string

ztzshiwo

2017 年 9 月 16 日

写在前面

我这套 PPT 的内容都很简单 · · · · · ·

完全没有前面的同学的难度，所以基本上所有的题目我的 AC 代码都会传到 FTP 上，如果有听不懂的同学可以看课件结合代码理解 (虽然有些题网上的一些题解比我高到不知道哪里去了，我的代码可能还有调试信息没删，而且还没写注释 · · ·)。

请一眼就秒掉大概看出来做法的同学想想具体实现 (不然的话全部秒了我很尴尬的)。

Contents

Hash

Hash 其实是一种复杂且困难的题目类型。

Hash

Hash 其实是一种复杂且困难的题目类型。

其主要涉及的思想就是通过随机化等方法给一个不太容易快速判断相等的模型分配数值或其他相对容易判断相等的东西，并且这种东西在模型不同时几乎不可能相等。

Hash

Hash 其实是一种复杂且困难的题目类型。

其主要涉及的思想就是通过随机化等方法给一个不太容易快速判断相等的模型分配数值或其他相对容易判断相等的东西，并且这种东西在模型不同时几乎不可能相等。

比较经典的问题有字符串的同构，树的同构，图的同构等等。

Hash

Hash 其实是一种复杂且困难的题目类型。

其主要涉及的思想就是通过随机化等方法给一个不太容易快速判断相等的模型分配数值或其他相对容易判断相等的东西，并且这种东西在模型不同时几乎不可能相等。

比较经典的问题有字符串的同构，树的同构，图的同构等等。
其实后面三个本来我都不会的，但是我请教了某刘姓 $\cdots\cdots$
接下来我就大概讲一下树的重构与图的重构 (字符串判相同就不用讲了吧 $\cdots$)。

Hash 套路题 1[1]

给你 m 个无根树，树的点数为 n，请你把它们按同构关系分成若干个等价类。
$m \leq 50, n \leq 50$

既然这是一个套路题，我就不耽误时间了。

[1] BZOJ 4337

Hash 套路题 1[1]

给你 m 个无根树，树的点数为 n，请你把它们按同构关系分成若干个等价类。
$m \leq 50, n \leq 50$

既然这是一个套路题，我就不耽误时间了。

首先对于一个有根树，我们可以随便用一些比较强的 $Hash$.
我的 $Hash$ 是这样的:
对于一个节点 u，先求出它所有儿子节点的 $Hash$ 值，并从小到大排序，记作 $H_1, H_2, , H_D$。
那么 v 的 $Hash$ 值就可以计算为:
$Hash(u) = (((\cdots(((A * B\, xor\, H_1) mod\, Q) *$
$P\, xor\, H_2) mod\, Q \cdots)\, mod\, Q) * P\, xor\, H_D) * B\, mod\, Q$

[1] BZOJ 4337

Hash 套路题 1

用 Hash 的时候只要注意排序 Hash 值 (因为顺序无关), 最后一定要乘 B 就可以了。
具体实现的时候可以直接用 $unsigned\,long\,long$, 不用取模。

接下来考虑怎么做无根树。

Hash 套路题 1

用 Hash 的时候只要注意排序 Hash 值 (因为顺序无关), 最后一定要乘 B 就可以了。
具体实现的时候可以直接用 $unsigned\,long\,long$, 不用取模。

接下来考虑怎么做无根树。
我们可以发现不管树以哪个点为根, 他的中心和重心都是不变的。所以我们可以以中心或者重心为根算 $Hash$ 值, 判断相等即可。

Hash 套路题 1

用 Hash 的时候只要注意排序 Hash 值 (因为顺序无关), 最后一定要乘 B 就可以了。
具体实现的时候可以直接用 $unsigned\,long\,long$, 不用取模。

接下来考虑怎么做无根树。
我们可以发现不管树以哪个点为根, 他的中心和重心都是不变的。所以我们可以以中心或者重心为根算 $Hash$ 值, 判断相等即可。
注意中心和重心有可能有两个, 一种处理方法是给这两个点连到一个点上算。这种处理方法注意要与连成的树区分 (通过点数或者直径长度区分就可以了)。

