The Preliminary Contest for ICPC Asia Nanjing 2019

The beautiful values of the palace

本题一个思维点 通过 X，Y 的坐标求出相应数字，可以通过分类讨论，或者推公式找出结论

地图大小为 $n \times n$ n一定为奇数

$x = x - n/2 - 1$;

$y = x - n/2 - 1$

$t = \max(abs(x), abs(y))$； //确定该点在第几圈螺旋

$if(x >= y)ans = N * N - 4 * t * t - 2 * t - x - y$; //在向右与向上的路线上

$elseans = N * N - 4 * t * t + 2 * t + x + y$; //在向左与向下的路线上

对于这个题而言

就转化为求任意子矩阵之和

离散化 宫殿和询问的y坐标

对于每个询问要拆成4次询问

$map[i][j]$ 代表 1，1 这点为左下节点 i，j 为右上节点所求得矩阵和

对于查询 $x1, y1, x2, y2$

$ans = map[x2][y2] - map[x2][y1-1] - map[x1-1][y2] + map[x1-1][y1-1]$
;

因此一次询问就可以看出求求四次二维前缀和 标记每次询问是加还是减；

将宫殿和询问按 x 坐标排序

维护一个带修树状数组

每加入一个宫殿添加其值

每加入个询问就求其前缀和

super_log

题目要求算出幂塔函数 $a^{a^{a^{\cdots}}}$ 共 b 个 a 乘幂的结果模m的值。由于指数会非常大，使用欧拉定理递归降幂即可。需要特判一下 a 与 m 不互质的情况。

欧拉函数很快会迭代到 1，可以提前退出，速度很快。

Tsy's number 5

设 $f_i=\sum\limits_{k=1}^{n}[\varphi(k)=i]$

则原式等价于

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{n}f_if_jij2^{ij}\\
=&2\sum_{i=1}^{n}\sum_{j=1}^{i}f_if_jij2^{ij}-\sum_{i=1}^{n}i^2f_i^22^{i^2}\\
=&2\sum_{i=1}^{n}if_i\sum_{j=1}^{i}jf_j\sqrt{2}^{i^2+j^2-(i-j)^2}-\sum_{i=1}^{n}i^2f_i^22^{i^2}\\
=&2\sum_{i=1}^{n}if_i\sqrt{2}^{i^2}\sum_{j=1}^{i}jf_j\sqrt{2}^{j^2}\sqrt{2}^{-(i-j)^2}-\sum_{i=1}^{n}i^2f_i^22^{i^2}
\end{aligned}
$$

第二层求和为考虑 NTT 卷积即可

Robots

首先发现这题的转移是个经典的概率dp转移即去解 $dp[u] = dp[u] \cdot \frac{1}{outdeg[u]+1} + \frac{1}{outdeg[u]+1} \cdot \sum dp[v] + 1$ 来代表天数期望，之后再用所求的天数期望去再通过类似的转移求花费价值的期望。由于此图为有向无环图，所以跑边拓扑边转移即可

K Sum

1. 考察点

积性函数、莫比乌斯反演、杜教筛（狄利克雷卷积）、等比数列求和、欧拉降幂、数论分块以及推式子的能力。

本题用到的都是上述内容中比较基础的知识点。

2. 解题过程

$$
\begin{aligned}
f_n(k) &= \sum_{l_1=1}^{n}\sum_{l_2=1}^{n}\cdots\sum_{l_k=1}^{n}(\gcd(l_1,l_2,\cdots,l_k))^2 \\
&= \sum_{i=1}^{n}\sum_{l_1=1}^{n}\sum_{l_2=1}^{n}\cdots\sum_{l_k=1}^{n} i^2[\gcd(l_1,l_2,\cdots,l_k)=i] \\
&= \sum_{i=1}^{n}\sum_{l_1=1}^{\lfloor\frac{n}{i}\rfloor}\sum_{l_2=1}^{\lfloor\frac{n}{i}\rfloor}\cdots\sum_{l_k=1}^{\lfloor\frac{n}{i}\rfloor} i^2[\gcd(l_1,l_2,\cdots,l_k)=1] \\
&= \sum_{i=1}^{n}\sum_{l_1=1}^{\lfloor\frac{n}{i}\rfloor}\sum_{l_2=1}^{\lfloor\frac{n}{i}\rfloor}\cdots\sum_{l_k=1}^{\lfloor\frac{n}{i}\rfloor} i^2\sum_{d=1}^{\lfloor\frac{n}{i}\rfloor}\mu(d)[d|l_1][d|l_2]\cdots[d|l_k] \\
&= \sum_{i=1}^{n} i^2\sum_{d=1}^{\lfloor\frac{n}{i}\rfloor}\mu(d)\sum_{l_1=1}^{\lfloor\frac{n}{id}\rfloor}\sum_{l_2=1}^{\lfloor\frac{n}{id}\rfloor}\cdots\sum_{l_k=1}^{\lfloor\frac{n}{id}\rfloor} 1 \\
&= \sum_{i=1}^{n} i^2\sum_{d=1}^{\lfloor\frac{n}{i}\rfloor}\mu(d)\lfloor\frac{n}{id}\rfloor^k \\
&= \sum_{l=1}^{n}\lfloor\frac{n}{l}\rfloor^k\sum_{d|l}\mu(d)\cdot\frac{l^2}{d^2}, l=id
\end{aligned}
$$

