2 QOJ10094 Slot Machine

2.1 题目大意

有一个显示 k 位数字的机器，其内部有一个未知的 k 位十进制数（可能有前导零）。初始时机器每一位的值都未定义。

现在可以进行若干次操作，一次操作可以改变机器某一位的值为某个 $[0,9]$ 的整数，或向机器进行一次询问。如果机器显示的位上有未定义的位，则这次询问非法。否则机器会告知其显示的 k 位数与内部的 k 位数的大小关系。

向机器询问不需要花费代价，但改变机器的某一位需要花费 1 的代价。求最小的代价，使得能让机器上显示其内部的数。在所有操作之前，可以得知一个长为 10^k 的 01 串 s，其中 $s_i = 1$ 当且仅当机器内部的数可能是 i。

2.2 数据范围

$T \leq 10^4$ 组多测，满足 $k \leq 5, \sum 10^k \leq 10^5$。$s$ 中至少有一个 1。

2.3 解题过程

由于机器会告知我们大小关系，所以在任意时刻，我们能知道的信息必然形如："机器中的数 x 满足 $l \leq x \leq r$"。基于这一信息，我们可以设计一个 dp：$f_{l,r,v}$ 表示当且确定 $l \leq x \leq r$，且机器上显示的数为 v 时，一直到结束最少还需要多少代价。

这个 dp 的转移比较容易：如果 $[l,r]$ 中只有 v 满足 $s_v = 1$，则 $f_{l,r,v} = 0$。否则要么枚举一位并改变它的值，要么进行一次询问。对于前者，会从 $f_{l,r,v'} + 1$ 转移而来，其中 v' 为 v 改变某一位的结果。对于后者，会从 $\max(f_{l,v-1,v}, f_{v+1,r,v})$ 转移而来。

然而这个 dp 的状态数 $O(10^{3k})$，转移复杂度 $O(k)$，复杂度过高，不可接受。考虑只在 dp 时维护每次询问之后的状态，此时必然有 $v = l - 1$ 或 $v = r + 1$，所以只需要记录一个 $t = 0/1$ 表示是哪种情况即可。对于这个新的 dp，转移时需要枚举下一次将机器显示的数变成了什么，看起来复杂度并没有改观。

为了进一步优化，需要进行一些观察。注意到在机器上的数固定的前提下，所知道的信息越多，需要的步数必然越少。下面来证明这一结论：

定理 2.1. *对于任意 $l < r$，都有 $f_{l,r,0} \geq f_{l,r-1,0}$ 且 $f_{l,r,1} \geq f_{l+1,r,1}$。*

证明. 对 $r - l$ 归纳。如果 $[l,r]$ 内只有至多一个 i 满足 $s_i = 1$，则结论显然成立。

由于对称性，我们只需证明 $f_{l,r,0} \geq f_{l,r-1,0}$ 即可。此时考察 $f_{l,r,0}$ 的转移。枚举 v，则 $f_{l,r,0}$ 会从 $\max(f_{l,v-1,1}, f_{v+1,r,0}) + c(l-1, v)$ 转移而来，其中 $c(x,y)$ 表示将 x 变成 y 需要的步数。

考虑使得 $f_{l,r,0}$ 取到最小值的转移 v。此时根据归纳假设，$f_{v+1,r,0} \ge f_{v+1,r-1,0}$，所以在转移 $f_{l,r-1,0}$ 时，v 处转移的每一项都比当且的值更小，故自然有 $f_{l,r-1,0} \le f_{l,r,0}$。

□

除此之外，即使我们采用最直接的二分，也可以做到 $O(k \log 10^k)$ 次操作。因此 dp 的值域应当是很少的。

定理 2.2. 对于任意 l,r,t，都有 $f_{l,r,t} \le k\lceil \log_2(r-l+1)\rceil$。

证明. 在转移时取 $v = \lfloor \frac{l+r}{2} \rfloor$，则 $f_{l,r,t} \le \max(f_{l,v-1,1}, f_{v+1,r,0}) + k$。对 $r-l$ 归纳，即有 $f_{l,r,t} \le k\lceil \log_2 \frac{r-l+1}{2} \rceil + k \le k\lceil \log_2(r-l+1)\rceil$。

□

有了上述结论，我们可以将 dp 的值域记在状态中。具体地，记 $fl_{l,v}$ 表示最大的 r 满足 $f_{l,r,0} \le v$。记 $fr_{r,v}$ 表示最小的 l 满足 $f_{l,r,1} \le v$。则我们需要求出最小的 v 使得 $fl_{0,v} \ge 10^k - 1$。这个 dp 的第二维只需要记录到 $O(k \log 10^k)$，所以状态数为 $O(10^k k \log 10^k)$。

按照 v 升序进行转移。假设已经求得了 $< v$ 的所有 dp 数组信息，考虑对 $v+1$ 计算所有值。由于对称性，下面重点考察 $fl_{l,v}$ 的转移。

考虑一个状态 $(l,r,0)$，如果 $f_{l,r,0} \le v$，则需要存在 $l \le x \le r$，使得 $\max(f_{l,x-1,1}, f_{x+1,r,0}) + c(l-1,x) \le v$。使用 fl, fr 表示，即 $l \ge fr_{x-1,v-c(l-1,x)}$ 且 $r \le fl_{x+1,v-c(l-1,x)}$。因此枚举一个 x 满足 $l \ge fr_{x-1,v-c(l-1,x)}$，则可以从 $fl_{x+1,v-c(l-1,x)}$ 转移到 $fl_{l,v}$。

直接实现这个转移复杂度 $O(10^k)$，总复杂度 $O(10^{2k} k \log 10^k)$，还是太慢了。

注意到如果确定了 $c(l-1,x)$，那么一个 x 可以转移到的 l 是一段前缀，此时可以使用单调队列 [**monotonic_queue**] 进行优化。

然而如果只是枚举 $c(l-1,x) = d$，无法限定转移到的 l 必须满足 $c(l-1,x) = d$，可能会导致不合法的转移。所以我们转而钦定一个位置的集合，表示 $l-1 \to x$ 的过程中这些位被改变了。如果实际并没有改变其中的值，也是没有关系的，因为这样只会是算出的代价更大。

假如枚举 S 中的位，表示这些位可以被改变，则其它位不能变。因此我们枚举所有其它位的可能性，则现在 $c(l-1,x)$ 的值已经被固定，并且我们额外要求 $l-1, x$ 均属于当前我们枚举的这一类数中。有了这一限制，就可以自由转移了，所以枚举 x，使用单调队列可以对所有 l 求出其值。

这样做对于一个固定的 v，需要枚举一个集合 S，在内部还需要枚举 x。故单次转移复杂度 $O(20^k)$，总复杂度 $O(20^k k \log 10^k)$。实际上我们从小到大枚举 v 时，一旦遇到一个 v 已经合法，就不用再转移了。最坏情况下（即 $k = 5$，且每一个数都有可能出现）这个最大的 v 为 $V = 28$，所以实际复杂度为 $O(20^k V)$，可以通过。