序列

尝试构造：

把那些连续递减序列叫做逆序块，显然块大小不超过 y，

将初始序列置为 $1, 2, \cdots, n$，将最后 y 个数逆序得到
$1, 2, \cdots, n-y(n, n-1, \cdots, n-y+1)$，

等价于在末尾构造了一个 y 大小的逆序块，使得 LIS 减小了 $y-1$ 变为了 $n-(y-1)$。

所以，我们通过不断构造逆序块来减小 LIS，同时逆序块大小不超过 y，也满足最长下降子序列的条件。

最后分析**无解**的情况：

我们提出了构造逆序块的方法，那么最恰好的情况，所有逆序块大小均为 y 且总长刚好为 n，此时 LIS 即为逆序块的个数，记为 x，那么有 $x \times y = n$。

如果 $x \times y < n$，说明构造了 x 个大小为 y 的逆序块后，序列前端还有数字，此时不可能满足条件，LIS 必然大于 x，无解。

另一种情况：若 $x + y > n + 1$，显然无解。

链接

由题：每条边的权值是 2^i，我们可以发现对于第 i 条边满足，$2^1 + 2^2 + ... + 2^{i-1} < 2^i$。

所以如果两个点能通过前 $i-1$ 条边到达（加入前 $i-1$ 条边后联通），那肯定比通过第 i 条更优，所以我们从 1 到 n 按顺序建最小生成树（Kruskal）。

对于 0 点和 1 点的最短路，我们只需枚举最小生成树上的每条边，这条边两侧的任意 01 点组合的最短路都会通过这条边，所以统计两侧的黑白点个数，计算贡献即可。

数字对

30%：暴力枚举判断。$\mathcal{O}(n^4)$。

60%：特殊区间的特点实际上就是区间最小值等于这个区间的 GCD，于是暴力或递推算出每个区间最小值与 GCD。而对于最大价值，可以通过二分来进行求解。复杂度 $\mathcal{O}(n^2)$。

100%：在 60% 的基础上，最小值与 GCD 都使用 RMQ 算法来求解，对于这道题推荐使用 ST 表。最大价值仍然使用二分。复杂度 $\mathcal{O}(nlogn)$。

不死

设 $f[i][S]$ 表示当前已经上完了第 i 节课，且集合 S 中的科目目前不能再睡，这时 LYM 的最小疲劳值。

转移方程：

设 $p[i]$ 为第 i 节课的科目代号，$Down[i]$ 为睡第 i 节课疲劳的减少值， $Up[i]$ 为上第 i 节课疲劳的增加值，如果 i 不是主科 $f[i][s] = f[i-1][s] - Down[i]$，直接睡第 i 节课。

如果 i 是主科：

1. 如果睡第 i 节睡，即 $(S >> (p[i]-1) \& 1)$ 的值为 1：第 i 节课上的决策是睡觉。因为 S 是上完这节课之后的状态，上完这节课之后，该科目不能再睡，说明这节课肯定是睡了的。$f[i][s] = f[i-1][s^(1 << (p[i]-1)] - Down[i]$。
2. 如果不睡第 i 节：
   $f[i][s] = min(f[i-1][s] + Up[i], f[i-1][s\ XOR\ (1 << (p[i]-1)] + Up[i])$
   这里有两种状态需要决策：
   $f[i-1][s] + Up[i]$ 表示若第 i 节可以睡，但没有睡。
   $f[i-1][s\ XOR\ (1 << (p[i]-1)] + Up[i]$ 表示若第 i 节不能睡，上完第 i 节后，状态 s 变为可以睡。

上述的转移方程还应有所限制：只要疲劳值超过了疲劳限度，LYM 会死,因此在转移时，只有疲劳值小于等于忍耐限度的状态，才可以被转移。

但是我们并不知道疲劳限度，最简单的处理方式是二分答案，这很显然满足**二分性质**：

如果忍耐限度不够高，第 n 天的所有状态的都超过了限度， LYM 死定了。如果第 n 天存在小于等于限度的状态，就说明忍耐限度足够了。

状压 DP + 二分答案，时间复杂度 $\mathcal{O}(S*n*logMAXN)$。其中 S 是子集的数目， 64 个， $MAXN$ 是二分上限。