  $y = (1 \ll l) - 1$ ，设枚举第  $i$  位为 0，则判断  $y - (1 \ll i)$  是否小于等于  $x$

# Z

- 题意：有  $n$  种物品，每种物品有大小、收益。每种物品还有一定个数。要求收益至少为  $P$  的情况下大小最小。有  $m$  种箱子，每种箱子有大小和代价。每种箱子还有一定个数，需要用最少代价去装上一个问题的物品。
- 30%：暴搜即可
- 60%：第一个问题就是多重背包。 $f[i]$  表示收益为  $i$  的最小大小。第二个问题也是多重背包。 $g[i]$  表示代价为  $i$  时的最大大小。复杂度  $O(n * P * 100)$
- 100%：考虑用二进制优化。对于每种物品的个数，我们可以将其分为 1 个，2 个，4 个，... 的众多二的次方之和。对于个数不足 2 的次方的，可以再新建一个物品。对于每个新物品，假设由原来物品的  $p$  个组合而成，则价值为原价值  $*p$ ，大小为原大小  $*p$ 。这样我们就可以把原问题转化为 01 背包。复杂度  $O(n * P * \log(100))$

# short

- 30%：暴力枚举哪些边使用魔法，跑最短路
- 60%：考虑 floyd， $dis[k][i][j]$  表示使用了  $k$  次魔法  $i$  到  $j$  的最短路。 $dis[k][i][j] = \min\{dis[k1][i][mid] + dis[k - k1][mid][j]\}$
- 100%：可以使用类似快速幂的方法预处理  $dis[k][i][j]$ 。 $dis[k][i][j]$  表示使用了  $2^k$  次魔法  $i$  到  $j$  的最短路。 $dis[k][i][j] = dis[k-1][i][mid] + dis[k - 1][mid][j]$
- 对于  $K$ ，可以看做一些  $2^k$  的和。
- $ans[i][j] = \min\{dis[k1][i][mid1] + \dots + dis[km][midm][j]\}$