基础优化技巧1

wrpwrp
洛谷省选计划

目录

三分

三分

三分用来解决求单峰函数的极值点的问题。

单峰函数： 在所考虑的区间中只有一个极值点的函数。

洛谷3382 【模板】三分

给出一个 N 次函数， 保证在范围 [l, r] 内存在一点 x， 使得 [l, x] 上单调增， [x, r] 上单调减。试求出 x 的值。

输出为一行，包含一个实数，即为 x 的值。若你的答案与标准答案的相对或绝对误差不超过 10^{-5} 则算正确。$|l|, |r| \leq 10$ 。

洛谷3382 【模板】三分

令 lmid, rmid 是区间任意的两个分段点， 把区间分成三段。
丢弃极值点一定不在的那一段区间， 使得区间长度缩小。

以这题上凸的函数为例子， 当 f (lmid) > f (rmid) 的时候， 丢弃 (rmid, R) 这一段， 令 R = rmid 继续循环， 反之则丢弃 (L, lmid) 这一段。

上凸函数 : 开口向下的单峰函数

洛谷3382 【模板】三分

```
const double eps = 1e-8;

int main() {
    cin >> n >> L >> R;
    for (int i = 0; i <= n; i ++) cin >> f[i];
    while (fabs (L - R) >= eps) {
        double len = R - L; len /= 3;
        double lmid = L + len, rmid = R - len;
        if (calc (lmid) < calc (rmid)) L = lmid;
        else R = rmid;
    }
    cout << fixed << setprecision (6) << L << endl;
    return 0;
}
```

洛谷3382 【模板】三分

可以发现其实只会丢掉最两边的段， 中间这一段不会丢， 可以让两边的段尽量长。

那么我们令 lmid = mid – delta, rmid = mid + delta, 这样每次可以让区间长度缩小一半, 其中 delta 是一个很小的量。

洛谷3382 【模板】三分

```
const double eps = 1e-8;

int main() {
    cin >> n >> L >> R;
    for (int i = 0; i <= n; i ++) cin >> f[i];
    while (fabs (L - R) >= eps) {
        double len = 1e-5;
        double mid = (L + R) / 2;
        double lmid = mid - len, rmid = mid + len;
        if (calc (lmid) < calc (rmid)) L = mid;
        else R = mid;
    }
    cout << fixed << setprecision (6) << L << endl;
    return 0;
}
```

洛谷3382 【模板】三分

当然也是可以二分斜率的。

二分斜率的时候为了避免求导， 可以计算 f(mid), f (mid + delta) 的值， 其中 delta 是一个很小的数来判断怎么取舍区间。

写成代码和上张ppt的三分其实是一样的。

没有啥难度， 一般就是观察到函数有什么单调性就可以用用。

整体二分 / 01分数规划

01分数规划

形如：

有 n 个物品，每个物品有两个权值 a, b，然后让你选出一些物品使得两个权值和的比值最大或者最小。也就是令 w_i 表示选 $(w_i = 1)$ 或者不选 $(w_i = 0)$，最大或者最小化下列值。

$$\frac{\sum w_i \times a_i}{\sum w_i \times b_i}$$

除此之外，一般还会对选择的件数或者是选择的物品给出一些限制。

01分数规划

一般来说我们使用二分求解。二分一个值 t 以后， 得到：

$$\frac{\sum w_i \times a_i}{\sum w_i \times b_i} \geq t$$

意思是二分一个值， 然后判定能不能大于这个值， 然后移项:

