hash

skip2004

瞎扯

为什么字符串这点东西要讲三天。
讲的到三天有鬼了。
所以你可以看到不知道为什么 hash 这点东西都能有 20 页。

目录

1. union bound

对于一系列事件 $A_1, A_2, A_3, \cdots$：

$$\Pr\left[\cup_{i=1}^{\infty} A_i\right] \leq \sum_{i=1}^{n} \Pr\left[A_i\right]$$

2. Schwartz-Zippel 引理

对于一个**域**中，度数为 d 的多项式 $P(x_1, \cdots, x_n)$:
我们在该域一个子集 S 中均匀独立选取 n 个值 $r_1, \cdots, r_n$。

$$\Pr\left[P(r_1, \cdots, r_n) = 0\right] \le \frac{d}{|S|}$$

常见域

1. 复数域，实数域，有理数域。
2. 一些有限域，常见的有 F_p 等。

hash 常用域

随机一个质数 p，取 F_p。
通过手段扩张得到 $GF(p^k)$。
事实上，域的大小也只能是质数的幂次。
域的特征只能是质数，因此这两种域可以构造出所有我们可能得到的大小与特征。

[P1] 【UNR #7】比特迷宫 的 checker

小青鱼来到了重（zhòng）庆市的一个迷宫，名为比特迷宫。听说只有最聪明的人才能从里面走出。

这个迷宫看似容易，但在小青鱼即将走出迷宫的时候，却被 $n = 2^k$ 个比特机器人拦住了去路。这些机器人从左到右显示着 $a_0, a_1, \cdots, a_{n-1}$，表示一个 $n-1$ 次的多项式 $P(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ 的系数。比特机器人喜欢比特，所以每个系数只可能是 0 或者 1。

小青鱼是无法单独修改某一个比特机器人显示的系数的，但是这个迷宫提供了一个批量修改的操作：输入两个非负整数 a, b $(a + b \le n - 1)$，显示第 i 次项系数的比特机器人会算出 $x^a(1 + x)^b$ 的第 i 次项系数 c_i，然后将自己显示的值 a_i 修改为 $(a_i + c_i) \bmod 2$，其中 $\bmod 2$ 表示对 2 取模。