因此，$f_n(k)$ 的前缀和（不考虑 $k=1$ 的情况）也可得到：

$$\begin{aligned}\sum_{i=2}^{k} f_n(i) &= \sum_{i=2}^{k}\sum_{l=1}^{n}\lfloor\frac{n}{l}\rfloor^{i}\sum_{d|l}\mu(d)\cdot\frac{l^2}{d^2} \\ &= \sum_{l=1}^{n}(\sum_{i=2}^{k}\lfloor\frac{n}{l}\rfloor^{i})(\sum_{d|l}\mu(d)\cdot\frac{l^2}{d^2})\end{aligned}$$

其中，$\sum_{i=2}^{k}\lfloor\frac{n}{l}\rfloor^{i}$ 为等比数列的前 i 项和减去第 1 项，因为 k 很大，根据欧拉定理，可以将 k 对 10^9+6 取模之后，再进行处理。基于等比数列的前 i 项和公式，可以通过快速幂处理出这部分答案。**注意特判公比为 1 的情况**。

设 $g(l)=\sum_{d|l}\mu(d)\cdot\frac{l^2}{d^2}$，每一轮计算我们需要得到 $g(l)$ 的前缀和。

显然，$g(l)$ 为积性函数，注意到 $n\le 10^9$，因此我们可以通过线性筛预处理出前 10^6 项的前缀和，后面的项的前缀和可以通过杜教筛求出，求这一部分答案的时间复杂度为 $O(n^{\frac{2}{3}})$。

我们发现，$g(1)=1, g(p)=p^2-1, g(p^k)=p^{2k-2}(p^2-1)$，利用这部分信息就可以完成线性筛求和的部分。

我们又发现，$g=\mu*Id^2$，因此我们可以对 g 卷上 I，得到
$g*I=(\mu*Id^2)*I=\mu*I*Id^2=Id^2$，利用这部分信息可以完成杜教筛的求解。

对于最外层，直接数论分块即可。

至此，整个求解过程就已完成。

总的时间复杂度为 $O(n^{\frac{2}{3}}+\sqrt{n}\cdot\log 10^9)$。

如果考虑 T 组的话，总的时间复杂度为 $O(T\cdot(n^{\frac{2}{3}}+\sqrt{n}\cdot\log 10^9))$。

Greedy Sequence

首先贪心的思想，每个元素的下一个元素一定是与他距离不超过 k 的所有元素中，权值最大的元素，所以每个元素的下一个元素是固定的，我们可以通过滑动窗口 + set 上二分的预处理计算出每个元素的下一个元素，之后通过一个记忆化搜索即可 O(n) 求出每一个贪心序列的长度。

Quadrilateral

解题思路：

1、对于 a，b，c，d 能够组成四边形的情况，要求任意三边的长度之和必须大于第四条边的长度。

2、因为测试样例数为 1000，d 的范围也在 1000 以内，所以先通过枚举 d 的长度，来保证当前的复杂度为 10^6 以内。

3、在枚举每个 d 的长度的情况下，其它三条边 a，b，c 各自对应一个范围[La,Ra],[Lb,Rb],[Lc,Rc]。如果我们能计算出其范围表示为从 1 开始的情况，如[1,Ra],[1,Rb],[1,Rc]的情况（简化这种表示为(Ra,Rb,Rc)），那么就能通过容斥原理，得到最后的结果。最后的结果为（Ra，Rb，Rc）-（La-1，Rb，Rc）-（Ra，Lb-1，Rc）-（Ra，Rb，Lc-1）+（La-1，Lb-1，Rc）+（La-1，Rb，Lc-1）+（Ra，Lb-1，Lc-1）-（La-1，Lb-1，Lc-1）。

