E

分类讨论，对每种情况构造能达到上界的方案

1. x是偶数，输出x/2，当x最高位是1时得到LCS=|x|-1，否则得到LCS=|x|
2. x是奇数
   1. x最高位是1且x%11!=10，构造floor(x/11)和x-10*floor(x/11)，得到LCS=|x|-1
   2. x所有位均为9，构造4545...和5454...，得到LCS=|x|-1
   3. 否则考虑最低位非9数字c，记str(x)=str(p)+str(c)+str(999..99)（p可能为0，但是不影响后续分析，以下构造结果需要去除前导0）
      1. 如果p是偶数，构造str(p/2)+str(c09090...)和str(p/2)+str(090909...)，当x最高位是1时得到LCS=|x|-2，否则得到LCS=|x|-1
      2. 如果p是奇数，构造str((p-1)/2)+str((c+1)90909...)和str((p-1)/2)+str(909090...)，当x最高位是1时得到LCS=|x|-2，否则得到LCS=|x|-1

A

令 G/u 表示从朋友关系图 G 中删除点 u 以及所有相邻边后剩下的图，令 $f(G,u)$ 表示图 G 中，如果 u 被最先选为龙，是否能够存活。那么可以发现 $f(G,u)$ 为真，当且仅当对于任意的 v 不是 u 的朋友，$f(G/u,v)$ 均为假。

该状态压缩动态规划等价于一个著名的博弈模型：一张无向图，从一个点出发，两个人轮流移动，经过的点不能再经过，谁不能移动谁输。因此，根据这个博弈模型的结论，$f(G,u)$ 为真当且仅当存在一个 G 的最大匹配，u 不在匹配中。

一般图最大匹配，用带花树求解，时间复杂度 $O(n^3)$。

B

分治，假设当前分治区间为 $[l,r]$，中点为 m，一次性计算 $[l,r]$ 的所有跨过中点 m 的子区间的最大子段和。对于 $i \in [l,m]$，预处理 f_i 为 $[i,m]$ 的最大后缀和，w_i 为 $[i,m]$ 的最大子段和；对于 $i \in [m+1,r]$，预处理 f_i 为 $[m+1,i]$ 的最大前缀和，w_i 为 $[m+1,i]$ 的最大子段和。那么，跨过 m 的区间 $[a,b]$ 的最大字段和为 $\max(w_a, w_b, f_a+f_b)$。分类讨论后，可以发现能够用树状数组来快速求解。时间复杂度 $O(n\log^2 n)$。

C

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}[d(i,j,T)] = n\sum_{i=1}^{n}\mathbb{E}[\mathrm{depth}(i)] - 2\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d|(i,j)}\Pr[\mathrm{lca}(i,j)=d]\mathbb{E}[\mathrm{depth(d)}]$$

计算所有的 $\mathbb{E}[\mathrm{depth}(i)]$ 是 trivial 的，可以在 $O(n\log n)$ 的时间内完成。后半部分的概率通过莫比乌斯反演+容斥可以在 $O(n\log^2 n)$ 的时间复杂度内算出来。

D

定义从字符串到整数数组的变换 $f(s)$ 为：如果 s_i 是该字符在 s 中的第一次出现，那么 $f(s)_i=0$，否则设上一次出现为 s_j，那么 $f(s)_j = i-j$。可以发现两个子区间 x,y 对应的表演一样当且仅当 $f(x)=f(y)$。

同时观察可以发现，对于 s 的子区间 x，$f(x)$ 几乎是 $f(s)$ 的子串，除了 x 中每一个字符出现的第一个位置变成了 0。因此 $f(x)$ 可以看做 $f(s)$ 的 26 个子串用 0 拼接得到。因此，对 $f(s)$ 求解后缀数组之后，对于 s 的任意两个后缀 x,y，我们都可以在 26 的线性时间内求出 $f(x),f(y)$ 的最长公共前缀与大小关系。

带上一个排序就做完了，时间复杂度为 $O(n\log n\times 26)$。

E

大分类讨论。

F

关键点为起点加上所有多边形的顶点。首先用最短路求出到所有关键点的最短路。假设关键点 P_i 的最短距离为 $d_i(d_i \le K)$，那么加入关键圆 C_i，它以 P_i 为圆心，半径为 $C_i - d_i$。

