1 T1 s1mple

author: Kewth

1.1 Algorithm 1

对于 $q=1$ ，暴力状压，一个一个考虑，设 $f(S,i)$ 表示前面若干个数使用的数的集合为 S ，最后一个是 i 的方案数。

复杂度 $O(q\,2^n\,n^2)$ 。

1.2 Algorithm 2

M 的每行至多只有一个 1 ，应该有一些奇怪的做法。

此时沿用 Algorithm 1 的暴力 DP 单次复杂度是 $O(2^n\,n)$ 的。

1.3 Algorithm 3

说实话不知道这题该怎么给部分分。~~不过没关系，切掉就完事了~~

不妨直接求出所有可能的 01 序列的答案，记 $A(S)$ 表示 a 压缩为 S 时的答案。

直接计算 A 是很困难的，因为除了 1 的限制外还有 0 的限制。 考虑计算 A 的父集和 B ，即 $B(T)=\sum_{T\subseteq S}A(S)$ ，然后通过反演还原 A 。 那么这样就只有 1 的限制， $B(S)$ 的意义就是钦定排列一些相邻位置 p_i,p_{i+1} 满足 $M_{p_i,p_{i+1}}=1$ ， 而未被钦定的位置没有任何限制。

可以发现，如果将 $M_{i,j}=1$ 看作连边 (i,j) ，$B(S)$ 相当于将排列划分为若干连续段，要求每个段内必须连成一条链。 那么对于 $B(S),B(T)$ ，如果 S,T 代表的划分本质相同，就一定有 $B(S)=B(T)$ 。 也就是说只需要对于所有 n 的划分计算 $B(S)$ 即可。

沿用 Algorithm 1 的做法暴力计算，复杂度 $O(p(n)\,2^n\,n^2)$ ，其中 $p(n)$ 是 n 的划分数。

1.4 Algorithm 4

预处理出 $f(S)$ 表示用集合 S 内的点能组成多少条链。 那么对于一个划分的答案，就是所有对应长度的点集划分的 f 的乘积和。 枚举点集划分是不可取的，事实上，将对应的 f 做一个并卷积，最终序列的全集位置的值就是答案。 因为确定了长度划分，如果若干集合存在一对有交，它们的并的长度就会小于 n ，不可能是全集。 具体的，设 $F_k(S)=[|S|=k]\,f(S)$ ，那么一个划分的答案就是对应的 F_k 的并卷积的末项。

复杂度 $O(p(n)\,2^n\,n+2^n\,n^2)$ ，实现好的话能做到 $O((\sum_{i=1}^n p(i))\,2^n+2^n\,n^2)$ 甚至 $O(p(n)\,2^n+2^n\,n^2)$ 。

ps: 17 的划分数是 297。

2 T2 s2mple

author: justin_cao

2.1 10 分做法

直接哈希找到所有本质不同的字符串，然后对于每次询问暴力枚举匹配即可，复杂度$O(n^4)$。

2.2 20 分做法

先像上面那样对每次询问的模式串用$O(n^2)$求出$f[l][r]$为$s[l\sim r]$区间中，模式串的出现次数。对于每次询问枚举本质不同的串利用$f[l][r]$求出出现次数求和即可，复杂度$O(n^3)$。

2.3 40 分做法

建出SAM，并预处理出SAM上每个点的任意一个right，枚举SAM上的每个节点，可以发现，每次要询问的是对于一个右端点，左端点在一个区间之内的模式串的出现次数之和。于是可以先哈希出模式串在原串中的每个出现位置之后，把上述的若干个询问按照右端点排序，枚举右端点做一遍扫描线即可，复杂度$O(n^2\log n)$。

2.4 50 分做法

出题人也不会$O(n^2)$的做法，这档部分分放在这里看有没有神仙会$O(n^2)$的算法。

2.5 另外的 10 分做法

这档部分分只是用来提示满分做法的，如果能做这个却不能做满分的话，那出题人也只能给你10分当做安慰了。

2.6 100 分做法

首先对题意转化，可以发现我们要求的实际上就是在 $s[l\sim r]$ 的前面加一些字符，在它的后面也加一些字符，使得这个串是原串的子串的方案数。

而方案本质不同，当且仅当前面加的字符**或者**后面加的字符不一样。

那么可以发现，这样的转化能计数恰好符合题目要求的字符串个数。

考虑怎么计数转化后的字符串。

在前面加字符，显然是需要找到以 $s[l\sim r]$ 为后缀的本质不同的 s 的子串，这些串就是在 s 的后缀自动机的 $parent$ 树上 $s[l\sim r]$ 所对应的节点的子树内的节点。