$$\sum w_i \times (a_i - t \times b_i) \geq 0$$

[HNOI2009] 最小圈

给一张 n 个点 m 条边的有向带权图，在图中找一个环，使得环 上边权之和除以节点个数最小，求这个最小平均值。

$n \le 3000, m \le 10000$, 边权 $\le 10^7$

[HNOI2009] 最小圈

观察到答案是一个“比率” 的形式，联想到分数规划问题。

考虑把 边 对应到模型中的 a， 点 对应到模型中的 b。

二分一个答案 t 后， 边权 v 重新赋值为 v – t，使用 SPFA 算法找负环即可。

复杂度 $O(\log_2 v \times n \times m)$

Ps : 这题因为某些历史原因可能 SPFA 判环过不去， 知道自己是对的就好（。

整体二分

有多个需要二分的对象的时候， 有时候可以放在一起来二分。

P3527 [POI2011] MET-Meteors

有 n 个成员国。现在它发现了一颗新的星球，这颗星球的轨道被分为 m 份（第 m 份和第 1 份相邻），第 i 份上有第 a_i 个国家的太空站。

这个星球经常会下陨石雨。BIU 已经预测了接下来 k 场陨石雨的情况。

BIU 的第 i 个成员国希望能够收集 p_i 单位的陨石样本。你的任务是判断对于每个国家，它需要在第几次陨石雨之后，才能收集足够的陨石。

输入格式

第一行是两个数 n, m。

第二行有 m 个数，第 i 个数 o_i 表示第 i 段轨道上有第 o_i 个国家的太空站。

第三行有 n 个数，第 i 个数 p_i 表示第 i 个国家希望收集的陨石数量。

第四行有一个数 k，表示 BIU 预测了接下来的 k 场陨石雨。接下来 k 行，每行有三个数 l_i, r_i, a_i，表示第 k 场陨石雨的发生地点在从 l_i 顺时针到 r_i 的区间中（如果 $l_i \le r_i$，则是 $l_i, l_i+1 \cdots, r_i$，否则就是 $l_i, l_i+1, \cdots m-1, m, 1, 2, \cdots r_i$），向区间中的每个太空站提供 a_i 单位的陨石样本。

输出格式

输出 n 行。第 i 行的数 w_i 表示第 i 个国家在第 w_i 波陨石雨之后能够收集到足够的陨石样本。如果到第 k 波结束后仍然收集不到，输出 `NIE`。

数据范围

$1 \le n, m, k \le 3 \cdot 10^5$;

$1 \le p_i, a_i \le 10^9$;

P3527 [POI2011] MET-Meteors

考虑只有一个国家?

可以考虑二分一下时间， 然后试着加入陨石雨。

看起来来有点傻， 但是进一步拓展到整体一起二分呢?

P3527 [POI2011] MET-Meteors

考虑对所有国家同时进行二分， 使用 vector 存当前要处理的所有国家以及还没有处理完的陨石雨， 定义一个函数 solve (country, range, L, R) 表示当前要处理 v 中的国家， 可能的答案区间是 [L, R], 二分出一个 mid = (L + R) / 2 以后， 递归处理 solve (countryl, rangel, L, mid) 和 solve (countryr, ranger, mid + 1, R)。 其中 countryl 是答案小于等于 mid 的国家， countryr 是答案大于 mid 的国家。

复杂度为 $O(n \log_2^2 n)$

P3527 [POI2011] MET-Meteors

```cpp
void solve(std :: vector<int> &country, auto & range, int L, int R) {
    if (L == R) {
        for (int x : country) ans[x] = L; // 边界
        return ;
    }
    if (country.size() == 0) return ;
    int mid = (L + R) >> 1;

    std :: vector<std :: tuple<int, int, int, int> > lrange, rrange;

    for (auto &[l, r, a, i] : range) {
        if (i > mid) {
            rrange.push_back ( {l, r, a, i} );
            continue;
        }
        bit.add(l, r, a);
        lrange.push_back ( {l, r, a, i} );
    } // 处理 mid 以前的区间

    std :: vector<int> countryl, countryr;
    for (auto &x : country) {
        i64 sum = 0;
        for (int y : lnk[x]) sum += bit.qry(y), sum = std :: min (sum, lim);
        if (sum >= p[x]) countryl.push_back (x);
        else countryr.push_back (x), p[x] -= sum;
    } // 判断每个国家收集是否足够

    for (auto &[l, r, a, i] : lrange) {
        bit.set (l);
        bit.set (r + 1);
    } // 清空

    solve(countryl, lrange, L, mid);
    solve(countryr, rrange, mid + 1, R);
}
```

分治

分治

主要思想大概是把大问题分成小问题再合并起来。

所以一般要求问题有可分性， 同时有可合并性。

种类， 功能， 应用场景都非常多，本次讲课做简单介绍， 后面会有更详细的阐述。

让我们从最经典而简单的开始。

P3810 【模板】三维偏序

题目描述

有 n 个元素，第 i 个元素有 a_i, b_i, c_i 三个属性，设 $f(i)$ 表示满足 $a_j \leq a_i$ 且 $b_j \leq b_i$ 且 $c_j \leq c_i$ 且 $j \neq i$ 的 j 的数量。