Hash 套路题 2^2

如果一个无向图重标号后与另一个无向图完全一致 (即对于任意两点，他们之间的边在两个图中都存在或都不存在)，则称两个无向图同构。给定两个 n 个点 m 条边的无向图，判定两个无向图是否同构。

$n \leq 200, m \leq 4000$

这个题我也不浪费时间了。

[2] 51nod 1676

Hash 套路题 2[2]

如果一个无向图重标号后与另一个无向图完全一致 (即对于任意两点，他们之间的边在两个图中都存在或都不存在)，则称两个无向图同构。
给定两个 n 个点 m 条边的无向图，判定两个无向图是否同构。
$n \leq 200, m \leq 4000$
这个题我也不浪费时间了。
初始时设每个点为点权为 1，之后进行 n 次迭代。
每次迭代每个点的值更替为与其相邻的点和他本身上一次迭代后的权值计算出的 hash 值。
即:

$$Hash[u][j] = A * Hash[u][j-1] + B * \sum_{(u,v) \in E} Hash[v][j-1] + j$$

[2] 51nod 1676

Hash 套路题 2

这是题解:
如果两个无向图同构则 n 次迭代后, 所有权值排序后应该对应相等。
1. 如果是两个同构的无向图, 则最后算出来的值排序后必定相同。
2. 如果是两个不同构的无向图, 这张图上的权值相当于是基于这张图的形状而给出的随机值,
两个不同构的无向图上的权值相等的概率只有 1/p,p 为 hash 时的取模数, 当 p 取非常大时几乎可以认定两个图同构当且仅当 hash 值相同。
也可以通过多取几个取模数、增加迭代次数等增加正确率, 但此题只需取一个 p, 迭代 n 次已经完全足够了。
复杂度 O(T*nmlogn)

Hash 例题 1[3]

给出一个长度为 n 的由小写英文字母组成的字符串 S, 再给出 q 个询问, 要求回答 S 某个子串的最短循环节。
如果字符串 B 是字符串 A 的循环节, 那么 A 可以由 B 重复若干次得到。
数据范围:$n \leq 5 * 10^5, q \leq 2 * 10^6$

[3]BZOJ 2795 权限题

Hash 例题 1

这个问题可以转化为我们已经知道一个串的 Hash 值，问怎么快速求循环节长度

Hash 例题 1

这个问题可以转化为我们已经知道一个串的 Hash 值，问怎么快速求循环节长度

首先我们可以知道如果 $x|n$，并且 $S_{1\sim n-x} = S_{x+1\sim n}$，那么 x 一定是 S 的一个循环节。

Hash 例题 1

这个问题可以转化为我们已经知道一个串的 Hash 值，问怎么快速求循环节长度

首先我们可以知道如果 $x|n$，并且 $S_{1\sim n-x} = S_{x+1\sim n}$，那么 x 一定是 S 的一个循环节。

同时我们还发现如果 x 是 S 的循环节，那么 $kx(kx \mid n)$ 也一定是 S 的循环节。

Hash 例题 1

因此我们可以枚举 n 的质因数，假设当前的循环节长度为 m(初始时为 n)，当前枚举的质因数为 x_i，只要 $\frac{m}{x}$ 满足条件我们就一直让 m 除 x_i，然后继续用下一个质因数检验

Hash 例题 1

因此我们可以枚举 n 的质因数, 假设当前的循环节长度为 m(初始时为 n), 当前枚举的质因数为 x_i, 只要 $\frac{m}{x}$ 满足条件我们就一直让 m 除 x_i, 然后继续用下一个质因数检验

可以知道最多的尝试个数是 $log_2(n)$ 的.
因此复杂度就是 $O(log_2(n)q)$

Hash 例题 2^4

两个字符串相似定义为：

- 1. 两个字符串长度相等
- 2. 两个字符串对应位置上有且仅有至多一个位置所对应的字符不相同

给定一个字符串，每次询问两个子串在给定的规则下是否相似。
给定的规则指每次给出一些等价关系，如‘a’=’ b’，‘b’=’ c’ 等，注意这里的等价关系具有传递性，即若‘a’=’ b’，‘b’=’ c’，则‘a’=’ c’。

