math

$n\le 5$ 直接枚举 X 检验。$n\le 10^4$ 考虑数位 dp，计算 $[1,n]$ 中 $f(X)=A$ 的 X 个数。

套路地设 $f_{i,S,0/1}$ 表示从高到低考虑了前 i 位，使用了 S 中的数码，且当前是否顶满 n 上界的方案数，容易做到 $O(2^B Bn)$，其中 $B=10$，可以得到 70 分。

优化转移没前途，考虑如果在第 i 位第一次不顶上界，那么枚举第 i 位的数码，设 $1\sim i$ 位用的数码集合是 S_1，则第 $i+1\sim n$ 位使用的数码集合 S_2 应该满足 $|S_1\cup S_2|=A$，那么问题转化为任意填后 $n-i$ 位，使用的数码是 S_1 中的或者是 S_1 以外的，但要求恰好使用 $A-|S_1|$ 个 S_1 以外的数码。很容易 dp 出 $g_{i,j}$ 表示填 i 个数码，有 j 个不同数码的方案数，用 g 在统计的时候拼一下答案即可。
$O(nB)$。

triangle

sub1 模拟题意 $O(qn^3)$，sub2,3 注意到如果把可用长度拉出来排序，则使用的木棍一定是相邻三个，假设从小到大排序后的长度是 $b_{1\sim m}$，使用了 $i<j<k$，若 $i+1<j$，则使用 $i+1,j,k$ 一定仍然满足条件，且周长更大或不变；若 $j+1<k$，同理可以把 j 换成 $j+1$，那么就可以做到 $O(qn\log n)$ 了。

sub4 考虑到合法的三角形边长只能是三边长度相等，枚举一下长度 2^k，再查一下区间里有多少个这个数，可以用各种数据结构解决。

仍然考虑区间排序后得到的长度序列 b，一个贪心策略是从大到小试试相邻三个能否组成三角形，直到找到一组可以的就结束。考虑到若 $i,i+1,i+2$ 不能组成三角形，则有 $b_i+b_{i+1}\le b_{i+2}$，又有 $b_i\le b_{i+1}$，所以 $b_i\le\frac{b_{i+2}}{2}$，所以只需要约 $2\log_2 V$ 次就会使得 b_i 不能再变小了。于是我们一定能在前 $O(\log V)$ 大元素里面找到解，使用线段树等数据结构可以轻松维护 $O(c\log n)$ 找区间前 c 大。

$O(q\log V\log n)$。

set

显然需要求出所有 T 的 $f(T)$，sub1,2 暴力算一算，sub3 对于单个 $|T|\ge k$ 可以用组合数算出 $|S_i\cap T|\ge k$ 的 i 的个数为 $\sum_{i=|T|}^{n}\binom{n-|T|}{i-|T|}$。

接下来考虑每个 i 的贡献：设 $F_i(T)=[|S_i\cap T|\ge k]$，则 $f(T)=\sum F_i(T)$。对于 sub4，只需要求出 $F_i(T)=[S_i\subseteq T]$，那么以 S_i 为下标的桶做高维前缀和得到 F_i，将这些桶叠加起来一起做高维前缀和就可以得到 f 了。

更一般地，要想求出 F_i，不妨对其做 FMT 得到 f_i，然后再高维前缀和回来得到 F_i，也就是说要找到一个 f_i 满足 $\sum_{X\subseteq T} f_i(X)=F_i(T)$。设 $|S_i\cap T|=l$，$l\ge k$，那么 X 中有 $\binom{l}{j}$ 个大小为 j 的 S_i 的子集，如果能找到容斥系数 a 满足 $\sum_{i=k}^{l}a_i\binom{l}{i}=1$，那么将 $X\subseteq S_i\wedge|X|\ge k$ 的 $f_i(X)$ 设置为 $a_{|X|}$，其余设置为 0，就能满足 f_i 的定义。而 a 显然是好求的，只需要 $O(n^2)$ 按照定义递推一下，即 $a_k=1$，$a_l=1-\sum_{i=k}^{l-1}a_i\binom{l}{i}$（$k<l\le n$），$a_l=0$（$l<k$）。

得到了 f_i 的计算式，那么令 $h=f_1+f_2+\cdots+f_m$，对 h 做高维前缀和就能得到 f 了。此时发现 $h(X)=a_{|X|}\times\sum_{i=1}^{m}[X\subseteq S_i]$，于是预先做一次 S_i 的桶的超集和得到后面的因子，再分别乘上 $|X|$ 即为 h。$O(m+2^n)$。

注意到本题没有取模，所以计算 a_l 时需要估计其量级或是进行额外处理。

首先可以证明 $a_l=(-1)^{l-k}\binom{l-1}{k-1}$，可以验证其符合 a 的定义，则 a_l 的范围在 long long 以内。如果不能得到这个系数的表达式，也可以考虑到最终答案都是 $\le m$ 的，只需要在 $\bmod$ 一个比 m 大的数的条件下计算，得到的答案也是准确的。比如可以在 unsigned int 下操作。

graph

对于 $m=n-1$ 的树的情况显然是沿唯一最短路径来回走一次，代价是路径边权和。则问题为 $dis(u,v)\le k$ 的 (u,v) 对数，这是经典的点分治问题，可以拿到 40 分，不会写点分治也能拿到 15 分。

接下来是 sub4：$k\le 100$。在一个边双内部，任意两点间均有两条边不交的路径，则这些点之间的 f 值均为 0，需要统计。如果 u,v 在不同边双内，用圆方树把同一个边双缩成一个点，则图的结构变成树，不同边双之间有至多一条边，则问题转到树上。$k\le 100$ 可以直接在 LCA 处统计答案：dfs 过程中记录子树内距离 u 为 i（$0\le i\le k$）的点的数量，前缀和之后拼起来即可，需要容斥掉两个点出自同一子树的情况。发现圆方树的做法之后也可以顺便拿下 sub5,6。

整道题的做法就是把上述算法拼起来，先缩点，再在圆方树上跑点分治。$O(m+n\log n)$。