

T1 染色 (paint)

数据范围很小，而且所涉及到的端点全都是整数，所以我们可以直接枚举 i ，判断 $[i, i + 1)$ 这一段是否即涂蓝又涂红，实则是一个区间包含关系的判断。

但是也不用这么麻烦，翻译题意后，我们要求的是两个区间的交。我们可以先判断两个区间是否有交集，若存在交集，答案为 $\min(R_1, R_2) - \max(L_1, L_2)$ ，否则答案为 0。

T2 石头称重 (stone)

最暴力的方式是将所有可能的重量都求出来，排序后选择第 k 小的。那么该如何优化解法呢？

题目中有一个非常重要的性质：编号为 i 的石头比所有编号小于 i 的石头加起来的重量还要重。意思就是无论在所有编号小于 i 的石头怎么选重量都不如直接选 i 得到的重量重。

这其实与二进制有相同的性质，对于二进制的第 i 位，一定比 $0 \sim i - 1$ 位之和更大。于是我们利用二进制的方式解决此题。

如果 k 的二进制下第 0 位为 1，则选择第 1 块石头；第 1 位为 1，则选择第 2 块石头；依次类推，最终我们就得到了所选的石头集合，将重量相加即可。

T3 字符串 (string)

首先来进行分析，上面情况下字符串满足规则。题意中的规则过于复杂，我们从简单的情况来进行分析。首先对于长度为 2 的连续子串，只要两个字符不同就满足。对于长度为 3 的，要求三个字符各不相同。当考虑长度大于 3 的连续子串时，我们发现只要其内部没有不合法的长度为 3 的子串，它就一定合法。例如 $abaca$ 是不合法的子串，那么其内部一定有长度为 3 的不合法子串，如 aba 。

有了上述性质，我们尝试使用动态规划解决该题。设 $f_{i,j,k}$ 表示前 i 个字符中，第 $i - 1$ 个字符为 j ，第 i 个字符为 k ，且到第 i 个字符为止的串合法。转移时保证下一个字符与前两个字符不同即可。

T4 喵喵 (meow)

首先能看出这是一道最短路题目。由于最终所有的流浪猫都要聚集到点 1，我们运用 Dijkstra 算法求出每个点到点 1 的最短路径改如何走（这里要注意字典序）。

然后来考虑如何求解总时间。先假设不增加捷径。由于所有流浪猫走向点 1 的路径最后会形成一棵树，也就是所谓的最短路径树，我们可以直接建出这棵树，在树上求解总时间。

求解总时间有两种方法。一是对于每一个点，求从这个点到点 1 的距离，再求和。二是考虑最短路径树上的一条边，求出经过这条边的总猫猫数量，也即是子树内的猫猫数量。

此处使用第二种方式更利于后续分析。现在我们思考如何增加捷径。沿用上文的第二种方法，当子树 i 内的所有猫猫都汇聚在点 i ，我们在 $1, i$ 之间建立一条捷径后，每只猫猫会少走 $dis_i - t$ ，其中 dis_i 表示点 i 到点 1 的最短距离。于是，枚举所有点求出节省时间最大值即可。