$|s| \leq 300000, T \leq 300000$

[4] 51nod 1753

Hash 例题 2

这个题应该很多人都看过了，我就讲快一点吧。

假设字符串的第 i 位字符为 x_i，那么 $S_{l\sim r}$ 的 Hash 值为 $\sum_{i=l}^{r} x_i * p^{r-i}$
其实就是关于 $x_{l\sim r}$ 的多项式，由于只有小写字母，所以是一个有 26 个未知数的多项式。
对于等价的字母，只要保证带入的值相同，那么就可以判断相同。

Hash 例题 2

这个题应该很多人都看过了，我就讲快一点吧。

假设字符串的第 i 位字符为 x_i，那么 $S_{l\sim r}$ 的 Hash 值为 $\sum_{i=l}^{r} x_i * p^{r-i}$
其实就是关于 $x_{l\sim r}$ 的多项式，由于只有小写字母，所以是一个有 26 个未知数的多项式。
对于等价的字母，只要保证带入的值相同，那么就可以判断相同。
对于只有一个位置不同的两个串，我们可以二分不同的位置在哪里，如果前面的串不同就缩小范围，否则就增大范围。然后再判断后面的串是否相等就可以了。
这里就可以在 $O(26nlogn)$ 的时间复杂度之内解决了。

这几个内容也是比较重要的，但是好像找不到什么难题

如果有同学想要训练这方面的题的话，之前 HYY 的课件里面有提到，做了那些题基本上就不用担心有什么问题了。

AC automaton

AC 自动机是一种非常常用的用于字符串匹配的结构。

AC automaton

AC 自动机是一种非常常用的用于字符串匹配的结构。

比较简单的题目是在 AC 自动机上 DP 或者利用 AC 自动机的节点进行矩乘。

稍微复杂点的题目主要运用了 AC 自动机的 fail 树，利用 dfs 序或者数据结构维护可以出一些更灵活的题.

AC automaton 例题 1[5]

某人读论文，一篇论文是由许多单词组成。但他发现一个单词会在论文中出现很多次，现在想知道每个单词分别在论文中出现多少次。
每个单词由小写字母组成，$n \leq 200$，单词总长度 $\leq 10^6$

[5] BZOJ 3172

AC automaton 例题 1

这题还是非常简单的，相信应该马上会有大佬秒掉。
可以发现一个单词在论文中出现的次数就是它的结束节点在 fail 树的子树中的节点被经过的次数。

AC automaton 例题 1

这题还是非常简单的，相信应该马上会有大佬秒掉。
可以发现一个单词在论文中出现的次数就是它的结束节点在 fail 树的子树中的节点被经过的次数。
那么我们只要反过来首先给每个经过的点 $++$，然后弄个子树统计，这时候只要输出结束节点的权值就可以了。
时间复杂度为 $O(26 * 10^6)$

AC automaton 例题 2[6]

给定 n 个字符串 S_i, 每次询问给出 l, r, k, 求 $\sum_{i=l}^{r} occur(S_i, S_k)$.

其中 $occur(P, T)$ 代表串 P 在串 T 中的出现次数。

$n, q, \sum |S_i| \leq 100000$

[6] CF 587F

AC automaton 例题 2

提示: 这个题的 $\sum |S_i| \leq 100000$,

因此 $|S_i| \geq \sqrt{n}$ 的不超过 $\sqrt{n}$ 个。

可以对于 $|S_i| \leq \sqrt{n}$ 和 $|S_i| > \sqrt{n}$ 的情况分开考虑.