可以发现，最终图形的所有边界一定是关键圆上的圆弧，以及多边形的边上面的线段，利用格林公式，只需要找出边界上的所有线，算一圈贡献就行了。

对于每一个圆弧，求出其他所有关键圆/边与它的交点，拆分成圆弧。对于每一段圆弧，判断其中点到起点的距离是否是 K，如果是，那么这一段就在边界上。

对于每一条线段，求出其他所有关键圆/边与它的交点，拆分成线段。对于每一段咸蛋，判断其重点到起点的距离是否小于 K，如果是，那么这一段就在边界上。

求距离只需要看这个点到每一个关键点的线段是否和多边形有交就可以了。时间复杂度 $O(n^4)$ 至 $O(n^5)$，都能过。

G

所有为 0 的点一定是原图的一个极大独立集，所有为 1 的点一定是去掉为 0 的点后的一个极大独立集，以此类推。直接状态压缩动态规划，时间复杂度 $O(3^n)$。

H

对于每一棵子树，令 $f(x)$ 为这棵子树的父亲的海拔是 x 的时候，这棵子树内部的所有边的最小代价。通过归纳法，可以发现 $f(x)$ 是一个二次函数，并且我们可以在 $O(1)$ 的时间内从两个子树的 $f(x_l), f(x_r)$ 合并出整棵树的 $f(x)$。因此对于更新的时候，我们只需要把修改的节点到根路径上的所有二次函数更新一下就可以了。

时间复杂度 $O(n \log n)$。

I

可以发现第一种情况不会发生，因为第一种情况发生的唯一好处是，积累到 K 天不回答问题，帮助后面节约代价。但是节约的代价一定小于等于 T，而超过阈值后超出的代价一定大于 T，因此这样肯定没有每天都回答问题赚。

可以写出动态规划式子，用单调队列优化，即可做到 $O(n)$。

J

通过简单的推导可以发现，假设从下到上第 i 本书到书桌边界的距离最大，那么更上面的 $n-i$ 本书的整体重心一定在第 i 本书的左边界，而对于所有的 $j \in [1, i]$，第 j 本以及更往上的书的整体重心一定在第 $j-1$ 本书的右边界。简单的状态压缩动态规划即可找到最优解，时间复杂度 $O(n2^n)$。

K

会当随机过程继续的数字必须满足它的所有前缀都不是合数。这样的数字很少，因此直接暴力模拟就可以了。

L

令 A_i 表示将军 i 控制了第 A_i 个城市，$f(i,j)$ 表示前 $i-1$ 个将军的选择已经固定，第 i 个将军控制的是第 j 个城市的方案数，那么只有三种情况：

1. 第 i 个将军最终仍然控制第 j 个城市，递归到 $f(i+1, A_{i+1})$。
2. 第 i 个将军的城市变动了，且 j 与 A_{i+1} 相邻，递归到 $f(i+1, j)$。
3. 第 i 个将军的城市变动了，且 j 与 A_{i+1} 不相邻。令 A_k 为 $i+1$ 后第一个与 j 相邻的城市，那么这种情况能够发生当且仅当 A_{i+1} 至 A_{k-1} 均与 A_k 相邻，递归到 $f(k, A_{k-1})$。

简单分析可以发现状态数是 $O(n)$ 的，直接记忆化搜索就可以了。

M

对原树进行轻重链剖分；
假设当前在处理以w为根的大小为n的子树，且w是重链头，当前莫队状态为w的子树补。
可以在O(n)时间扫出w所在重链上，每个结点的子树补（这部分总代价为O(nlogn)）；
然后将w所在重链的每个点的轻儿子找出来，之后要递归处理；
将这些轻儿子按子树大小加权，以及重链上每个点按1加权，建哈夫曼树，在哈夫曼树上dfs，维护哈夫曼树上当前点的子树补，到达叶子时若叶子对应轻儿子则递归处理轻子树。
将整个递归过程中，产生的哈夫曼树拼接起来，得到所有叶子权为1的哈夫曼树，由于哈夫曼树上，每向下两层子树权值和至少减小一个常数比例，由此得到遍历哈夫曼树过程的莫队的复杂度是O(nlogn)的。
因此总时间O(nlogn)。

使用轻重链剖分

1. 对根所在重链，可以在n次插入得到重链上每个点的答案，
2. 轻子树需要建立哈夫曼树，叶子的权重为子树规模，重链也作为一个叶子，权重为重链长度；
   在哈夫曼树上dfs，维护当前子树外对应的集合，移动到叶子时就递归到了轻子树上的子问题，所需插入次数为所有非叶结点的子树规模之和（子树规模为子树中叶结点的权重之和）；