4、对于四条边分别为[1,Ra],[1,Rb],[1,Rc],Rd 的情况，最多可以组成的四边形种类为 Ra*Rb*Rc，之后分别枚举四条边各自为最长边的情况下，不能形成四边形的种类数。假设 a 为最长边，则不能形成四边形的情况为 b+c+d<=a，即[1,Rb]+[1,Rc]<=[1-Rd,Ra-Rd]（也就是[1,Ra-Rd]）。对于 d 为最长边时，不能组成四边形的情况为 a+b+c<=d，即[1,Rb]+[1,Rc]<=[Rd-Ra,Rd-1]，可以转化为[1,Rb]+[1,Rc]<=[1,Rd-1]的种类数减去[1,Rb]+[1,Rc]<=[1,Rd-Ra-1]的种类数。

5、之前我们已经把问题转化成了类似于求解[1,Ra]+[1,Rb]<=[1,Rc]形式的种类数的问题。（也就是[1,Ra]+[1,Rb]-1<[1,Rc]）在这里我们要用 O（1）的复杂度进行求解。首先假设 Ra<=Rb，如果 Ra>Rb 则交换两者即可。再假设 Ra=5，Rb=9，则 a 与 b 能够组成的 5*9=45 种类型可以转化为下面表格的形式。表格中 5 列分别对应 a=[1,5]，每列里面包含 9 个数值，对应 b=[1,9]，每个位置的数值为 a+b-1。

对于每一个位置的数值 k，其满足<[1,Rc]的种类数就是 Rc-k。因此，如果我们算出前 Rc 行的位置数 m，和前 Rc 行每个位置的数值和 sum，就能求解出满足[1,Ra]+[1,Rb]-1<[1,Rc]的种类数为 Rc*m-sum。

我们可以将表格中的行分为 3 个部分。(1)前 Ra 行(也就是前 5 行)的位置数为 1+2+⋯+5（高斯算法），所有位置上的和为 1^2+2^2+⋯+5^5 (平方和公式)。(2) 中间 Ra+Rb-1-2*Ra 行，每行包含固定为 Ra 个数值，计算方式更为简单。(3) 后 Ra 行也可以利用类似于第 1 部分的方法去进行计算。

因为我们只计算前 Rc 行，所以 Rc 可能会将上述 3 个部分中的某一个部分截断，此处需要细心处理。

1				
2	2			
3	3	3		
4	4	4	4	
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
	10	10	10	10
		11	11	11
			12	12
				13

6、通过上述步骤，就能算出组合成的四边形的种类数，注意最后的结果要用 64 位进行输出。

Holy Grail

对于题目中给定的任意一个点对 S、T，记要添加的有向边 e_{ST} 的权值为 c_{ST}。显然，当 $c_{ST} = -dis_{TS}$时，c_{ST}即为我们要求的解，其中 $-dis_{TS}$为从 T 到 S 的最短路径长度取反后的值。

每次求得一条边的权值后将该边添加进原图再次跑 **Bellman-Ford** 算法即可。

对于每个 $case$，共需跑6次 **Bellman-Ford**，时间复杂度为 $O(6 * |V| * |E|)$（二分也可以通过此题）。

Washing clothes

首先，使用洗衣机的人一定是到来时间靠后的人，或者说是一个后缀，为了方便，我们不妨假设这N个人是按到来时间排好序的。如果是从第i个人开始使用洗衣机，那么可以得到此时的答案是$max(t_{i-1}+y,max_{j=i}^{N}t_j+(N-j+1)\times x)$，其中可以认为$t_0=-y$。设$f(i)=t_{i-1}+y,g(i)=max_{j=i}^{N}t_j+(N-j+1)\times x$，显然，$f(i)$是单调递增函数，而$g(i)$是单调递减函数，故$max(f(i),g(i))$为凸函数，故我们的目标变成寻找该函数的极小值点，又因为$g(i)_x\le g(i)_{x+1}$，故该函数的极小值点是随着x的增大而后移的，且$g(i)$其实是一堆直线取max，因此我们可以考虑从大到小枚举x，然后不断的左移极小值点，并用李超树(或其他合适的数据结构)维护这一堆直线以快速查找单点最值，由于这些线段的定义域是相同的，故使用李超树的时间复杂度为$O((n+y)logn)$。