随机算法

王蔚澄

2025 年 2 月 7 日

什么是随机算法

所谓随机算法，是指允许算法读取一系列随机比特，并希望在平均情况具有正确性或更优的性能. 因此，算法输出和算法运行时间都是关于这一系列随机比特的随机变量.

运行时间有保证，但可能输出错误结果的算法称为 Monte Carlo 算法. 相反，答案总是正确，但仅保证期望运行时间有限的算法称为 Las Vegas 算法.

Monte Carlo 算法

此时，我们考虑有可能输出错误结果的算法，更具体的，错误结果被分为**单侧错误**和**双侧错误**.

Monte Carlo 算法

此时，我们考虑有可能输出错误结果的算法，更具体的，错误结果被分为**单侧错误**和**双侧错误**.

问题 (单侧错误)

假设采用重复运行的策略，设单次运行的错误概率为 $\delta > 0$，我们想要错误概率至多 $\varepsilon > 0$，应该重复运行几次？

Monte Carlo 算法

此时，我们考虑有可能输出错误结果的算法，更具体的，错误结果被分为**单侧错误**和**双侧错误**.

问题 (单侧错误)

假设采用重复运行的策略，设单次运行的错误概率为 $\delta > 0$，我们想要错误概率至多 $\varepsilon > 0$，应该重复运行几次？

解

$t = \log_\delta \varepsilon = O(\log(1/\varepsilon))$.

这告诉我们，对一个常数错误概率的算法，我们只需要付出重复不太多次的代价，就可以转化为一个几乎总是正确的算法.

Monte Carlo 算法

问题 (双侧错误)

假设采用重复运行取众数的策略，设单次运行的错误概率为 $1/2 - \delta > 0$，我们想要错误概率至多 $\varepsilon > 0$，应该重复运行几次？

Monte Carlo 算法

问题 (双侧错误)

假设采用重复运行取众数的策略, 设单次运行的错误概率为 $1/2 - \delta > 0$, 我们想要错误概率至多 $\varepsilon > 0$, 应该重复运行几次?

解

假设运行 t 次, 有至多 $t/2$ 次正确, 概率是

$$\begin{aligned}\sum_{i=0}^{t/2} \binom{t}{i} \left(\frac{1}{2}+\delta\right)^i \left(\frac{1}{2}-\delta\right)^{t-i} &\le 2^t \left(\frac{1}{2}+\delta\right)^{t/2} \left(\frac{1}{2}-\delta\right)^{t/2} \\ &\le (1-4\delta^2)^{t/2}.\end{aligned}$$

Monte Carlo 算法

问题 (双侧错误)

假设采用重复运行取众数的策略, 设单次运行的错误概率为 $1/2 - \delta > 0$, 我们想要错误概率至多 $\varepsilon > 0$, 应该重复运行几次?

解

重复运行 t 次的错误概率 $\leq (1 - 4\delta^2)^{t/2}$.
若要 $(1 - 4\delta^2)^{t/2} < \varepsilon$, 只需 $t = O(\log(1/\varepsilon))$

这告诉我们, 修正双侧错误概率也不困难, 但事实上, 我们很少遇到需要修正双侧错误的情况.

CF1310D

问题

给定一个 N 个点的带权完全图, 求 1 号点经过恰好 K 条边回到 1 号点, 且路径上没有奇环的最短路.

$N \leq 80, K \leq 10.$

CF1310D

问题

给定一个 N 个点的带权完全图，求 1 号点经过恰好 K 条边回到 1 号点，且路径上没有奇环的最短路.
$N \leq 80, K \leq 10.$

解

注意到仅有偶环等价于是二分图. 将所有点随机二染色，限制只能走两边点颜色不一样的边. 此时可以 $O(N^2K)$ 得到答案.

假设固定 1 号点的颜色，一次随机的错误率是 $\delta = 1 - 2^{-K+1}$. 根据先前的分析，只需要 $T = \log_\delta \varepsilon$ 次随机. 取 $\varepsilon = 10^{-6}$ 时 $T \approx 7000.$

CF1305F

问题

给定 n 个数，每次可以选择一个数 $+1$ 或 -1，问最少需要多少次操作使得所有数都是正数且所有数的 $\gcd > 1$.
$n \leq 2 \times 10^5, a_i \leq 10^{12}$.

CF1305F

问题

给定 n 个数，每次可以选择一个数 $+1$ 或 -1，问最少需要多少次操作使得所有数都是正数且所有数的 $\gcd > 1$.
$n \leq 2 \times 10^5, a_i \leq 10^{12}$.

解

答案 $\leq n$，这是因为可以把所有数都变成偶数.

假如我们知道最终 gcd 的一个质因数，可以 $O(n)$ 知道答案.

CF1305F

问题

给定 n 个数, 每次可以选择一个数 $+1$ 或 -1, 问最少需要多少次操作使得所有数都是正数且所有数的 $\gcd > 1$.
$n \leq 2 \times 10^5, a_i \leq 10^{12}$.

解

答案 $\leq n$ 说明有至少 $n/2$ 个数操作次数 ≤ 1.
a_i 的不同质因数最多只有 11 个. 随机选取 t 个数并测试, 总复杂度为 $O(t\sqrt{V} + 11tn)$, 取 $t = 50$ 即得错误概率 2^{-50} 的算法.

