题解

定向越野

数据点 1

送分数据点。

数据点 $2 \sim 5$

给各种乱七八糟的做法（

数据点 $6 \sim 10$

首先考虑一个暴力做法：枚举到达目标点的顺序，将路径上这 n 个向量提取出来。此时考虑如何计算答案。通过观察，会发现初始移动方向一定是这 n 个向量（或其旋转 $\frac{\pi}{2}$）之一。（具体证明在结尾处）

由上述观察到的结论可知，初始的移动方向一定平行或垂直与某两点的连线。因此，可以考虑 $O(n^2)$ 枚举两点的连线，以此计算出在该方向下目标点间两两距离。在此基础上，以标准的 $O(n^2 2^n)$ 的 状压dp 即可计算出结果。总复杂度为 $O(n^4 2^n)$ 。

注意分数比较，注意爆 long long 。

引理简要证明

对于这 n 个向量 (d_i, θ_i) ，总距离为 $\sum_{i=1}^{n} d_i * (|\sin(\theta_i - x)| + |cos(\theta_i - x)|)$ 。可以发现该函数分段上凸，所以最小值一定不取在一段中间。

字符串

数据点 1

可以发现串 S 增加一位并不会导致其他位置的权值贡献变化，仅仅会增加新的位置的权值。

所以可以简单枚举 i, j, k 判断，小常数 $O(n^3)$ ，可以通过该数据点。

数据点 $2 \sim 3$

考虑使用 kmp 算法，我们可以枚举当前位置的所有 $border$ ，计算答案。

由于每个位置的 $border$ 数量不超过 n ，故复杂度为 $O(n^2)$ ，可以通过该数据点。

注：出题人没想到这题的离线做法，有好的想法欢迎联系。

数据点 4

由于数据随机，可以发现每个位置 $border$ 数量极少，同用前一个做法即可。

数据点 5

由于题目保证，每个位置 $border$ 数量不超过 $\log_2 n$ 个，同用前一个做法即可。

数据点 $6 \sim 7$

kmp 好像解决不了，怎么办呢?

Z 函数!

其实看到题面第一反应就应该更像是 Z 而不是 kmp （

~~首先我们先复制一个 Z 函数模板~~

然后我们再扫一遍，记录当前位置可用的 b 的和，将其与 a_i 相乘即可。

复杂度 $O(n)$

数据点 $8 \sim 11$

普通的 Z函数 写法好像不支持在线?

让我们考虑在线的 Z函数 。

假设当前处理到第 k 位，z_i 要么已经确定，要么还没确定。

而我们需要计算的就是所有暂未确定的 z_i 所对应的 b_i 的和。

显然对于每个 i ，z_i 只可能由不确定变为确定（而不会由确定变为不确定）。所以朴素地删到以确定的 b_i 是可行的。

问题就变为了如何快速地找到转变的 i

当前转变的点需要满足两个条件：

1. $S[1 \dots n-i] = S[i \dots n-1]$
2. $S[n-i+1] \neq S[n]$

考虑 kmp 构成的 nxt 树

为了满足第一个条件，这个点一定是 $n-1$ 在树上的祖先

为了满足第二个条件，我们可以先记录 f_x 表示 x 的祖先中第一个满足 $S[x+1] \neq S[y+1]$ 的 y。此时我们就可以从 $n-1$ 快速地跳至第一个转变的 i 。

综上，我们完成了一个在线的 Z函数 。套用上一组的做法，即可 $O(n)$ 、在线地完成本题

其他

我们都知道一个字符串的 $border$ 可以视为最多 $O(\log n)$ 个等差数列。而且同一个等差数列（除了其开头项以外）的位置的后继是一样的。所以这题也可以维护这 $O(\log n)$ 个等差数列。

这种做法的复杂度是 $O(n \log n)$ 的，劣于正解复杂度。但是可能更易于理解、也更易于想到，所以出题人出于私心放了 $O(n \log n)$ 的做法。

数数

写在前面的话

这题本来想着不卡常，于是按照 $1e7$ 的范围造数据。结果测试的时候发现最暴力的做法过了倒数第二个 sub……

于是为了卡掉最暴力的做法，数据范围不断增大，最后就成了现在这样。不得不加一句“请相信自己代码的常数，它真的足够小”。

不然也不会有人相信 1000 是给 n^3 的部分分吧……

subtask 1

对于每次询问，用 $O(n^2)$ 的时间复杂度枚举每一个位置，并检测对应数字出现次数是否足够即可。

subtask 1.5

这个检测方法其实存在一些常数上的优化，例如重复的数字不多次检测等等。但是由于没有复杂度的优化，所以没给分。

subtask 2

只看单次询问，假如我们应用了 sub1.5 的方法，会发现对于出现频率很少的数我们都各查询了一次，而出现频率很多（超过 $\frac{1}{3}$ ）的数我们同样的只查询了一次。

