A.

令 DP[a][b]表示串 s 前 a 位与串 t 前 b 位的最小编辑距离。

当 min(a, b) = 0 时，DP[a][b] = max(a, b)

否则有显然的转移 : DP[a][b] = min(DP[a-1][b], DP[a][b – 1], DP[a-1][b-1] + (s[a] != t[b]))

只有 abs(a-b)不超过 50 的状态是有效的。只需对这些状态计算 DP。可以用数组 f[a][c] = DP[a][a + c – 50]储存，其中 0 <= c <= 100

复杂度 O(50 * min(|S|, |T|))

B.

考虑将 2 的 k 次方用以下结构体储存:

记一个大小为 200 的数组，表示 2^k 的前 200 位。并用高精度储存 k。

容易利用 2^k 和 2^l 计算出 2^(k+l) 对应的结构体。（由于本题仅要求 100 位精度，200 位最后一位存在误差可以接受。）

我们试图求出前 200 位位于区间[L, R]中的最小 k。

考虑如下算法:

维护两个幂次 U，V 表示当前幂次中前 200 位最小的和最大的。初始设置成 2^1,2^3.

维护当前的数 X，初始为 1。任何时候，如果 X*U 的前 200 位不超过 R，则将 X 乘以 U。如果落在区间[L,R]中，则找到了答案。

如果 X*U 超过，我们计算 U*V。容易发现，它的前 200 位要么比 U 小要么比 V 大。如果比 U 小则用它更新 U，比 V 大则用它更新 V。

容易发现，U 的前 200 位是前缀的最小值，V 是前缀的最大值。在此基础上，容易证明算法正确。

复杂度: O(n^3)

C.

考虑用一种颜色染超过 1/3 的灯，使得任何时刻没有两个该颜色相邻。

然后可以去掉这种颜色的所有灯，对余下部分递归处理。

容易发现，倒序贪心染色即可。

复杂度 O(n)