LOJ 3400

问题

给定一张 N 个点 M 条边的图，每个点有点权 a_i，每条边有边权 w_i. 你需要选一个边集 S，满足 $|S| \leq K$，你需要最大化

$$\sum_{v \in N(S)} a_v - \sum_{e \in S} w_e,$$

其中 $N(S)$ 表示所有边的端点的并.
$2^K(N+M) \leq 10^6$.

LOJ 3400

问题

给定一张 N 个点 M 条边的图，每个点有点权 a_i，每条边有边权 w_i. 你需要选一个边集 S，满足 $|S| \leq K$，你需要最大化

$$\sum_{v \in N(S)} a_v - \sum_{e \in S} w_e,$$

其中 $N(S)$ 表示所有边的端点的并.
$2^K(N+M) \leq 10^6$.

解

观察：最终的结果一定是若干个菊花.

LOJ 3400

解

假设图是二分图, 考虑如下费用流算法:

- 对每个左部点 i, 向源点连流量为 1, 费用为 a_i 的边. 和流量为 ∞, 费用为 0 的边.
- 对每个右部点 i, 向汇点连流量为 1, 费用为 a_i 的边. 和流量为 ∞, 费用为 0 的边.
- 对每条边 (u_i, v_i, w_i), 在两点之间连流量为 1, 费用为 $-w_i$ 的边.

求限制流量 $\leq K$ 的最大费用流即可.

可以证明, 这部分的时间复杂度为 $O(K^2(N+M))$.

LOJ 3400

问题

$2^K(N+M) \le 10^6$.

解

随机将每个点分到一边，错误率为 $\delta = 1 - 2^{-K+1}$.
类似之前的推导需要的重复次数为 $\log_\delta \varepsilon \approx 2^K \log(1/\varepsilon)$.

什么是哈希

什么是哈希

对于一个组合对象集合 S, 我们希望构造一个它到某个有限集 (如模 m 同余类) 的映射 $f : S \to \mathbb{Z}_m$. 并且满足以下条件:

什么是哈希

对于一个组合对象集合 S, 我们希望构造一个它到某个有限集 (如模 m 同余类) 的映射 $f : S \to \mathbb{Z}_m$. 并且满足以下条件:

- 对任意 $0 \leq i < m$, $\Pr(f(x) = i) = 1/m$. 即哈希函数是均匀的.
- 对每个 $x \in S$, $f(x)$ 与其他 S 中的元素无关. 即哈希函数是独立的.

什么是哈希

对于一个组合对象集合 S, 我们希望构造一个它到某个有限集 (如模 m 同余类) 的映射 $f : S \to \mathbb{Z}_m$. 并且满足以下条件:

- 对任意 $0 \leq i < m$, $\Pr(f(x) = i) = 1/m$. 即哈希函数是均匀的.
- 对每个 $x \in S$, $f(x)$ 与其他 S 中的元素无关. 即哈希函数是独立的.
- f 能被快速计算.

什么是哈希

对于一个组合对象集合 S, 我们希望构造一个它到某个有限集 (如模 m 同余类) 的映射 $f : S \to \mathbb{Z}_m$. 并且满足以下条件:

- 对任意 $0 \leq i < m$, $\Pr(f(x) = i) = 1/m$. 即哈希函数是均匀的.
- 对每个 $x \in S$, $f(x)$ 与其他 S 中的元素无关. 即哈希函数是独立的.
- f 能被快速计算.

但是我们想的好像太美了, 这些条件很难满足.

什么是哈希

对于一个组合对象集合 S, 我们希望构造一个它到某个有限集 (如模 m 同余类) 的映射 $f : S \to \mathbb{Z}_m$.

仔细想一想我们到底需要什么.

什么是哈希

对于一个组合对象集合 S, 我们希望构造一个它到某个有限集 (如模 m 同余类) 的映射 $f : S \to \mathbb{Z}_m$.

仔细想一想我们到底需要什么.

通常来说, 只需要比较两个 S 中的元素是否相同. 这意味着, 我们只需要

- 对不同的 $x, y \in S$, 以高概率 $f(x) \neq f(y)$.
- f 能被快速计算.

注意: 判断一个集合 T 中有几个不同的元素, 实际上做了 $\binom{|T|}{2}$ 次 "比较两个元素是否相同".

最简单的情形

问题

给定两个长为 n 的字符串 s, t, 判断 s 和 t 是否相同.
$n \leq 10^6$.

最简单的情形

问题

给定两个长为 n 的字符串 s, t, 判断 s 和 t 是否相同.
$n \leq 10^6$.

解

考虑如下哈希函数: 对固定的素数 M, 随机生成
$a_0, \ldots, a_{n-1} \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} a_i s_i \bmod M$.

最简单的情形

问题

给定两个长为 n 的字符串 s, t, 判断 s 和 t 是否相同.
$n \leq 10^6$.

解

考虑如下哈希函数: 对固定的素数 M, 随机生成 $a_0, \ldots, a_{n-1} \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} a_i s_i \bmod M$.

错误概率

如果 s 和 t 不同, 则 $h(s) = h(t)$ 相当于 $\sum_{i=0}^{n-1} a_i(s_i - t_i) \equiv 0 \pmod{M}$.