对于 $d \in [0, n)$，求 $f(i) = d$ 的数量。

说明/提示

$1 \leq n \leq 10^5$，$1 \leq a_i, b_i, c_i \leq k \leq 2 \times 10^5$。

P3810 【模板】三维偏序

排序解决一个维度， 分治解决一个维度， 剩下的维度使用树状数组解决。

P3810 【模板】三维偏序

按照 a 排序，然后通过不断取中点分治理，然后把点分成三类：左边的，右边的，分别考虑其贡献。

每次考虑计算跨越中点的贡献，分别对左边右边的点排序，利用树状数组进行累加。

P3810 【模板】三维偏序

这是经典的高维偏序问题， 除分治外， 还可使用KDT，树套树，bitset， 分块等方法来处理。

一般来说高维的用 bitset， 低维度的直接分治或者树套树即可。

bitset 求解高维偏序 - -Wallace- - 博客园 (cnblogs.com)

CDQ 分治

其实就是我们前面提到的三维偏序的做法。鉴于这是属于数据结构的内容， 不过多展开。

感兴趣的同学可以看看链接的题练习试试（

P4655 [CEOI2017] Building Bridges - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)

[ZJOI2016] 旅行者

小 Y 来到了一个新的城市旅行。她发现了这个城市的布局是网格状的，也就是有 n 条从东到西的道路和 m 条从南到北的道路，这些道路两两相交形成 $n \times m$ 个路口 (i,j)，$(1 \leq i \leq n, 1 \leq j \leq m)$

她发现不同的道路路况不同，所以通过不同的路口需要不同的时间。通过调查发现，从路口 (i,j) 到路口 $(i,j+1)$ 需要时间 $r(i,j)$ ，从路口 (i,j) 到路口 $(i+1,j)$ 需要时间 $c(i,j)$ 。注意这里的道路是双向的。小 Y 有 q 个询问，她想知道从路口 $(x1,y1)$ 到路口 $(x2,y2)$ 最少需要花多少时间。

说明/提示

数据规模与约定

- $n \times m \leq 2 \times 10^4$。
- $q \leq 10^5$。
- $1 \leq r(i,j), c(i,j) \leq 10^4$。

[ZJOI2016] 旅行者

观察到 $nm \leq 2 \times 10^5$，那么 n, m 中较小的一个不会超过500， 可考虑枚举小边分治。

具体的， 考虑每次从中间切开较长的一边， 处理这条中线上所有点到其余点的最短路， 每次处理所有跨越这条线的询问。

令 A = 2e5, 复杂度不超过 A sqrt(A) log(A) log (A)

CF1156E Special Segments of Permutation

题意翻译

给定一个长度为 n 的排列 p，求有多少区间 $[l,r]$ 满足，$p_l+p_r=\max\limits_{i=l}^{r}\{p_i\}$。

The first line contains one integer n ($3 \le n \le 2 \cdot 10^5$).

The second line contains n integers p_1 , p_2 , ..., p_n ($1 \le p_i \le n$). All these integers are pairwise distinct.

CF1156E Special Segments of Permutation

枚举最大值的位置进行分治。

统计所有跨过最大值的位置，使得复杂度只和短的一边相关，复杂度 $O(n\ \log_2 n)$ 。

一个类似思想的好题：

[P5979 [PA2014] Druzyny - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)]

倍增

倍增

一般是处理一个 f[i][j] 表示 [i, i + 2^k – 1] 这个区间或者从某个点开始跳 2^k 步的信息， 然后通过合并两个小区间的信息得到大区间的信息。

LCA, ST 表都是倍增的经典应用。

洛谷

P4155 [SCOI2015] 国旗计划

题目描述

复制Markdown 展开

A 国正在开展一项伟大的计划 —— 国旗计划。这项计划的内容是边防战士手举国旗环绕边境线奔袭一圈。这项计划需要多名边防战士以接力的形式共同完成，为此，国土安全局已经挑选了 N 名优秀的边防战上作为这项计划的候选人。

A 国幅员辽阔，边境线上设有 M 个边防站，顺时针编号 1 至 M。每名边防战士常驻两个边防站，并且善于在这两个边防站之间长途奔袭，我们称这两个边防站之间的路程是这个边防战士的奔袭区间。N 名边防战士都是精心挑选的，身体素质极佳，所以每名边防战士的奔袭区间都不会被其他边防战士的奔袭区间所包含。