AC automaton 例题 2

对于 $S_k \geq \sqrt{n}$ 的询问

之前说过 A 串在 B 串出现次数等于 A 串的结束节点在 fail 树的子树中出现多少个 B 串的节点。

AC automaton 例题 2

对于 $S_k \geq \sqrt{n}$ 的询问

之前说过 A 串在 B 串出现次数等于 A 串的结束节点在 fail 树的子树中出现多少个 B 串的节点。

那么我们可以对于 S_k 串在 fail 树上 $++$，然后搞一次子树统计，这样每个串在这个串中的出现次数就可以 $O(1)$ 统计，然后搞一个前缀和就可以 $O(n)$ 处理所有这种类型的询问了。

AC automaton 例题 2

对于 $S_k < \sqrt{n}$ 的询问

考虑每一个询问可以变成差分的形式，然后询问就转为前 i 个字符串在 S_k 中的出现次数。

AC automaton 例题 2

对于 $S_k < \sqrt{n}$ 的询问

考虑每一个询问可以变成差分的形式，然后询问就转为前 i 个字符串在 S_k 中的出现次数。

接下来我们对询问进行排序，然后考虑反过来统计答案，添加一个串就相当于给它结束节点的子树 $++$。统计答案就相当于对于这个串的每个节点统计被覆盖的次数。

AC automaton 例题 2

对于 $S_k < \sqrt{n}$ 的询问

考虑每一个询问可以变成差分的形式，然后询问就转为前 i 个字符串在 S_k 中的出现次数。

接下来我们对询问进行排序，然后考虑反过来统计答案，添加一个串就相当于给它结束节点的子树 $++$。统计答案就相当于对于这个串的每个节点统计被覆盖的次数。
需要一个支持 $O(n)$ 次区间修改，$O(n\sqrt{n})$ 次单点查询的结构。
考虑分块，然后总的复杂度就是 $O(n\sqrt{n})$

Suffix Related

后缀数组和后缀自动机是处理单个串或多个串之间比较复杂的匹配问题时常用的数据结构。

Suffix Related

后缀数组和后缀自动机是处理单个串或多个串之间比较复杂的匹配问题时常用的数据结构。

虽然号称后缀数组的所有问题都可以用后缀自动机解决，但是我觉得有些问题在后缀自动机上很难解决。后缀数组套各种数据结构在 OI 界已经十分常见，而后缀自动机的各种应用还没有普及。

Suffix Related

后缀数组和后缀自动机是处理单个串或多个串之间比较复杂的匹配问题时常用的数据结构。

虽然号称后缀数组的所有问题都可以用后缀自动机解决，但是我觉得有些问题在后缀自动机上很难解决。后缀数组套各种数据结构在 OI 界已经十分常见，而后缀自动机的各种应用还没有普及。

我主要讲一些后缀数组的工业套路题和后缀自动机的常见应用。

后缀模板题 1[7]

给出一个串，询问这个串不同的字串个数。

- $|S| \leq 100000$ 字符集 ≤ 26
- $|S| \leq 100000$ 字符集 $\leq 10^9$

[7] SPOJ DISUBSTR

后缀模板题 1

这是真的模板题 · · ·

我就不浪费时间了。用后缀数组和后缀自动机都可以做。

后缀模板题 1

这是真的模板题 $\cdots$

我就不浪费时间了。用后缀数组和后缀自动机都可以做。

- 后缀数组只要排完序求一个 $height$, 总长度减去 $\sum height$ 就是答案。
- 后缀自动机只要插入完以后每个节点的 $Right - Left$(实际上就是这个节点的 $Right$-上个节点的 $Right$) 就是答案。

后缀例题 1[8]

给出两个串 A, B, 分别求 $1 \sim n$ 中的每一个长度 x 有多少个二元组 (i, j) 满足

$$i + x - 1 \leq |A|, j + x - 1 \leq |B|, A_{i \sim i+x-1} = B_{j \sim j+x-1}$$

$|A|, |B| \leq 200000$

- 后缀数组解法？
- 后缀自动机解法？

[8] CC TANGLED

后缀例题 1

首先重点讲一下后缀数组的解法。
首先将 A, B 串拼在一起做后缀数组。
可以发现对于长度为 i 时的答案，就相当于将后缀数组分成很多块，每一块里面出现在 A 串中的次数 * 出现在 B 串中的次数。

后缀例题 1

首先重点讲一下后缀数组的解法。
首先将 A, B 串拼在一起做后缀数组。
可以发现对于长度为 i 时的答案，就相当于将后缀数组分成很多块，每一块里面出现在 A 串中的次数 * 出现在 B 串中的次数。
这样我们维护并查集，按照 height 从高到低合并，维护当前集合在 A 串中出现的个数和在 B 串中出现的次数。
然后前缀和一下就可以求出答案了。