最简单的情形

解

考虑如下哈希函数: 对固定的素数 M, 随机生成 $a_0, \ldots, a_{n-1} \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} a_i s_i \bmod M$.

错误概率

如果 s 和 t 不同, 则 $h(s) = h(t)$ 相当于 $\sum_{i=0}^{n-1} a_i(s_i - t_i) \equiv 0 \pmod{M}$.
考虑最后一个不同的位置 i, 即可得到错误概率为 $1/M$.

一般的情形

问题

给定长为 n 的字符串 s, 和长为 m 的字符串 t 判断 s 是否是 t 的子串. $n < m \leq 10^6$.

一般的情形

问题

给定长为 n 的字符串 s, 和长为 m 的字符串 t 判断 s 是否是 t 的子串. $n < m \leq 10^6$.

解 (Rolling Hash)

考虑如下哈希函数: 对固定的素数 M, 随机生成 $x \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} s_i x^i \bmod M$.

一般的情形

问题

给定长为 n 的字符串 s, 和长为 m 的字符串 t 判断 s 是否是 t 的子串. $n < m \leq 10^6$.

解 (Rolling Hash)

考虑如下哈希函数: 对固定的素数 M, 随机生成 $x \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} s_i x^i \bmod M$.

错误概率

如果 s 和 t 不同, 则 $h(s) = h(t)$ 相当于 $\sum_{i=1}^{n}(s_i - t_i)x^i \equiv 0 \pmod{M}$.

一般的情形

解 (Rolling Hash)

考虑如下哈希函数: 对固定的素数 M, 随机生成 $x \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} s_i x^i \bmod M$.

错误概率

如果 s 和 t 不同, 毛估估一下, 错误概率很可能是 $1/M$.

一般的情形

解 (Rolling Hash)

考虑如下哈希函数: 对固定的素数 M, 随机生成 $x \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} s_i x^i \bmod M$.

错误概率

如果 s 和 t 不同, 毛估估一下, 错误概率很可能是 $1/M$.

这对吗?

一般的情形

解 (Rolling Hash)

考虑如下哈希函数: 对固定的素数 M, 随机生成 $x \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} s_i x^i \bmod M$.

错误概率

如果 s 和 t 不同, 代数基本定理说明 n 次多项式至多只有 n 个根, 所以错误概率 $\leq n/M$.

一般的情形

解 (Rolling Hash)

考虑如下哈希函数: 对固定的素数 M, 随机生成 $x \in [0, M)$, $h(s) = \sum_{i=0}^{n-1} s_i x^i \bmod M$.

错误概率

如果 s 和 t 不同, 代数基本定理说明 n 次多项式至多只有 n 个根, 所以错误概率 $\leq n/M$.

事实上, 这一错误概率对一些素数是紧的, 考虑 $s = 1 \ldots 12, t = 21 \ldots 1$, $h(s) - h(t) = x^{n-1} - 1$, 如果模 M 意义下存在 $n-1$ 次单位根, 则确实会有 $n-1$ 个根.

树哈希

问题

给一颗有根树 T, 问它的所有子树有多少种不同构的树.
$n \leq 10^6$.

树哈希

问题

给一颗有根树 T, 问它的所有子树有多少种不同构的树.
$n \leq 10^6$.

分析

事实上, 我们想要构造一种把所有儿子的哈希值再哈希的方法.

集合哈希

问题

给定两个可重集 S, T, 判断两个集合是否相同.
$n \leq 10^6$.

集合哈希

问题

给定两个可重集 S, T, 判断两个集合是否相同.
$n \leq 10^6$.

解

一个简单的想法是先排序, 然后做字符串哈希.

集合哈希

问题

给定两个可重集 S, T, 判断两个集合是否相同.
$n \leq 10^6$.

解

一个简单的想法是先排序, 然后做字符串哈希.

能不能避免排序?

集合哈希

问题

给定两个可重集 S, T, 判断两个集合是否相同.
$n \leq 10^6$.

解

考虑如下哈希函数: 对固定的素数 M, 随机生成 $x \in [0, M)$,
$h(S) = \prod_{s \in S}(x + s) \bmod M$.

错误概率

类似的, 代数基本定理说明 n 次多项式至多只有 n 个根, 所以错误概率 $\leq n/M$.

树哈希

问题

给一颗有根树 T，问它的所有子树有多少种不同构的树.

分析

我们想要模仿先前的集合哈希，并且做一些适当的调整. 事实上，我们总是在想办法构造一个多项式并对其随机代值.

树哈希

问题

给一颗有根树 T, 问它的所有子树有多少种不同构的树.

分析

我们想要模仿先前的集合哈希, 并且做一些适当的调整. 事实上, 我们总是在想办法构造一个多项式并对其随机代值.

解

注意此时我们不能只维护一元多项式, 需要在多项式变元中加入子树大小或深度的信息.
每个点维护一个多项式 f_u 以及它和下面最深的叶子的距离 d, 则点 u 的多项式为 $f_u = \prod_{v \in S_u} (x_d + f_v)$.

正确率分析

定理 (Union Bound)

对时间 A, B, $\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$.