现在，国十安全局局长希望知道，至少需要多少名边防战士，才能使得他们的奔袭区间覆盖全部的边境线，从而顺利地完成国旗计划。不仅如此，安全局局长还希望知道更详细的信息：对于每一名边防战士，在他必须参加国旗计划的前提下，至少需要多少名边防战士才能覆盖全部边境线，从而顺利地完成国旗计划。

说明/提示

$N \leqslant 2 \times 10^5, M < 10^9, 1 \leqslant C_i, D_i \leqslant M$。

P4155 [SCOI2015] 国旗计划

因为不存在一个区间完全包含另外一个区间， 那么把所有区间按照左端点排序， 右端点必然单调增加， 我们总是可以找到以一个战士最远可以跳到的那个战士所在的位置。

我们定义跳到最远那个战士为跳一步。 F[x][0] 表示 x 跳一步可以到的地方， 倍增即可。

哈希

哈希

哈希用于把某个对象通过对其特征的“描述”映射为一个值来进行代表，用于实现判重/判相等。

比如字符串哈希，树哈希等等。

还有一些利用哈希的一些小技巧比较实用。

字符串哈希

把字符串看成一个B (B > 26) 进制的数， 计算这个数对某个质数取模的结果作为哈希值。我们随机选取这个 B 和质数 P。一个结论是两个随机的不同字符串按照这样生成的哈希值相同的概率大约是 $\frac{1}{P}$， 所以比较次数很大的时候， 我们需要使用两个模数计算两个哈希值， 共同作为哈希值使用， 这样相同的概率得到减少。

对所有子串计算哈希值

```
pw[0] = 1;
for (int i = 1; i <= (int) s.size(); i ++) pw[i] = 1ll * pw[i - 1] * bs % P;
f[0] = 0;
for (int i = 1; i <= (int) s.size(); i ++) f[i] = (1ll * f[i - 1] * bs + s[i - 1]) % P;
```

令字符串 s 的哈希值是 $\sum s_i \times B^{n-i-1}$ ，那么 $f_r - f_{l-1} \times B^{r-l+1}$ 就是子串 [l, r] 的哈希值。

哈希

哈希的要点： 设计合适的哈希函数， 使得你想要保留的特征杯保留， 不希望保留的特征不被保留。

[ABC250E] Prefix Equality

题意翻译

给定两个长为 N 的数列 A, B 与 Q 次询问，每次询问给出 x_i, y_i，求出 A 的前 x_i 项去重后是否与 B 的前 y_i 项去重后相同。

说明/提示

制約

- $1 \le N, Q \le 2 \times 10^5$
- $1 \le a_i, b_i \le 10^9$
- $1 \le x_i, y_i \le N$
- 入力は全て整数

[ABC250E] Prefix Equality

考虑构造一个和顺序无关的哈希函数。

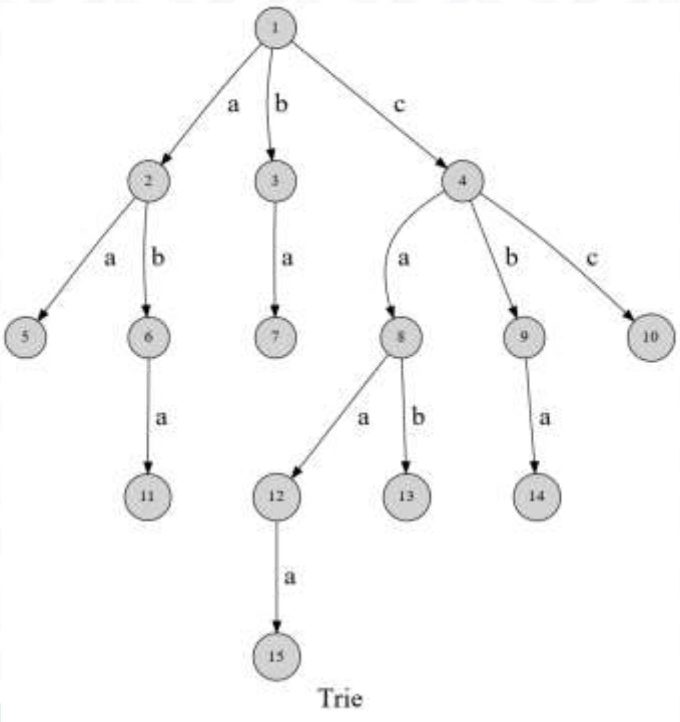
对每个数随机一个权值， 然后从前往后扫描处理一个 F[i] 表示这个序列前 i 个数的哈希值， 如果第 i 个数没出现过， $F[i] = F[i-1]\ xor\ h[a[i]]$ 。如果出现过即不变。