后缀例题 1

首先重点讲一下后缀数组的解法。
首先将 A, B 串拼在一起做后缀数组。
可以发现对于长度为 i 时的答案，就相当于将后缀数组分成很多块，每一块里面出现在 A 串中的次数 * 出现在 B 串中的次数。
这样我们维护并查集，按照 height 从高到低合并，维护当前集合在 A 串中出现的个数和在 B 串中出现的次数。
然后前缀和一下就可以求出答案了。

后缀自动机的解法是考虑每一个节点，可以知道对当前节点的 $left \sim right$ 长度都有一个 A 出现个数 *B 出现个数的贡献，弄个 BIT 维护就可以了。

后缀例题 2^9

给出一个字符串 T，要求你回答下面两种询问：

- $Select\,k1\,k2$：询问在所有 T 的不同子串中，字典序在第 $k1$ 位的字符串，如果有相同的，输出它第 $k2$ 次出现的位置。
- $Rank\,l\,r$：询问子串 $T_{l\sim r}$ 在所有不同子串中排在第几位，如果有相同的，输出它在所有起始位置中排名第几。

$|T| \leq 10^5, Q \leq 10^5$

[9] CC QSTRING

后缀例题 2

首先考虑第一种询问：

我们可以发现在后缀数组找不同子串个数的时候，实际上不同子串已经按照顺序排好了。

后缀例题 2

首先考虑第一种询问：

我们可以发现在后缀数组找不同子串个数的时候，实际上不同子串已经按照顺序排好了。
首先找到在后缀数组中第 k1 个子串第一次出现的位置，然后二分找到后面 height>= 这个字串长度的最远位置。
这些位置在原串中的位置都出现了这个子串，然后搞个主席树查询第 k2 个就可以了。

后缀例题 2

然后考虑第二种询问：

其实跟第一个询问差不多，二分出跟当前位置 height$>=$ 串长度的最左边的位置和最右边的位置。
第一个排名可以直接计算，第二个排名同样是在主席树中查询。

后缀例题 2

然后考虑第二种询问：

其实跟第一个询问差不多，二分出跟当前位置 height>= 串长度的最左边的位置和最右边的位置。
第一个排名可以直接计算，第二个排名同样是在主席树中查询。

其实这道题我还没过，一直 T···，因为数据特别卡，好像只有一个人 A 了。我复制这个人的代码上去还是 T 了 ···
话说这题还是上古神牛 $xiaodao$ 出的，出到比赛里面的时候好像还出了事故，$O(n^2)$ 跑得比 $O(nlogn)$ 还快 ···

后缀例题 3^{10}

给出一个长度为 n 的串 S, 给出 m 个询问。
每次询问给出四个参数 (a, b, c, d).
询问 $S_{a\sim b}$ 的所有子串与串 $S_{c\sim d}$ 的最长前缀长度。

$n, m \leq 10^5$

[10] BZOJ 4556

后缀例题 3

首先假设答案求的是 $min(d-c+1, max(lcp(S_{i\sim n}, S_{c\sim n}(a \le i \le b))))$，也就是不考虑 b 带来的长度限制。
考虑所有 i 在后缀数组中的位置，可以知道答案一定在离 c 出现的最近的两个位置中。
那么同样可以用主席树查询前驱与后继，就可以得到答案了。

后缀例题 3

首先假设答案求的是 $min(d-c+1, max(lcp(S_{i\sim n}, S_{c\sim n}(a \le i \le b))))$，也就是不考虑 b 带来的长度限制。
考虑所有 i 在后缀数组中的位置，可以知道答案一定在离 c 出现的最近的两个位置中。
那么同样可以用主席树查询前驱与后继，就可以得到答案了。

接下来考虑如何计算 b 的影响，可以像 dyy 之前讲的一道题一样，二分答案，然后判断 $a \sim b-x+1$ 的位置有没有 $\ge$x 的，统计答案就可以了。

后缀例题 4 [11]