整体来看，这个操作的作用是将这个多项式 $P(x)$ 在 $\bmod 2$ 意义上加上 $x^a(1 + x)^b$。

由于这个迷宫是困难模式，小青鱼只能操作不超过 T 次。当多项式变为 0 的时候，也即所有比特机器人都显示 0 的时候，他就通关了。

小青鱼左思右想没有想到通关的方式，于是他找到了你来帮忙。

[S1] 【UNR #7】比特迷宫 的 checker

容易发现，我们需要做的事情是判断两个 F_2 下的多项式是否相同。

[S1] 【UNR #7】比特迷宫 的 checker

容易发现，我们需要做的事情是判断两个 F_2 下的多项式是否相同。

判断两个多项式相同其实并非困难的事情，通过 Schwartz-Zippel 引理我们知道，只要带入几个值看一不一样就可以了。

[S1] 【UNR #7】比特迷宫 的 checker

容易发现，我们需要做的事情是判断两个 F_2 下的多项式是否相同。

判断两个多项式相同其实并非困难的事情，通过 Schwartz-Zippel 引理我们知道，只要带入几个值看一不一样就可以了。

但是在这里，我们需要判断的是 F_2 下的多项式，我们就只有两种值，完全不足以我们用于判断两个多项式是否相同。

[S1] 【UNR #7】比特迷宫 的 checker

容易发现，我们需要做的事情是判断两个 F_2 下的多项式是否相同。

判断两个多项式相同其实并非困难的事情，通过 Schwartz-Zippel 引理我们知道，只要带入几个值看一不一样就可以了。

但是在这里，我们需要判断的是 F_2 下的多项式，我们就只有两种值，完全不足以我们用于判断两个多项式是否相同。

仔细分析我们可以得到，只要我们选取特征为 2 的域即可，因此我们可以在 $GF(2^k)$ 中做求值就可以了。

[S1] 【UNR #7】比特迷宫 的 checker

容易发现，我们需要做的事情是判断两个 F_2 下的多项式是否相同。

判断两个多项式相同其实并非困难的事情，通过 Schwartz-Zippel 引理我们知道，只要带入几个值看一不一样就可以了。

但是在这里，我们需要判断的是 F_2 下的多项式，我们就只有两种值，完全不足以我们用于判断两个多项式是否相同。

仔细分析我们可以得到，只要我们选取特征为 2 的域即可，因此我们可以在 $GF(2^k)$ 中做求值就可以了。

我们还可以用 nimbers 来做这件事情。

[S1] 【UNR #7】比特迷宫 的 checker

容易发现，我们需要做的事情是判断两个 F_2 下的多项式是否相同。

判断两个多项式相同其实并非困难的事情，通过 Schwartz-Zippel 引理我们知道，只要带入几个值看一不一样就可以了。

但是在这里，我们需要判断的是 F_2 下的多项式，我们就只有两种值，完全不足以我们用于判断两个多项式是否相同。

仔细分析我们可以得到，只要我们选取特征为 2 的域即可，因此我们可以在 $GF(2^k)$ 中做求值就可以了。

我们还可以用 nimbers 来做这件事情。

单次错误率不超过 $\frac{n}{2^k}$，可以多次运行降低错误率。

字符串哈希 1

给定一个长度为 n 的字符串 s，以及 q 次询问，每次询问给定两个子串 $s[l_1 : r_1], s[l_2 : r_2]$，问是否相同。

字符串哈希 1

我们最常见的 Hash 为：

$$hash(s) = \sum_{i=0}^{len(s)-1} s[i] * base^i \pmod P$$

如果两个字符串的 hash 值相同，我们视为相同，如果 hash 值不同，那么肯定不同。

如果我们在 $[0, P)$ 中均匀随机选取 $base$，试试看分析一次询问的错误率。

字符串哈希 1

如果 P 是质数，我们可以使用 Schwartz-Zippel 引理分析出错误率不超过 $\frac{n}{P}$。

如果不是，请注意危险！

试试看分析 q 次询问的错误率。

字符串哈希 2

给定一个长度为 n 的字符串 s，以及给定 q 个子串 $s[l_i : r_i]$，问有多少本质不同的子串。

如果我们在 $[0, P)$ 中均匀随机选取 $base$，试试看分析询问的错误率。

字符串哈希 3

给定一个长度为 n 的字符串 s，以及 q 次询问，每次询问给定两个子串 $s[l_1 : r_1], s[l_2 : r_2]$，问是否相同。

如果我们在 $[T, 2T)$ 中均匀随机选取 P，试试看分析一次询问的错误率。

在刚刚所有的分析中，我们分析的错误率均有 $O(\frac{n}{P})$ 级别。

在理想情况下，我们有可能做到 $O(\frac{1}{P})$ 的错误率。但是在一些常规的尝试下，我们都只能证明到 $O(\frac{n}{P})$，但是有很多手段让我们实际上难以达到这个错误率。