直接比较两个序列前缀的哈希值是否相等即可。

字典树

字典树（trie）

大概就是一个长得像字典的树。



字典树（trie）

都叫字典树了说明肯定可以当字典用。支持查询字符串是否出现和插入字符串。

另外， trie 可以反映两个字符串之间的前后缀关系。 如一个串的结束节点是另外一个串结束节点的祖先， 说明前者是后者的前缀。前后缀的判断可以通过这个性质转化成树上的问题。

P3065 [USACO12DEC] First! G

给定 $n\ (1 \le n \le 3 \times 10^4)$ 个总长不超过 $m\ (1 \le m \le 3 \times 10^5)$ 的互不相同的字符串，现在你可以任意指定字符之间的大小关系。问有多少个串可能成为字典序最小的串，并输出这些串。

P3065 [USACO12DEC] First! G

考虑到字典序比大小只和第一个不相同的字符有关，联想到可以在 Trie 树上做这个事情。

分情况讨论， 对26个字母连边表示如果要当前串字典序最小， 这些字母应该满足什么要求即可。

P3065 [USACO12DEC] First! G

考虑拿着这个串在Trie上走， 那么走到当前位置以后， 因为要求当前串最小， 所以要对同一层的其余子树里面有串的字符连边， 表示当前字母小， 这样可以得到一个有向图， 判断这个有向图有没有环就可以判断有没有矛盾。

这样我们考虑了所有没有前后缀关系的串， 对于有前后缀关系的串， 如果一个串的终止节点的祖先中存在别的串终止节点， 这个串一定不是最小串。

P4551 最长异或路径

给定一棵 n 个点的带权树，结点下标从 1 开始到 n。寻找树中找两个结点，求最长的异或路径。

异或路径指的是指两个结点之间唯一路径上的所有边权的异或。

数据范围

$1 \leq n \leq 100000; 0 < u, v \leq n; 0 \leq w < 2^{31}$。

P4551 最长异或路径

dfs一遍对每个点求出到根的异或和， 问题转化为求两个点使得这两个点异或和最大。

异或和最大先把所有数插入 Trie， 然后枚举一个数， 尽量走和当前边上数字相反的边进行贪心就好了。

CF888G Xor-MST

题意翻译

- 给定 n 个结点的无向完全图。每个点有一个点权为 a_i。连接 i 号结点和 j 号结点的边的边权为 $a_i \oplus a_j$。
- 求这个图的 MST 的权值。
- $1 \leq n \leq 2 \times 10^5$，$0 \leq a_i < 2^{30}$。

CF888G Xor-MST

异或最小生成树是 Trie 的经典习题之一。

考虑到一个求最小生成树的方法， 是认为开始有 n 个连通块， 每次寻找每个连通块连出去的最小边， 然后合并这条边两个点所在联通块， 把这条边加入最小生成树。

因为每次连通块个数会减半， 最多只会做 $\log_2 n$ 轮， 所以这也是一个 $m\log_2 n$复杂度的算法。

这个算法也叫 Borůvka's algorithm

CF888G Xor-MST

两个数异或最小， 等价于二进制最高位的 lcp 尽量长， 那么异或值就会小。

Lcp：最长公共前缀。

那么如果我们建立一个 Trie, 然后把数插入进去， 相当于我们要在尽量靠下的位置合并两个点， 这样就达到了B算法要求找到连通块向外连出的最小边的要求。

CF888G Xor-MST

那么 DFS Trie 树， 在每个点处， 如果左子树和右子树内都有点，枚举数较少的一边的数， 放到较大的一边查询异或最小值即可得到联通左右子树的边权。

因为每次枚举的是较少的一边， 复杂度同启发式合并。

复杂度 $O(n\log_2 n\log_2 v)$

KMP

KMP

KMP 应该算是最经典的字符串算法之一。

用于解决字符串匹配问题， 可以求出自身每个前缀的最长 border。

border： 既是前缀又是后缀的非本身的子串。

P5829【模板】失配树

题目描述

复制Markdown 展开

给定一个字符串 s，定义它的 k **前缀** pre_k 为字符串 $s_{1\dots k}$，k **后缀** suf_k 为字符串 $s_{|s|-k+1\dots|s|}$，其中 $1 \le k \le |s|$。