那么接下来就考虑这些字符串在后面加字符的方案了。对于 $right$ 集合相同的串，显然在后面加字符的方案是一样的，于是我们只用对每个后缀自动机的节点单独考虑。

对于一个 $right$ 集合，设一个位置 $pos\in right$集合，那么实际上在后面加不同串的方案就是这个 $right$ 集合内所有的 pos 对应的以 $pos+1$ 为开头的后缀的本质不同的前缀个数。

看到这个后缀的本质不同前缀个数很容易想到后缀数组。那么我们可以用 set 的启发式合并或者线段树合并来找出 $right$ 集合，那么方案数就是这些后缀的长度 $-$ 这些后缀在后缀数组中排名相邻的后缀的最长公共前缀长度+1（包括空集），这个可以用上面两个东西很方便的维护。

然后对 $parent$ 树求个子树和即可，对于 $s[l\sim r]$ 对应的节点，特殊计算一下即可。

用线段树合并复杂度 $O(n\log n)$，set 启发式合并 $O(n\log^2 n)$。

3 T3 s3mple

author: hjsjhn

3.1 Subtask1

暴力枚举所有排列然后判断是否满足条件

3.2 Subtask2 & Subtask3

首先发现对于一个区间 $[l,r]$ 如果 p_{l-1},p_{r+1} 比区间内每一个数都大，那么只需要枚举 $[l,r]$ 中数最大的位置 k ，计算他的贡献，然后再分别计算 $[l,k-1],[k+1,r]$ 就行了，显然接下来的两个区间也满足之前的两端点旁边最大的性质

那么可以考虑设计状态 $f(l,r,x)$ 表示区间 $[l,r]$ 中满足 $\sum_{i=l}^{r} v_i = x$ 的排列方案，那么可以枚举最高点以及左右边的分配转移

$$f(l,r,x)=\sum_{k=l}^{r}\sum_{y=1}^{x-t} f(l,k-1,y)\cdot f(k+1,r,x-t-y)\binom{r-l}{k-l}$$

其中 $t = min\,(k-l+1, r+1-k)$

直接 dp 复杂度 $O(n^3x^2)$ ， 可以通过 Subtask3

通过~~打表~~观察可以发现 x 超过 $n\log n$ 级别时是无解的， 所以可以把 x 降到 $O(n\log n)$ 的级别，总复杂度 $O(n^5\log^2 n)$， 可以通过 Subtask2

3.3 Subtask4 & Subtask5

其实可以将这个结构看成一颗笛卡尔树，那么一个区间的代表树的根为数最大的那个位置 k ，并且他的两颗子树分别是 k 分隔开的左右区间的代表树，并且根的贡献是两颗子树中较小的那颗的大小

那么可以证明这棵树为满二叉树时 x 可以取最大值， x 的级别是 $O(n\log n)$ 的，所以 $x > n\log n$ 时一定无解

所以可以将 dp 改为只跟子树大小 n 和 区间 v_i 之和 m 有关的式子：

$$f(n,m)=\sum_{i=0}^{n-1}\sum_{j=0}^{m} f(i,j)\cdot f(n-1-i, m-min\,(i,n-1-i)-j)\cdot\binom{n-1}{i}$$

如果没注意到 x 的上限直接 dp 复杂度 $O(n^2x^2)$ ， 可以通过 Subtask5

注意到 x 的上限后复杂度为 $O(n^4\log^2 x)$ ， 可以通过 Subtask4

3.4 Subtask6

将状态看做关于 x 的 m 次多项式 $F(n)$ ，发现转移其实是枚举 i 做卷积，那么式子可以变为：

$$F(n)=\sum_{i=0}^{n-1} F(i) * F\,(n-1-i)\binom{n-1}{i}x^{min(i,n-1-i)}$$

做到了这一步， 应该不可能不注意到 x 是 $O(n\log n)$ 的级别了， 所以没给 x 很小的分

由于模数为 998244353 所以可以暴力上 NTT ， 复杂度 $O(n^3\log^2 n)$ ， 可以通过 Subtask6

3.5 Subtask7

应该是一个烂大街的技巧了

可以只保留 $F(n)$ 的点值做卷积，那么转移复杂度为 $n^3 \log n$

对于每组询问，可以用 lagrange 插值以 $O(n^2 \log^2 n)$ 的复杂度插出对应次数的系数就行了

总复杂度 $O(n^3 \log n + n^2 \log^2 n)$ ， 可以通过所有数据

3.6 总结

本题应该是本场比赛里一道比较简单的题目， 由于数据范围不大， 所以部分分不是特别容易区分，可能出现水过某些 Subtask 的情况， 也可能出现部分由于常数太大而被卡掉部分点的情况， 所以我把部分分出的比较多， 每个部分分的分数也分布的比较均匀， 希望实际分数不会与预计分数差别太大