climb

solution 1

考虑朴素的 dp，设 $f_{i,j}$ 表示考虑了 $[1,i]$，其中第 i 个数修改为 j 的最小代价，转移是：
$f_{i,j}=\min_{j-d\le k\le j+d}\{f_{i-1,k}\}+|j-h_i|$，转移区间的左右端点都是单调递增，可以用单调队列优化使得转移做到 $O(1)$，总复杂度就是状态数的 $O(nV)$，无法通过。

转移无法优化了，考虑优化状态数，尝试寻找到一个集合 S，使得最终所有 h_i 的取值都在集合 S 中，这样的话复杂度就能将至 $O(n|S|)$。这个集合也是容易猜到的，每个数取临界状态一定不劣，猜想 $S=\{h_i+xd|i\in[1,n],x\in[-n,n]\}$，证明也是简单的，考虑归纳，考虑最终最小的数 h'_j 的取值，要么和初值相同，要么为 $\min\{h'_{j-1},h'_{j+1}\}-d$，之后对于 $[1,j-1]$ 和 $[j+1,n]$ 使以相同的过程即可。这样的话 $|S|$ 是 $O(n^2)$ 级别，总复杂度为 $O(n^3)$。

solution 2

还是同样的 dp，考虑用更少的信息量记录状态。有个显然的结论， 函数 $f_i(j)$ 是一个段数为 $O(i)$ 级别的下凸函数。证明同样考虑是以归纳法，显然 $f_1(j)$ 是满足条件的，是一个点 $(h_1,0)$。假设前 $i-1$ 满足条件，并且最低点为 (a,b)，那么函数 $F_i(j)=\min_{j-d\le k\le j+d}\{f_{i-1,k}\}$ 就是将 $f_{i-1}(j)$ 在 $j=a$ 左边的部分向左平移 d 单位、在 $j=a$ 右边的部分向右平移 d 单位，这样 $[a-d,b+d]$ 就多出了一段 $y=b$ 水平线，没有破坏下凸函数的性质，然后加上函数 $y=|j-h_i|$，则是将二阶导数的一个点 $+2$，不破坏下凸函数二阶导数非负的性质，因此 $f_i(j)$ 仍是下凸函数，段数增加两段。

这个过程暴力维护即可做到 $O(n^2)$。考虑操作都是区间加和插入，用平衡树维护即可做到 $O(n\log n)$。

---

graph

solution

问题是要求最小化链的长度，根据 dilworth 定理，等价于用最少的反链覆盖整张图。

我们可以 $O(2^n)$ 预处理枚举集合，并 $O(n)$ 判断该点集是否构成反链即互相没有边（偏序关系）相连。然后设 f_S 表示覆盖点集 S 最少需要多少条反链，转移时枚举 S 的子集 T（要求 T 是一条反链），然后 f_S 既可以从 $f_{S/T}$ 转移而来，dp 部分的复杂度为 $O(3^n)$。

总复杂度为 $O(n2^n+3^n)$。

（关于方案的构造，考虑定向完的图是一个 dag，每条反链就是 dag 上的一层，钦定 dag 的方向后每条边的方向就定下了）

---

cycle

solution

首先将存在包含关系的字符串处理掉，只留下不被包含的串（这可以用 kmp $O(n^2|S|)$ 处理），这样一定不劣。

先忽略环，当成序列处理。然后考虑类似单词接龙的过程，因为不存在完全被其它串包含串，重叠部分一定是相邻两个串的首尾相接。用 kmp 即可以预处理出当串 i 在前，串 j 在后，其中 i 是否翻转的状态为 a，j 是否翻转的状态为 b，首尾相接的最长重叠部分是 $mx_{i,j,a,b}$（a,b 都为 $0/1$）。时间复杂度为 $O(n^2|S|)$。

之后就可以考虑 dp 了，设 $f_{S,i,a}$ 表示用掉的字符串集合为 S，末尾的串为 i、翻转状态为 a 的最小长度，转移时枚举下一个串选哪个、是否翻转即可，时间复杂度为 $O(n2^n)$。

然后考虑环如何处理，对于环我们一定可以旋转使得第一个串开头，最后再处理覆盖完后最后一个串和第一个串的首尾相接时的重叠部分即可。

时间复杂度为 $O(n^2|S|+n2^n)$。

（注意 $n=1$ 要特殊处理，答案为 $len_1-border_{1,len_1}$）

---

name

solution

有个比较 naive 的 dp，设 $f_{c,i}$ 表示大写字符 c 可以转化到的字典序最小的长度为 i 的串，$g_{i,j,k}$ 表示第 i 条规则的串 T_i 的前缀 $[1,j]$ 可以转化到的字典序最小的长度为 i 的串。

转移也是简单的，f 与 g 互相转移，其中 $f_{c,l}=\min_{S_i=c}\{g_{i,len_i,l}\}$，$g_{i,j,k}=\min\{g_{i,j-1,d}+f_{T_{i,j},k-d}\}$（要求 $T_{i,j}$ 为大写字母）。

但是仔细一想没那么简单，因为转移是存在环的，需要特殊处理。实际上环的情况是非常单调的，只有当某条规则 T_i 为单独的大写字母时转移才可能有环，其它情况，要么转移到空字符，没有出边，要么长度一直增加，不可能有环。因此我们可以预处理出大写字母的互相转化（用 floyd 在 $o(26^3)$ 内处理出），之后的转移忽略这样的规则即可即可。

最终时间复杂度为 $O(nl^3)$。

（实际上也可以有更粗暴的转移方法，将朴素的 dp 跑若干遍即可，正确性的证明可以考虑转移边和状态实际上构成图，转移的过程就是跑最短路，上述的过程可以看成是 SPFA 算法。这题图的结构特殊，跑 SPFA 的复杂度更接近于 $O(km)$ 而不是 $O(nm)$，其中 k 是小常数)