在下面一个章节，我将讲述一些卡 hash 的手段来帮助大家更好的理解。

合数模数

我们在这个章节将会讲解一些卡将合数作为模数的 hash。
我们假设我们知道选手所采用的模数。

自然溢出

很遗憾，有非常简单的构造可以解决自然溢出哈希。

$len = 1024$

$s_0[i] = popcount(i)\%2?a : b$

$s_1[i] = popcount(i)\%2?b : a$

大家可以自己去试试。

多个小质数的乘积

我们考虑它的因子为 $p_1, \cdots, p_m$。

如果我们有两个长度相同的串 s_1, s_2，使得它们的 hash 值在前 k 个质因子的模意义下都相同：

我们可以将它们作为“字符”，在模 p_{k+1} 的意义下进行碰撞，在小质数下的碰撞我们可以使用生日碰撞等手段。

如此反复进行，我们可以得出结果。

生日碰撞

在一个范围随机生成的数中，约 sqrt 次就会出现相同的数。

因此我们可以大力开随小长度的字符串，求哈希值，祈求发现两个不同的但是哈希值相同的串。

可以用于攻击 10^{15} 左右范围的模数

缺点是空间需求很大，我们是否有办法改进？

生日碰撞

在一个范围随机生成的数中，约 sqrt 次就会出现相同的数。

因此我们可以大力开随小长度的字符串，求哈希值，祈求发现两个不同的但是哈希值相同的串。

可以用于攻击 10^{15} 左右范围的模数

缺点是空间需求很大，我们是否有办法改进？

可以使用 floyd 找环法改进空间。

(Multi-) Tree attack

内容有点复杂，现场讲。

多模数 Hash

我们考虑它的模数为 $p_1, \cdots, p_m$。

如果我们有两个长度相同的串 s_1, s_2，使得它们的 hash 值在前 k 个模数的模意义下都相同：

我们可以将它们作为“字符”，在模 p_{k+1} 的意义下进行碰撞，在单个质数下的碰撞我们可以使用 Multi-tree attack 等手段。

如此反复进行，我们可以得出结果。

小 base 攻击

即使我们模数很大，但是如果 $base$ 只比字符集略大一点的话：

我们可以随机一个字符串 s，求 $hash(s)$，转为 $base$ 进制，很大概率每一位都是合法的。

卡掉了！

[P2]【SPC #2】美丽的序列 Beautiful Sequence

小 ω 定义美丽的数字为在一个区间中，它出现了偶数次；小 ω 又定义了美丽的区间，一个区间是美丽的当且仅当它里面所有出现过的数都是美丽的；然后小 ω 定义了连续序列的美丽值，也就是这个序列中有多少**连续**子序列是美丽的。

所以小 ω 给出一个序列，求它的美丽值。

但小 ω 觉得这题太水了，于是小 ω 又加了一个多次询问：每次给出一个区间 $[l,r]$，询问原序列的连续子序列 $S[l..r]$ 的美丽值。

限制与约定

$1 \le N,Q \le 10^5; S_i \in [1,10^6]$。

时间限制：1s

空间限制：512MB

[S2]【SPC #2】美丽的序列 Beautiful Sequence

给每个数映射一个随机值。区间所有数出现偶数次一定异或和为 0。
试着分析正确率。
试着设计后续算法。

[P3] 「THUSCH 2017」巧克力

题目描述

「人生就像一盒巧克力，你永远不知道吃到的下一块是什么味道。」

明明收到了一大块巧克力，里面有若干小块，排成 n 行 m 列。每一小块都有自己特别的图案 $c_{i,j}$，它们有的是海星，有的是贝壳，有的是海螺……其中还有一些因为挤压，已经分辨不出是什么图案了。明明给每一小块巧克力标上了一个美味值 $a_{i,j}$（$0 \le a_{i,j} \le 10^6$），这个值越大，表示这一小块巧克力越美味。

正当明明咽了咽口水，准备享用美味时，舟舟神奇地出现了。看到舟舟恳求的目光，明明决定从中选出一些小块与舟舟一同分享。

舟舟希望这些被选出的巧克力是连通的（两块巧克力连通当且仅当他们有公共边），而且这些巧克力要包含至少 k（$1 \le k \le 5$）种。而那些被挤压过的巧克力则是不能被选中的。

明明想满足舟舟的愿望，但他又有点「抠」，想将美味尽可能多地留给自己。所以明明希望选出的巧克力块数能够尽可能地少。如果在选出的块数最少的前提下，美味值的中位数（我们定义 n 个数的中位数为第 $\lfloor \frac{n+1}{2} \rfloor$ 小的数）能够达到最小就更好了。

你能帮帮明明吗？

$$1 \le n \times m \le 233$$

[S3] 「THUSCH 2017」巧克力

给一个每种巧克力到 $[1,5]$ 的随机映射。

然后发现最优解五种颜色映射后不同的概率是 $\frac{5!}{5^5}$，或者我们大约可以看成 e^5。

后面用二分状压的做法，搞就完了。