正确率分析

定理 (Schwartz-Zippel)

对域上的多元多项式 $P(x_1, \ldots, x_n)$, 次数为 d, S 是域的一个子集, 所有 r_i 在 S 中随机选取, 则 $\Pr(P(r_1, \ldots, r_n) = 0) \leq d/|S|$.

正确率分析

定理 (Schwartz-Zippel)

对域上的多元多项式 $P(x_1, \ldots, x_n)$, 次数为 d, S 是域的一个子集, 所有 r_i 在 S 中随机选取, 则 $\Pr(P(r_1, \ldots, r_n) = 0) \leq d/|S|$.

证明.

考虑数学归纳法, 当 $n = 1$ 时归结于代数基本定理.

正确率分析

定理 (Schwartz-Zippel)

对域上的多元多项式 $P(x_1, \ldots, x_n)$, 次数为 d, S 是域的一个子集, 所有 r_i 在 S 中随机选取, 则 $\Pr(P(r_1, \ldots, r_n) = 0) \leq d/|S|$.

证明.

考虑数学归纳法, 当 $n = 1$ 时归结于代数基本定理.
否则假设 $P = \sum_{i=0}^{d} x_1^i P_i(x_2, \ldots, x_n)$, 考虑最大的 i 使得 $P_i \neq 0$, 则 $\Pr(P = 0) \leq \Pr(P_i = 0) + \Pr(P_i \neq 0) \cdot (i/|S|)$.
使用归纳假设即得 $\Pr(P = 0) \leq d/|S|$. □

正确率分析

回到之前的树哈希算法.

解

每个点维护一个多项式 f_u 以及它和下面最深的叶子的距离 d, 则点 u 的多项式为 $f_u = \prod_{v \in S_u} (x_d + f_v)$.

正确率分析

回到之前的树哈希算法.

解

每个点维护一个多项式 f_u 以及它和下面最深的叶子的距离 d, 则点 u 的多项式为 $f_u = \prod_{v \in S_u} (x_d + f_v)$.

定理

对两棵不同构的树, 对固定的模数 M, 随机生成 $x_1, \ldots, x_n$, 哈希冲突的概率 $\leq n/M$.

正确率分析

问题

给一颗有根树 T, 问它的所有子树有多少种不同构的树.

正确率分析

问题

给一颗有根树 T, 问它的所有子树有多少种不同构的树.

此时, 我们能输出正确答案需要对 $\binom{n}{2}$ 个比较都不冲突. 使用 Union Bound 可以得到如下结论.

定理

对固定的模数 M, 随机生成 $x_1, \ldots, x_n$, 所有哈希均冲突的概率 $\leq O(n^3/M)$.

无根树的哈希

问题

给两颗无根树 T_1, T_2, 判断他们是否同构.
$n \leq 10^6$.

无根树的哈希

问题

给两颗无根树 T_1, T_2, 判断他们是否同构.
$n \leq 10^6$.

解

对无根树哈希时, 可以取重心作为根再做有根树哈希. 如果有两个重心, 可以分别做哈希再合并; 或者直接在每条边上再加一个点.

QOJ 9774

问题

你需要维护一个整数序列 $a_1, \ldots, a_n$, 支持如下两个操作

- 给定 L, R, v, 对 $i \in [L, R]$ 执行 $a_i \leftarrow a_i + v$.
- 给定 L, R, 判断 $a_L, \ldots, a_R$ 是否能分成 $(R - L + 1)/2$ 个和相等的 *pair*.

QOJ 9774

问题

你需要维护一个整数序列 $a_1, \ldots, a_n$, 支持如下两个操作

- 给定 L, R, v, 对 $i \in [L, R]$ 执行 $a_i \leftarrow a_i + v$.
- 给定 L, R, 判断 $a_L, \ldots, a_R$ 是否能分成 $(R - L + 1)/2$ 个和相等的 *pair*.

解

事实上, 我们需要判断区间出现次数序列是否是回文的.

QOJ 9774

问题

你需要维护一个整数序列 $a_1, \ldots, a_n$, 支持如下两个操作

- 给定 L, R, v, 对 $i \in [L, R]$ 执行 $a_i \leftarrow a_i + v$.
- 给定 L, R, 判断 $a_L, \ldots, a_R$ 是否能分成 $(R - L + 1)/2$ 个和相等的 *pair*.

解

事实上, 我们需要判断区间出现次数序列是否是回文的.
利用经典的 *Rolling Hash*, 实际上我们需要维护 $\sum_{i=L}^{R} x^{a_i}$, 使用线段树即可.

QOJ 5107

问题

给定一个 $n_1 \times m_1$ 的二维非负整数方阵 a_{ij} 和一个 $n_2 \times m_2$ 的二维正整数方阵 b_{ij}, 你需要对每个可能的 p, q, 判断将 a 中的所有 0 视作通配符后与 $b[p, p + n_1 - 1][q, q + m_1 - 1]$ 是否匹配.

$n_i, m_i \leq 1000$.

QOJ 5107

问题

给定一个 $n_1 \times m_1$ 的二维非负整数方阵 a_{ij} 和一个 $n_2 \times m_2$ 的二维正整数方阵 b_{ij}, 你需要对每个可能的 p, q, 判断将 a 中的所有 0 视作通配符后与 $b[p, p + n_1 - 1][q, q + m_1 - 1]$ 是否匹配.
$n_i, m_i \leq 1000$.

