

20260813模拟赛题解

题号	题目	难度系数	备注
1	乘积	1600	数学、预处理
2	计数	2000	树，数论，动态规划
3	连接城市	2200	构造
4	迷宫	2300	Dijkstra+状压DP优化

T1.乘积

解法分析

对于每一个需要枚举区间的题目，首先考虑的是 n^2 直接枚举；如果数据范围较大，可以考虑通过某种性质优化为 $n\log n$ 的分治算法，比如单调性等。因为本题的范围较小 ($n \leq 5000$)，所以可以直接枚举每个区间。那么，如何统计每个区间的翻转后的结果呢？

首先，不难想到将每个区间内的元素遍历求乘积和，再与原先区间乘积和相减，这样就能找出更改后最佳的值；区间乘积和可以用前缀和 $O(1)$ 查询。核心代码如下：

```
inline void query(int l, int r){
    long long res = 0;
    for(int i=l; i<=r; i++) res += a[i] * b[l+r-i];
    ans = max(ans, res - (sum[r] - sum[l-1]));
}
```

但是，这样做的时间复杂度为：

$$\sum_{i=1}^n i \times (n - i)$$

显然是过不了的，所以要考虑优化，预处理出区间 $[l, r]$ 反转后的贡献。

对于 a 中的任意两个元素 a_i 与 a_j ($i < j$)，他们交换后，对整体和的改变量为：

$$a_i \times b_j + a_j \times b_i - a_i \times b_i - a_j \times b_j$$

即：

$$(a_i - a_j) \times (b_j - b_i)$$

那么我们的思路就来了：

每一个大区间可以视为对应的小区间加上两个端点得到，所以只需要将区间从小到大枚举，再逐渐向外拓展，统计改变量最大的区间即可。

最小区间有两种，分别是**单个元素**和**相邻的两个元素构成的区间**，所以只需要先枚举这两种最小区间，然后我们依次枚举两个元素的三个元素的.....，由于在求 i 个元素的区间， $i - 2$ 个元素的区间一定已经求解好，所以可以直接得到答案。

因为每个区间都被枚举到，每次计算的时间复杂度为 $O(1)$ ，所以整体预处理的时间复杂度为 $O(n^2)$ 。

T2.计数

解法分析

考虑一个比较暴力的做法是直接枚举每一个质因子 p ，然后dfs整棵树，把 a_i 整除 p 的顶点标记为1，那问题就转化为了求最长被标记为1的路径长度。那么我们可以维护每个点从子树内到当前点的最长链的长度，每个点在子树中最长的两条可以对答案贡献，选出最长的一条往父亲继续贡献。

但是这样会超时，因为无用的信息太多，事实上每个数 a_i 最多只有 $\log a_i$ 级别的质因子，所以我们可以对于每一个顶点开一个vector，每一个元素是一个 $pair < x, y >$ ，表示子树内某条链权值全部为 x 的倍数时，链长最大为 y 。那么和暴力类似，我们只需要统计对于每一个 x ，最大和次大的 y 之和对答案贡献即可。注意必须要 x 被 a_i 整除才能往当前顶点 i 贡献，然后把当前顶点的信息往父亲传递即可。因为每个点最多本质不同的 $pair$ 只会有 $\log$ 级别，所以这样就可以保证复杂度正确了。

T3.连接城市 (connect)

解法分析

①假如没有P，那么就是B与B相连，R与R相连。

②假如只有BP，那么当看RP的时候，P是需要连通的，只有RP的时候同理。对于这种情况因此首先我们让P先全部连通，然后我们再对于两个相邻的p，找到中间的B或R，使得前一段B连接第一个P，后一段B连接第二个p，答案的最优解枚举即可得。

③RBP均存在，我们可以先按照②分别对RB做一次答案，对于两个相邻的P之间的R与B的答案是 $p_i - p_{i-1} + ansR + ansB$ ，这时候可能存在另一种方案，就是将PBP，PRP依次连接，答案是 $2 * (p_i - p_{i-1})$ ，因此我们的答案是 $\min(p_i - p_{i-1} + ansR, ansB, 2 * (p_i - p_{i-1}))$

T4.迷宫(maze)

解法分析

对于20%的数据， $n \leq 300, k \leq 10$

将起点和特殊点都称为关键点

用floyd计算出关键点之间及每个关键点到任意一个终点的最小距离，然后搜索即可 $O(n^3 + k!)$ 。

对于50%的数据， $n \leq 1000, k \leq 10$

改用dijkstra算法，枚举所有关键点为起点求单源最短路

$O(kn^2 + k!)$ 。

对于70%的数据， $n \leq 1000, k \leq 20$

状态压缩dp

具体地说，我们令 $f[i][S]$ 表示当前经过的关键点集合为 S ，最终到达 i 点经过的最短距离，这里用二进制数表示集合。

转移很简单。

$O(kn^2 + k^2 * 2^k)$ 。

对于100%的数据， $n \leq 10000, m \leq 50000, k \leq 20$ 。

堆优化dijkstra求单源最短路。

$O(km \log m + k^2 * 2^k)$ 。