一个长度为 n 的字符串 S, 令 T_i 表示它从第 i 个字符开始的后缀。求:

$$\sum_{1 \le i < j \le n} len(T_i) + len(T_j) - 2 * lcp(T_i, T_j)$$

$n \le 500000$, 字符串由小写字母组成。

- 后缀数组解法？
- 后缀自动机解法？

[11] BZOJ 3238

后缀例题 4

其实这题的后缀数组做法跟之前一道题基本上一样，我们可以请 cat 上来讲一下。

后缀例题 4

其实这题的后缀数组做法跟之前一道题基本上一样，我们可以请 cat 上来讲一下。

接下来讲后缀自动机解法:
可以知道答案难的就是任意两个后缀的 lcp，考虑后缀自动机上每一个点的贡献就是

$$\sum_{v_1,v_2 \in Son(u), v_1 != v_2} \text{v1 包含的后缀个数} * \text{v2 包含的后缀个数} * Right(u)$$

后缀例题 4

其实这题的后缀数组做法跟之前一道题基本上一样，我们可以请 cat 上来讲一下。

接下来讲后缀自动机解法:
可以知道答案难的就是任意两个后缀的 lcp，考虑后缀自动机上每一个点的贡献就是

$$\sum_{v_1,v_2 \in Son(u), v_1 != v_2} \text{v1 包含的后缀个数} * \text{v2 包含的后缀个数} * Right(u)$$

我们可以弄出拓扑序，然后统计的时候逆序把点就到自己的父亲上，然后给答案加上当前点的后缀个数 * 当前父亲点已经统计过的后缀个数 $*Right($父亲节点$)$。

这道题就可以 $O(n)$ 解决了。

后缀例题 5[12]

给出一个长度为 n 的字符串 S，以及 m 个字符串 T。
给出 Q 个询问，每个询问 (l, r, x, y) 表示:
$S_{l\sim r}$ 在第 $x \sim y$ 个字符串中出现次数最多的是在哪个串，输出最大次数和串的最小编号。

$n, Q \leq 5 * 10^5$ $m, \sum T \leq 5 * 10^4$

- 后缀数组解法？
- 后缀自动机解法？

[12] CF 666E

后缀例题 5

后缀自动机的解法还是挺简单的:
首先把所有字符串加分隔符插入 SAM

后缀例题 5

后缀自动机的解法还是挺简单的:
首先把所有字符串加分隔符插入 SAM
接下来考虑如何在 SAM 上定位子串 $S_{l\sim r}$，通过 SAM 的性质可以发现子串所在的节点一定在 r 所在点的 link 链接中，只要倍增一下找到最前面的一个 $max \geq r-l+1$ 的点，这个点一定包含子串 $S_{l\sim r}$。

后缀例题 5

后缀自动机的解法还是挺简单的:
首先把所有字符串加分隔符插入 SAM
接下来考虑如何在 SAM 上定位子串 $S_{l\sim r}$，通过 SAM 的性质可以发现子串所在的节点一定在 r 所在点的 link 链接中，只要倍增一下找到最前面的一个 $max \geq r-l+1$ 的点，这个点一定包含子串 $S_{l\sim r}$。

接下来考虑怎么计算答案，可以知道包含串 $S_{l\sim r}$ 的点全都在当前点的子树内，那么我们只要离线用线段树合并就可以统计答案了。

后缀例题 5

其实后缀数组的解法我不会 · · ·
但是我在网上找到了一份代码，虽然我暂时没看懂 · · ·

后缀例题 5

其实后缀数组的解法我不会 · · ·
但是我在网上找到了一份代码，虽然我暂时没看懂 · · ·

给一下链接吧，有感兴趣的同学的话可以看一看
http://codeforces.com/contest/666/submission/29112399

后缀例题

本来还准备讲一道题的：

$BZOJ3926$ $ZJOI2015$ 诸神眷顾的幻想乡

后缀例题

本来还准备讲一道题的：

$BZOJ3926$ $ZJOI2015$ 诸神眷顾的幻想乡

这是一道广义 SAM 的题 (其实就是给 trie 构建 SAM)

感兴趣的同学可以去做一做

谢谢大家