解

固定模数 M, 对一个位置 p, q, 将哈希函数设为 $h(p, q) = \sum_{i=0}^{n_1} \sum_{j=0}^{m_1} w_{i,j} a_{i,j} (a_{i,j} - b_{p+i,q+j})$, 其中 w_{ij} 随机生成.
考虑不同的位置即可得到单次判断错误概率 $1/M$.
使用 *Union Bound* 合并错误概率, 即得总错误概率 $O(nm/M)$.

QOJ 5107

解

固定模数 M, 对一个位置 p, q, 将哈希函数设为 $h(p,q) = \sum_{i=0}^{n_1} \sum_{j=0}^{m_1} w_{i,j} a_{i,j} (a_{i,j} - b_{p+i,q+j})$, 其中 w_{ij} 随机生成. 考虑不同的位置即可得到错误概率 $1/M$.

为了计算这个哈希函数, 需要将 a 数组补全到 m_2 列, 然后利用 *FFT* 加速计算.

Hash Killer 1

问题

在上面的 *Rolling Hash* 中, 模数改为 2^{64}, 请设计输入证明其错误. 你需要同时卡掉所有底数, 即哈希得到的的多项式相同.

[1]事实上是交换整环

Hash Killer 1

问题

在上面的 *Rolling Hash* 中, 模数改为 2^{64}, 请设计输入证明其错误. 你需要同时卡掉所有底数, 即哈希得到的的多项式相同.

Hint

代数基本定理只在域[1]上成立, 模 2^{64} 在哪里出了问题.

[1]事实上是交换整环

Hash Killer 1

解

当底数 x 为偶数时, $x^{64} \equiv 0 \pmod 2^{64}$. 所以只要最后 64 个字符都相同就能卡掉偶数底数.

Hash Killer 1

解

当底数 x 为偶数时，$x^{64} \equiv 0 \pmod 2^{64}$. 所以只要最后 64 个字符都相同就能卡掉偶数底数.
当底数 x 为奇数时，考虑如下构造 $s_0 = 0, s_i = s_{i-1} \circ \hat{s_{i-1}}$. 其中 $\hat{s}$ 表示把 s 的所有 $0, 1$ 反转.

Hash Killer 1

解

当底数 x 为偶数时, $x^{64} \equiv 0 \pmod 2^{64}$. 所以只要最后 64 个字符都相同就能卡掉偶数底数.
当底数 x 为奇数时, 考虑如下构造 $s_0 = 0, s_i = s_{i-1} \circ \hat{s_{i-1}}$. 其中 $\hat{s}$ 表示把 s 的所有 $0, 1$ 反转.
$h(s_i) = h(s_{i-1})x^{2^{i-1}} + h(\hat{s_{i-1}}), h(\hat{s_i}) = h(\hat{s_{i-1}})x^{2^{i-1}} + h(s_i),$
$h(s_i) - h(\hat{s_i}) = (h(s_{i-1}) - h(\hat{s_{i-1}}))(x^{2^{i-1}} - 1).$

Hash Killer 1

解

当底数 x 为偶数时, $x^{64} \equiv 0 \pmod 2^{64}$. 所以只要最后 64 个字符都相同就能卡掉偶数底数.

当底数 x 为奇数时, 考虑如下构造 $s_0 = 0, s_i = s_{i-1} \circ \hat{s_{i-1}}$. 其中 $\hat{s}$ 表示把 s 的所有 $0, 1$ 反转.

$h(s_i) = h(s_{i-1})x^{2^{i-1}} + h(\hat{s_{i-1}})$, $h(\hat{s_i}) = h(\hat{s_{i-1}})x^{2^{i-1}} + h(s_i)$,

$h(s_i) - h(\hat{s_i}) = (h(s_{i-1}) - h(\hat{s_{i-1}}))(x^{2^{i-1}} - 1)$.

注意到 $2^i \mid (x^{2^{i-1}} - 1)$, 取 $t = 12$ 即得 $2^{64} \mid h(s_t) - h(\hat{s_t})$.

QOJ 8340

问题

给定 n 个整数 a_i 和 $M = 10^K - 1$, 问有多少三元组 $1 \le i \le j \le k$ 满足 $a_i + a_j + a_k \equiv 0 \pmod{M}$.
$n \le 500, K \le 20000, 0 \le a_i \le 10^{20000}$.

Hint

$\le n$ 的素数个数 $\pi(n) = O(n/\ln(n))$.

QOJ 8340

解

先把每个数调整到 $[0, M)$ 之间, 此时 $a + b + c \equiv 0 \pmod{M}$ 相当于加起来等于 $0, M, 2M$ 中的一个. 考虑使用随机模数判断是否相等.

假设我们现在要判断 a, b 是否相等. 如果在 $\leq T$ 的素数中随机选一个 p 作为模数, 则出现错误意味着 $p \mid |a - b|$. 而 $|a - b|$ 的素因子只有至多 $O(K)$ 个. 这意味着一次判断的错误概率为 $K/\pi(T)$.

QOJ 8340

解

先把每个数调整到 $[0, M)$ 之间，此时 $a + b + c \equiv 0 \pmod{M}$ 相当于加起来等于 $0, M, 2M$ 中的一个. 考虑使用随机模数判断是否相等.

