A

容易发现，我们加的边一定是从叶子到根，所以用树形dp简单计算即可。

B

首先，记$\omega(n)$为n的不同素因子个数，那么$f(n) = 2^{\omega(n)}$。

可以发现$f(ni) = f(n)f(i)/f((\gcd(n,i)))$，也就是一个数如果同时是$n, i$的因子，那么会计算重复，需要除掉。令$g = (1/f) \times \mu$，那么上式可以写成$f(ni) = f(n)f(i) * \sum_{d|n,d|i} g(d)$。

所以答案为$f(n) \sum_{d|n} g(d) \sum_{i=1}^{\lfloor n/d \rfloor} f(id)$，由于空间问题，后面这部分可以考虑离线计算。

时间复杂度为$O(n \log n + \sum d(n))$.

C

通过找规律可得，所有的必败态为 $(\lfloor \frac{m(1-k+\sqrt{k^2+2k+5})}{2} \rfloor, \lfloor \frac{m(3+k+\sqrt{k^2+2k+5})}{2} \rfloor)$.

以下是证明：

First we have $F(0,0) = 0$, then $\forall x,\ F(0,x) = F(x,x) = 1$. Therefore, next minimum x that $\exists F(x,y) = 0$ is 1, which have $F(1,2) = 0$, then $\forall x,\ F(1,x) = F(2,x) = F(x,x+1) = 1$. After that we know the next (x,y) that $F(x,y) = 0$ is $(3,5)$. Follow these steps we can get all $F(x,y) = 0$. It's not hard to find that for each m, There are an unique x that $F(x,x+m) = 0$.

Suppose $S_m = \{x | F(x,x+m) = 0\} \cup \{x+m | F(x,x+m) = 0\}$, we can find that the x which $F(x,x+m+1) = 0$ is the minimum positive number that haven't appeared in S_m, aka $mex(S_m)$.

Then let's generalize the conclusion for situations that $0 \le k \le 1e9$.

We still have $F(0,0) = 0$ first, and then $\forall x > 0, 0 \le a \le k,\ F(0,x) = F(x,x+a) = 1$, thus the next (x,y) that $F(x,y) = 0$ is $(1,k+2)$. And so on, the m-th (x,y) we can get satisfies $y - x = (m-1)(k+1)$.

Suppose $S_m = \{x | F(x,x+m(k+1)) = 0\} \cup \{x+m(k+1) | F(x,x+m(k+1)) = 0\}$, we still have the minimum x that $F(x,x+(m+1)(k+1)) = 0$ is $mex(S_m)$.

Now prove that the m-th (x,y) that $F(x,y) = 0$ is $(\lfloor \frac{m(1-k+\sqrt{k^2+2k+5})}{2} \rfloor, \lfloor \frac{m(3+k+\sqrt{k^2+2k+5})}{2} \rfloor)$.

For all (x,y) that $F(x,y) = 0$(except $(0,0)$), suppose the ascending series combines of every x is a and the ascending series combines of every y is b.

What can be inferred from the previous conclusion is $a_i + m(k+1) = b_i$ and $a \cup b = \mathbb{N}^*$. Besides, $a_i = mex(\{a_1, a_2, \ldots, a_{i-1}, b_1, b_2, \ldots, b_{i-1}\})$.

Then prove that $A_i = (\lfloor \frac{m(1-k+\sqrt{k^2+2k+5})}{2} \rfloor, B_i = \lfloor \frac{m(3+k+\sqrt{k^2+2k+5})}{2} \rfloor$ is what we want.

First of all, we need to prove $A \cup B = \mathbb{N}^*$, by proving the number of x in A and B that $x \le n$ is exactly n (It's easy to see that $A \cap B = \phi$ from the proving of previous conclusion).

Let $x = \frac{1-k+\sqrt{k^2+2k+5}}{2}$, $y = \frac{3+k+\sqrt{k^2+2k+5}}{2}$. Then we have $A_i = \lfloor ix \rfloor, B_i = \lfloor iy \rfloor$, then the total number of x in A, B that $x \le n$ is

$$N = \sum_{i=1}^{\infty} [A_i \le n] + [B_i \le n]$$

$$= \sum_{i=1}^{\infty} [\lfloor ix \rfloor < n+1] + [\lfloor iy \rfloor < n+1]$$

$$= \sum_{i=1}^{\infty} [ix < n+1] + [iy < n+1]$$

$$= \lceil \frac{n+1}{x} \rceil - 1 + \lceil \frac{n+1}{y} \rceil - 1$$

$$= \lfloor \frac{n+1}{x} \rceil + \lfloor \frac{n+1}{y} \rfloor$$

$$= \frac{n+1}{x} + \frac{n+1}{y} - \{\frac{n+1}{x}\} - \{\frac{n+1}{y}\}$$

By the reason of $\frac{1}{x} + \frac{1}{y} = 1$, $N = n + 1 - \{\frac{n+1}{x}\} - \{\frac{n+1}{y}\}$.

Besides we have $\{\frac{n+1}{x}\} \ne 0$, $\{\frac{n+1}{y}\} \ne 0$, $\{\frac{n+1}{x}\} + \{\frac{n+1}{y}\} = 1$, then we know $N = n + 1 - \{\frac{n+1}{x}\} - \{\frac{n+1}{y}\} = n$.