定义 $\textbf{Border}(s)$ 为**对于** $i \in [1, |s|)$，**满足** $pre_i = suf_i$ 的字符串 pre_i 的集合。$\textbf{Border}(s)$ 中的每个元素都称之为字符串 s 的 border。

有 m 组询问，每组询问给定 p, q，求 s 的 $\boldsymbol{p}$ **前缀** 和 $\boldsymbol{q}$ **前缀** 的 **最长公共** border 的长度。

对于 16% 的数据，s 中的字符全部相等。

对于 100% 的数据，$1 \le p, q \le |s| \le 10^6$，$1 \le m \le 10^5$，$s_i \in [\texttt{a}, \texttt{z}]$。

P5829【模板】失配树

先给出一个 border 的性质。

S[1…i] 的所有 border 由 next[S[1…i]], next[next[S[1…i]]], next[next[next[S[1…i]]]], … 等给出。

其中 next[i] 是 i 这个位置的 最长border 数组。

考虑 next[i] 是 i 最长的 border，　那么 i 剩下的border 一定比 next[i] 短，　且一定是 next[i] 的border。

P5829【模板】失配树

那么把 i -> next[i] 建边这就是一棵树， 公共最长border就是 lca 。
写一个求 LCA 即可。

KMP 自动机

其实就是维护出 trans[i][c] 表示当前匹配完模板串的第 i 个位置，如果文本串下一个位置的字符是 c， 模板串最长可以匹配到哪里。

这东西相当于一个预处理小技巧。

KMP 自动机

```
n = strlen (s + 1);
m = strlen (t + 1);
for (int i = 2, j = 0; i <= m; i ++) {
    while (j && t[j + 1] != t[i]) j = p[j];
    if (t[j + 1] == t[i]) ++ j;
    p[i] = j;
}
for (int i = 0; i < m; i ++) {
    for (int c = 0; c < 26; c ++) {
        if (t[i + 1] == c + 'a') trans[i][c] = i + 1;
        else trans[i][c] = trans[p[i]][c];
    }
}
```

CF808G Anthem of Berland

给定 s 串和 t 串，其中 s 串包含小写字母和问号，t 串只包含小写字母。

假设共有 k 个问号。

你需要给把每个问号变成一个小写字母，共有 26^k 种可能。

对于每种可能，设 t 匹配 s 的次数为 f_i，请输出 $\max(f_i)$ 。

$1 \le |s|, |t| \le 10^5, |s| \cdot |t| \le 10^7$

CF808G Anthem of Berland

建立 KMP 自动机， 设 dp(i , j) 表示匹配到文本串位置i， 当前在模板串位置 j的最大匹配次数即可简单转移。

P3193 [HNOI2008] GT考试

题目描述

复制Markdown 展开

阿申准备报名参加 GT 考试，准考证号为 N 位数$X_1, X_2 \dots X_n\ (0 \le X_i \le 9)$，他不希望准考证号上出现不吉利的数字。 他的不吉利数字$A_1, A_2, \cdots, A_m\ (0 \le A_i \le 9)$ 有 M 位，不出现是指 $X_1, X_2 \cdots X_n$ 中没有恰好一段等于 $A_1, A_2, \cdots, A_m$，A_1 和 X_1 可以为 0。

输出格式

阿申想知道不出现不吉利数字的号码有多少种，输出模 K 取余的结果。

输入输出样例

输入 #1 复制

```
4 3 100
111
```

输出 #1 复制

```
81
```

说明/提示

数据范围及约定

对于全部数据，$N \le 10^9$，$M \le 20$，$K \le 1000$。

P3193 [HNOI2008] GT考试

考虑设 f[i][j] 表示当前考虑到第 i 位，当前 A 串匹配到第 j 位。

$$f[i][j] \times g[j][k] \to f[i+1][k]$$

g[j][k] 表示当前填一个字符以后，原本 i 这个位置可以匹配的最长长度是j, i + 1 这个位置填完数以后可以匹配的最长长度是 k。

g是固定的，所以可以矩阵快速幂。

感谢