假设我们现在要判断 a, b 是否相等. 如果在 $\leq T$ 的素数中随机选一个 p 作为模数，则出现错误意味着 $p \mid |a - b|$. 而 $|a - b|$ 的素因子只有至多 $O(K)$ 个. 这意味着一次判断的错误概率为 $K/\pi(T)$.

使用 *Union Bound* 合并 $O(n^3)$ 次比较，再代入素数个数定理，得到总的错误概率为 $n^3 K \ln(T)/T$. 取 $T = 10^{18}$ 可以得到合理的错误概率.

随机算法 vs. 启发式算法

我们之前讨论的随机算法总是对最坏情况也有正确率或运行时间的保证, 但启发式算法则仅保证在一般情况 (如数据随机生成) 时正确或有运行时间的保证.

QOJ 8673

问题

有一个 n 个点 m 条边的带权有向图，q 次询问两点之间最短路长度. 保证边的端点和边权都独立随机生成，询问的两点也独立随机生成.

$n \leq 2 \times 10^5, m \leq 3 \times 10^6, q \leq 10^4$.

QOJ 8673

问题

有一个 n 个点 m 条边的带权有向图，q 次询问两点之间最短路长度. 保证边的端点和边权都独立随机生成，询问的两点也独立随机生成.

$n \leq 2 \times 10^5, m \leq 3 \times 10^6, q \leq 10^4$.

Hint

使用双向 Dijkstra.

QOJ 8673

问题

有一个 n 个点 m 条边的带权有向图，q 次询问两点之间最短路长度. 保证边的端点和边权都独立随机生成，询问的两点也独立随机生成.

解

使用双向 *Dijkstra*，每次从两个堆中取较小的更新，称出堆的点的距离被确认过. 对询问 u, v，假设在点 x 相遇，得到答案的上界 $d_{u,x} + d_{x,v}$，且此时真的最短路一定不经过没有确认过的点.

QOJ 8673

问题

有一个 n 个点 m 条边的带权有向图，q 次询问两点之间最短路长度. 保证边的端点和边权都独立随机生成，询问的两点也独立随机生成.

解

使用双向 *Dijkstra*，每次从两个堆中取较小的更新，称出堆的点的距离被确认过. 对询问 u, v，假设在点 x 相遇，得到答案的上界 $d_{u,x} + d_{x,v}$，且此时真的最短路一定不经过没有确认过的点.
枚举确认过的点，扫所有出边计算答案.

QOJ 8673

分析

回忆 Dijkstra 的过程，在堆里的候选比较只与距离有关，这说明每次更新时的点是在 $[1, n]$ 中均匀随机的. 同时，两边的 Dijkstra 也是独立的.

QOJ 8673

分析

回忆 Dijkstra 的过程，在堆里的候选比较只与距离有关，这说明每次更新时的点是在 $[1, n]$ 中均匀随机的. 同时，两边的 Dijkstra 也是独立的.

问题

每次从两个集合中随机选一个，随机加入一个新的点，什么时候两个集合有交？

QOJ 8673

分析

回忆 Dijkstra 的过程，在堆里的候选比较只与距离有关，这说明每次更新时的点是在 $[1, n]$ 中均匀随机的. 同时，两边的 Dijkstra 也是独立的.

问题

每次从两个集合中随机选一个，随机加入一个新的点，什么时候两个集合有交？

分析

由生日悖论，加入 $O(\sqrt{n})$ 次就会冲突. 同时生日悖论也说明两个集合内部的冲突只有 $O(1)$ 次.

QOJ 8673

分析

由生日悖论, 加入 $O(\sqrt{n})$ 次就会冲突. 同时生日悖论也说明两个集合内部的冲突只有 $O(1)$ 次.
这说明, 如果我们在 Dijkstra 的时候把一个点的出边排序, 并按顺序加入堆, 一共只会扫到 $O(\sqrt{n})$ 个点和 $O(\sqrt{n})$ 条边.

如果我们暴力扫描所有出边, 每个点期望有 $O(m/n)$ 条出边, 总复杂度为 $O(q\sqrt{n}\log n + qm/\sqrt{n} + m\log m)$, 最后一部分来自给边排序.

QOJ 8673

分析

由生日悖论, 加入 $O(\sqrt{n})$ 次就会冲突. 同时生日悖论也说明两个集合内部的冲突只有 $O(1)$ 次.
这说明, 如果我们在 Dijkstra 的时候把一个点的出边排序, 并按顺序加入堆, 一共只会扫到 $O(\sqrt{n})$ 个点和 $O(\sqrt{n})$ 条边.

如果我们暴力扫描所有出边, 每个点期望有 $O(m/n)$ 条出边, 总复杂度为 $O(q\sqrt{n}\log n + qm/\sqrt{n} + m\log m)$, 最后一部分来自给边排序.

还能再给力一点吗?

QOJ 8673

分析

回到之前的最短路: 对询问 u, v, 假设在点 x 相遇, 得到答案的上界 $d_{u,x} + d_{x,v}$. 对一条可能更新答案的边 $a \to b$, 我们需要 $d_{u,a} + e_{a,b} + d_{b,v} \le d_{u,x} + d_{x,v}$. 即 $e_{a,b} \le d_{u,x} - d_{u,a} + d_{x,v} - d_{b,v} \le 2\max(d_{u,x} - d_{u,a}, d_{x,v} - d_{b,v})$.