Therefore we have proved $A \cup B = \mathbb{N}^*$.

Obviously there is $A_i < B_i$, as A,B are both ascending series. Let $z = mex(\{A_1, A_2, \ldots, A_i, B_1, B_2, \ldots, B_i\})$. If $A_{i+1} > z$, then $\forall p > i+1, A_p > z$ and $\forall p > i, B_p > z$, thus what we can get is z will not appear in A, B, which is a contradiction with $A \cup B = \mathbb{N}^*$.

Besides, because $y = x + k + 1$, we have $B_i = A_i + i(k+1)$.

Now $A = a, B = b$ is proved.

D

这个题背后的想法是离散对数的index calculus算法中的一部分。我们设定一个阈值B，如果一个数的所有素因子都不超过B，那么就是B-smooth的。在这个题里面，我们把B设到了1000，当然你也可以自行调参。

于是，对于一个数，我们可以首先随机乘上一些小的数的逆元，然后尝试用B以下的数字分解，直到分解出来合法的答案。一个小技巧是如果使用比如100以下的素数试除之后，这个数字依然很大，那么我们可以认为大概率它不行，所以可以break来节省时间。

E

考虑两点之间的电阻，首先你把路径拿出来，可以看成若干条树边和环串联起来，也就是相加。所以考虑线性性，也就是对于每个树边和环分别考虑对答案的贡献。

每个树边的贡献为通过树边的点对，也就是两边大小相乘。对于环边，假设两个点下面的子树大小分别为S_i和S_j，环长为l，距离为$j-i$，那么等效电阻为$j-i$与$l-j+i$并联，也就是$(l-j+i)(j-i)/l$，共有S_iS_j对，可以将式子展开之后使用前缀和之类的方法计算，使用FFT可能会TLE。

F

考虑类似做数独的方法，维护每个格子的候选数。一些筛选的方法就是首先维护每行每列根据线索和候选数的可行排列，然后可以根据这些可行排列的并集筛除候选数。比如某一个它在行的所有排列都不能是1，那么它也不能是1。如果筛除了1，那么可以继续去更新列的可行排列，进行进一步的推理。还有就是根据每个数字出现的位置，进行筛选，也就是对于某个数字，看它可能在哪些行或者列，然后枚举所有可行的放法，看某些格子是不是始终不能放，那么也能筛除候选数。这种方法大概能解决85%左右的问题。对于无法解决的，可以用试错法，然后再使用上述规则进行筛选，看是否有矛盾。如果依旧无法解决，那么可以使用迭代加深，也就是先试错一层，如果得不到任何线索，那么考虑试错两层，但是在这个题里面，使用一层的试错可以解决所有的问题。

另外的解决方法是可以把所有的问题爬下来，然后使用本地的解决NPC问题的库(SAT solver之类)或者并不高效的搜索。如果全部手工处理比较的不高效，可以通过阅读js源码之类的发现网址是如何decode成线索的，那么手工只要复制所有的网址就行，可能会比写爬虫快一点。这也是我爬数据时候的方法。

G

直接模拟。每次操作等价于找到连续一段都$\geq y$的段，然后左移一位，使用平衡树维护即可。

H

直接莫队很难高效的计算每次加入一个点或者删除一个点之后的距离和。考虑二次离线莫队，只要支持$O(n)$次加点，$O(n^{1.5})$次查询一个点到这个点集的距离和，这里我们假设n, q同阶。

这个问题可以转化成$O(n)$次链加，$O(n^{1.5})$次链求和，也就是说可以把距离看成深度减去两倍的LCA的深度，那么每次只要求出LCA的深度和即可。每次加点的时候，可以把它到根的边，依次加上边的长度。那么查询的时候刚好是到根的边的和。

这个问题考虑树分块，将树分成$O(\sqrt{n})$个边数不超过$O(\sqrt{n})$的块，那么可以完成每次修改$O(\sqrt{n})$每次查询$O(1)$。

I

插头dp，在求哈密尔顿回路的基础上对每个格子加上一个状态表示是否为黑格即可。

J

首先我们回顾一下一个事实，就是如果我们每次删除当前树上所有度数为1的点，那么经过$O(T)$轮之后，树上的度数为1的点不会超过$O(n/T)$个，我们记这些点为关键点，那么可以发现，如果一个点往上跳了$O(T)$轮，那么一定在某个关键点之上，而一个关键点之上的点在一条链上，那么最浅点很容易维护。

所以算法大致上是这样的，首先每个点到关键点之前的这一段，我们暴力跳。之后我们对于每个关键点分别最浅的点，等价于区间减，区间最小值。

前面部分需要做$O(nT)$次单点修改，$O(n)$次区间查询，而后面部分需要做$O(n^2/T)$次修改和查询，这里我们假设n, q同阶。前面部分可以用分块将复杂度平衡成为$O(nT) + O(n^{1.5})$，后面部分时间复杂度为$O(n^2 \log n/T)$，所以总的时间复杂度为$O(n\sqrt{n \log n})$。