那么我们有没有办法多去查询出现频率高的而少去查询出现频率低的呢？

随机化！

我们可以随机区间内的位置，对随机到的数进行查询。这样出现频率高的数更有可能被我们选中来查询。我们只需要随机大约 100 次就可以解决问题，而不需要对区间内的每个数都查询一次。

这个“正确性”怎么证明呢？

以随机 50 次为例：单次询问，最坏的情况下有两个数出现频率恰好超过 $\frac{1}{3}$ ，此时正确率为 $1-2\times\frac{2^{100}}{3^{100}}+\frac{1}{3^{50}}$ ，错误率约为 10^{-16} 。即使考虑单个 subtask 最多也只有大概 10^{-9} 的错误率。你要真这么非那也没办法（

subtask 3

下面郑重介绍 “摩尔投票”！（NOI2022 考过的知识点

假如给你一堆数，让你求哪个数出现次数严格超过一半，该怎么做？

摩尔投票的思路就是每次任取两个不同的数，将他们一起删掉，可以发现所求数一定能够留到最后。因此直接扫一遍，判断一个数是否和目前记录的“剩下的数”一致，记录数量即可。

同样的思路可以拓展到“优势巨大的数”，每次删除三个不相同的数，则“优势巨大的数”一定能留到最后（但留到最后的不一定是优势巨大的数）。

所以每次询问只需 $O(n)$ 的时间，总计 $O(n^2)$ 的复杂度，可以通过该 subtask

subtask 4

在 sub 3 的基础上，我们考虑一下能不能用数据结构来优化。

可以发现，我们并不需要从左往右删除，任意的删除顺序都不影响正确性。

所以我们就有能力做到合并两个区间：判断其剩下的数是否相等，再进行一些删除……

有了合并区间之后，我们就可以用上线段树来优化时间复杂度

总计 $O(n \log n)$ 的时间复杂度，可以通过此题。

其他

可以发现前面的做法并不能保证找到的数一定是优势巨大的，但是~~良心的~~出题人写了 spj，排除掉了这个问题。

那 spj 是如何判断正误的呢？

~~出题是这样的。选手只要在胡出摩尔投票，写个线段树就好了，而出题人(的电脑)出数据要考虑的就很多。~~

~~实际上数据(似乎?)只能拿 $O(n^2)$ 的做法来暴力算，结果就是这样~~

~~这只是两个点啊！~~

出题人出完数据才意识到可以用分块的方法造数据，复杂度 $O(n\sqrt{n})$ ……

逃脱

数据点 1

送分

数据点 2

假设我们将“逃脱者”的初始位置视为树根。

分析原问题的策略：

- 首先，“围堵者”不断向上走是不劣的。
- 其次，“逃脱者”走回头路一定是不优的。

我们可以将按照每个点 “到根的距离” 和 “到任意出口的距离” 的大小关系将点分为两类：

我们称 “到根的距离” $\geq$ "到任意出口的距离" 的点称为 围堵方的“可控制点”

显然，如果一个点是 “可控制点”，那它的子节点也是 “可控制点”

而原题便转化为 “父节点不是可控制点” 的 “可控制点” 个数

由于这些点一定是叶子节点与根的中点，我们可以考虑这些中点，在换根的过程中维护他们的祖先关系。

复杂度：$O(n \times 10^{?})$

数据点 $3 \sim 4$

继续 2 的做法即可。

数据点 $5 \sim 10$

回顾 2 做法，发现叶子个数较多的情况下，我们并不方便维护中点。而其他统计 点对点 的贡献的算法又会被 “父节点不是可控制点” 的要求限制。

那么有没有办法删去 “父节点不是可控制点” 的要求?

一个简单的容斥方案便是将每个点的点权设为 $(1 - \text{子节点个数})$ 可以发现，这样一个子树的权值和为 1

于是可以预处理每个点到出口的距离，使用点分治算法，计算每个点与点之间的贡献。加和即可。

（别忘了特判“逃脱者”初始时处于出口的情况