QOJ 8673

分析

回到之前的最短路: 对询问 u, v, 假设在点 x 相遇, 得到答案的上界 $d_{u,x} + d_{x,v}$. 对一条可能更新答案的边 $a \to b$, 我们需要 $d_{u,a} + e_{a,b} + d_{b,v} \le d_{u,x} + d_{x,v}$. 即 $e_{a,b} \le d_{u,x} - d_{u,a} + d_{x,v} - d_{b,v} \le 2\max(d_{u,x} - d_{u,a}, d_{x,v} - d_{b,v})$.

对一个 u 侧的点 a, 只需要扫到权值 $\le 2(d_{u,x} - d_{u,a})$ 的边.

QOJ 8673

分析

回到之前的最短路: 对询问 u, v, 假设在点 x 相遇, 得到答案的上界 $d_{u,x} + d_{x,v}$. 对一条可能更新答案的边 $a \to b$, 我们需要 $d_{u,a} + e_{a,b} + d_{b,v} \le d_{u,x} + d_{x,v}$. 即 $e_{a,b} \le d_{u,x} - d_{u,a} + d_{x,v} - d_{b,v} \le 2\max(d_{u,x} - d_{u,a}, d_{x,v} - d_{b,v})$.

对一个 u 侧的点 a, 只需要扫到权值 $\le 2(d_{u,x} - d_{u,a})$ 的边.

对一个 u 侧的点 a, 权值 $\le d_{u,x} - d_{u,a}$ 被 Dijkstra 处理过的边数控制.

QOJ 8673

分析

回到之前的最短路: 对询问 u, v, 假设在点 x 相遇, 得到答案的上界 $d_{u,x} + d_{x,v}$. 对一条可能更新答案的边 $a \to b$, 我们需要 $d_{u,a} + e_{a,b} + d_{b,v} \le d_{u,x} + d_{x,v}$. 即 $e_{a,b} \le d_{u,x} - d_{u,a} + d_{x,v} - d_{b,v} \le 2\max(d_{u,x} - d_{u,a}, d_{x,v} - d_{b,v})$.

对一个 u 侧的点 a, 只需要扫到权值 $\le 2(d_{u,x} - d_{u,a})$ 的边.

对一个 u 侧的点 a, 权值 $\le d_{u,x} - d_{u,a}$ 被 Dijkstra 处理过的边数控制.

问题: a 的出边边权和 $\le d_{u,x} - d_{u,a}$ 不独立.

QOJ 8673

对节点 a, 假设有 A 条出边边权 $\leq v$, B 条出边边权 $\leq 2v$. 我们将证明以高概率 $B \leq 2(k+1)A\log n$. 注意取出所有 B 条边后, 每条出边的边权 $\leq v$ 的概率都是独立的 $1/2$.

QOJ 8673

对节点 a, 假设有 A 条出边边权 $\le v$, B 条出边边权 $\le 2v$. 我们将证明以高概率 $B \le 2(k+1)A\log n$. 注意取出所有 B 条边后, 每条出边的边权 $\le v$ 的概率都是独立的 $1/2$.

定理 (Hoeffding's inequality)

假设 $X_1, \ldots, X_n$ 是独立随机变量且 $a_i \le X_i \le b_i$, 记 $S = \sum_{i=1}^{n} X_i$, 则有

$$\Pr(|S - \mathbb{E}(S)| \ge t) \le 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

QOJ 8673

对节点 a, 假设有 A 条出边边权 $\le v$, B 条出边边权 $\le 2v$. 我们将证明以高概率 $B \le 2(k+1)A\log n$. 注意取出所有 B 条边后, 每条出边的边权 $\le v$ 的概率都是独立的 $1/2$.

定理 (Hoeffding's inequality)

假设 $X_1, \ldots, X_n$ 是独立随机变量且 $a_i \le X_i \le b_i$, 记 $S = \sum_{i=1}^{n} X_i$, 则有

$$\Pr(|S - \mathbb{E}(S)| \ge t) \le 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

代入 Hoeffding 不等式, 在 $B \le 2(k+1)A\log n$ 时,

$$\Pr(|A - \mathbb{E}(A)| \ge k\log nA) \le 2\exp\left(-\frac{k^2\log^2 n}{2(k+1)\log n}\right) = 2n^{-k^2/(2k+1)}.$$

QOJ 8673

对一个点 a, 取 $k = O(1)$ 即可得到 $n^{-O(1)}$ 的正确率, 使用 Union Bound 合并错误概率后仍是 $n^{-O(1)}$.

QOJ 8673

对一个点 a, 取 $k = O(1)$ 即可得到 $n^{-O(1)}$ 的正确率, 使用 Union Bound 合并错误概率后仍是 $n^{-O(1)}$.

所以以高概率时间复杂度是 $O(q\sqrt{n}\log n + m\log m)$.

