A:
题目比较长的一个模拟，注意要求模拟出来即可。

B:
状态压缩 dp。
从最简单的想法可以从 N!种排序方式枚举，不过这里的 N 是 20，估计剪枝也剪不过去。
不过可以知道的是，如果我们确定了当前用了哪几个路灯，那么之后添加的路灯不会对前面已经确定的路灯造成影响。这里就满足了状态压缩的 dp 特点，以后的状态不会对之前的造成影响。
状态数是 1<<20 个，枚举每个状态，然后向之后的状态拓展（也就是添加一个没有被添加的路灯），记录每个状态下的远距离即可。

C:
笛卡尔树（裸题）
直接用 treap 的构树方式是会超时的。
其实对 BST 的 key 排序了以后，我们可以通过找当前区间 weight 的最大值，从而决定了根是哪个结点。这里很多种 logN 找最值的方法可以实现。
不过对于笛卡尔树，如果确定了最后添加的结点的位置，可以通过向根的方向不停的寻找第一个比当前插入的 weight 大的第一个结点。
（1）如果搜到根了，根的 weight 也比插入的 weight 小，那么当前整棵树都是插入结点的左子树。
（2）否则，让当前结点的右子树成为插入结点的左子树，然后插入结点作为当前结点的新右子树。
可以很容易证明这是 O(NlogN+N)的算法，开始先对 key 排序，这里是 NlogN。最后插入的结点一定是整一棵树最右的结点，而且最右的树链最长是 N 的长度，每次向上搜的时候，被搜过的点不再是最后的树链上的结点，于是整体复杂度 O(N)。

D:
博弈，Nim 游戏求 SG 值。
《训练指南》P139 的例题 13，因为出题者的大白书早已不复存在，所以忘记有这题了。
书里有详解，我就不写了。

E:
Tarjan 缩点
首先，将添加多少条边转化为求“添加的边数加上原有边数”的最大值是多少。
我们可以知道，尽可能构造出一个完全图，从而得到上述最大值。同时，完全图的最大边数是(N-1)*N。由这个可以知道，我们要的到的图完全图的块数应该尽可能的少，同时，其中一块的 N 要尽可能大。
假设图在一开始的时候不强连通，我们一定能搞到两个连通块（形成完全图），这两个连通块都是其中一块的点指向另一块的点，而且没有回边，这时满足第一个条件，块数尽可能少。要找到这样的两块，于是我们首先可以缩点，生成一个 DAG。然后枚举每一个没有出边或入边的块（甚至可以同时没有），我们最终目标是这两个块不强连通，于是这两个块可以生成的边数是 e1*e2+e1*(e1-1)+e2*(e2-1)，其中 e1*e2 意思为把全部第一块的点指向第二块的点（当然，缩点以后是第一块指向第二块的）。

F:

dp+回溯。

这题主要是题意比较难理解，因为要回溯最小字典序的路径，所以要逆向 dp，同时记录路径。最后回溯一下就可以了。dp 转移方程比较简单。