QOJ 3276

问题

给定长为 n 的序列 a_i, 保证 a_i 从 $[-1000, 1001]$ 中独立随机生成. 定义一个区间的的权值为 $f(\ell, r) = (\sum_{i=\ell}^{r} a_i)^2/(r-\ell+1)$. 有 q 次询问, 每次询问给定 L, R, 你需要输出 $\max_{L\le\ell\le r\le R} f(\ell, r)$.

$n \le 10^5, q \le 3 \times 10^5$.

QOJ 3276

问题

给定长为 n 的序列 a_i, 保证 a_i 从 $[-1000, 1001]$ 中独立随机生成. 定义一个区间的的权值为 $f(\ell, r) = (\sum_{i=\ell}^{r} a_i)^2/(r-\ell+1)$. 有 q 次询问, 每次询问给定 L, R, 你需要输出 $\max_{L\le\ell\le r\le R} f(\ell, r)$.
$n \le 10^5, q \le 3\times 10^5$.

Hint(Hoeffding's inequality)

假设 $X_1, \ldots, X_n$ 是独立随机变量且 $a_i \le X_i \le b_i$, 记 $S = \sum_{i=1}^{n} X_i$, 则有

$$\Pr(|S - \mathbb{E}(S)| \ge t) \le 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right).$$

QOJ 3276

问题

给定长为 n 的序列 a_i, 保证 a_i 从 $[-1000, 1001]$ 中独立随机生成. 定义一个区间的的权值为 $f(\ell, r) = (\sum_{i=\ell}^{r} a_i)^2/(r-\ell+1)$. 有 q 次询问, 每次询问给定 L, R, 你需要输出 $\max_{L \le \ell \le r \le R} f(\ell, r)$.

解

注意到对一个区间, 如果他内部有一个区间和的绝对值比他大, 这个区间将永远不会作为答案. 这说明对一个固定的左端点, 只需要保留前缀和的前缀最大值所在的位置.

QOJ 3276

问题

给定长为 n 的序列 a_i, 保证 a_i 从 $[-1000, 1001]$ 中独立随机生成. 定义一个区间的的权值为 $f(\ell, r) = (\sum_{i=\ell}^{r} a_i)^2/(r-\ell+1)$. 有 q 次询问, 每次询问给定 L, R, 你需要输出 $\max_{L\le\ell\le r\le R} f(\ell, r)$.

解

注意到对一个区间, 如果他内部有一个区间和的绝对值比他大, 这个区间将永远不会作为答案. 这说明对一个固定的左端点, 只需要保留前缀和的前缀最大值所在的位置.
同时, *Hoeffding* 不等式说明, 如果一个区间比较长, 他大概率不是答案.

QOJ 3276

前缀和的前缀最大值到底有多少个?

QOJ 3276

前缀和的前缀最大值到底有多少个?

定理

对一个长为 n 的序列 a_i, 且 a_i 在 $\{-1, 1\}$ 中独立均匀随机, a 的前缀和的前缀最大值期望个数为 $O(\sqrt{n})$.

QOJ 3276

前缀和的前缀最大值到底有多少个?

定理

对一个长为 n 的序列 a_i, 且 a_i 在 $\{-1, 1\}$ 中独立均匀随机, a 的前缀和的前缀最大值期望个数为 $O(\sqrt{n})$.

分析

考虑一个前缀 i 是前缀最大值的概率, 相当于从 i 开始倒着往前, 每个前缀和都 < 0.
简单的反射容斥说明, 这一事件的概率是 $\binom{i}{\lfloor i/2 \rfloor}/2^i$.

QOJ 3276

定理 (Stirling)

$$\lim_{n\to\infty} \frac{n!}{\sqrt{2\pi n}(n/e)^n} = 1.$$

QOJ 3276

定理 (Stirling)

$$\lim_{n\to\infty} \frac{n!}{\sqrt{2\pi n}(n/e)^n} = 1.$$

这说明一个位置是前缀最大值的概率大约为

$$\binom{2n}{n}/4^n \approx \frac{\sqrt{4\pi n}(2n/e)^{2n}}{2\pi n(n/e)^{2n}} \cdot \frac{1}{4^n} = O(1/\sqrt{n}).$$

QOJ 3276

定理 (Stirling)

$$\lim_{n\to\infty} \frac{n!}{\sqrt{2\pi n}(n/e)^n} = 1.$$

这说明一个位置是前缀最大值的概率大约为

$$\binom{2n}{n}/4^n \approx \frac{\sqrt{4\pi n}(2n/e)^{2n}}{2\pi n(n/e)^{2n}} \cdot \frac{1}{4^n} = O(1/\sqrt{n}).$$

在原序列中把 ≤ 0 的看作 -1, > 0 的看作 1. Hoeffding 不等式说明以高概率新序列的前缀最大值就是原序列的前缀最大值.

QOJ 3276

问题

给定长为 n 的序列 a_i, 保证 a_i 从 $[-1000, 1001]$ 中独立随机生成. 定义一个区间的的权值为 $f(\ell, r) = (\sum_{i=\ell}^{r} a_i)^2/(r-\ell+1)$. 有 q 次询问, 每次询问给定 L, R, 你需要输出 $\max_{L\le\ell\le r\le R} f(\ell, r)$.

解

使用单调栈取出每个左端点的前缀和的前缀最大值, 将可能成为答案的区间记录, 然后做二维偏序即可.