Solution

商贸(path)

简单DP题，可以记$f_{i,j}$表示第i天到达j的最大收益。模拟一天的转移需要$O(M)$，由于每一天的时间代价是$O(C)$增长的，至多$\max\{m_i\}$天之后就不会有正收益了。

图(graph)

考虑扫描线，按照 x 升序排序后扫描，维护当前每个联通块，容易发现，能否加入联通块取决于目前该联通块中 y 坐标最小的点。

容易发现这个构成一个 y 从栈底到栈顶单调递减的单调栈。如果加入一个新的二元组 (u,v)，栈顶的 y 大于 v，那么 (u,v) 只能属于一个新的联通块；

否则， 可以把栈中所有小于等于 v 的元素所属的联通块和 (u,v) 合并到一起，弹栈后留下最小元素即可。

牛奶香浓 (milky)

设 $f(i,j)$ 表示到了位置 i ，匹配到了 `milky` 的第 j 位的方案数，转移可以直接使用矩阵。

区间问题可以直接套一个线段树 / 其他数据结构，这样可以做到 $O((n+q)B^3\log n), B=6$，不一定可以过。

考虑优化，我们可以使用前缀和，答案就是 $s_r \times inv(s_{l-1})$，其中 $inv()$ 表示矩阵求逆。

但是直接求逆要求逆元，在 $\mod 2^{32}$ 下非常困难。 注意到转移矩阵非常简单，就像这样：

$$\begin{bmatrix} 1 & 1 & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

它的逆就是：

$$\begin{bmatrix} 1 & -1 & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

直接前缀和处理出求逆的结果即可。

偷食(easiest)

考虑操作 1 怎么做。首先 $[l_1, l_2)$ 这段区间一定能取完。那么只需要对 $(r_2, r_1]$ 这段区间求解即可。

发现限制每个点被取的点的集合是这个点前的一段连续区间。一个数要是可以被取，就要把它到它上一个比它大的数中的所有数取完。这个区间用单调栈即可求出。发现所有区间要么不相交要么包含，构成了树的结构。

暴力枚举有多少个点的限制区间与询问区间 $[l_2, r_2]$ 有交，复杂度 $\mathcal{O}(n^2)$。

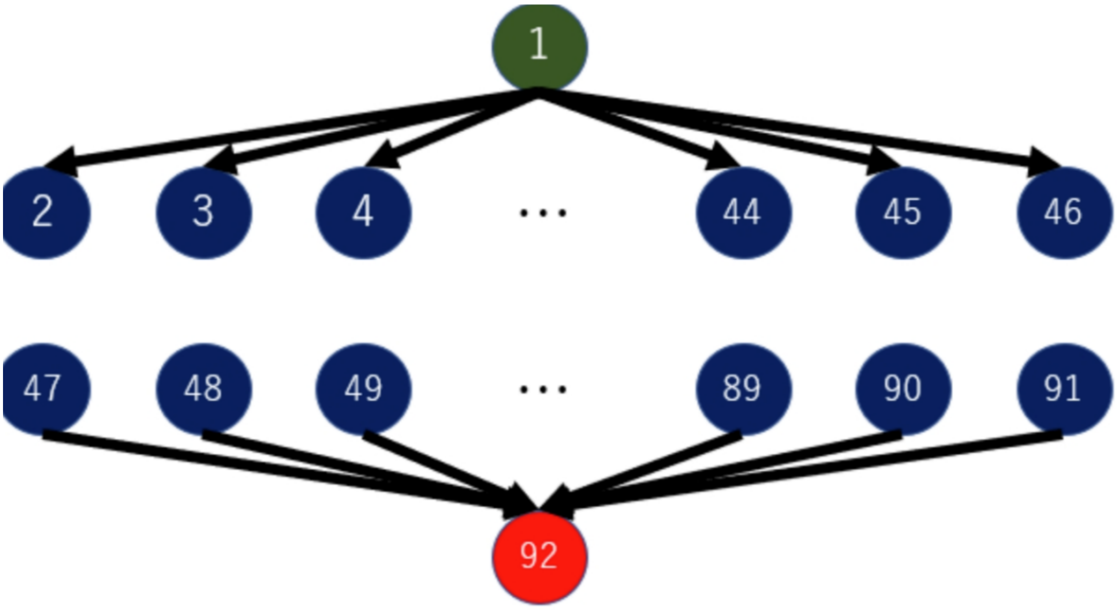
也可以从 r_2 开始做树上倍增，直到比 r_1 大，跳到的点都和 $[l_2, r_2]$ 有交。复杂度 $\mathcal{O}(n \log n)$ 。

对于操作 2 ，对于点对 (u, v) ，如果不合法，则需要满足 $a_u < a_v$ 、且不存在 $k \in (u, v)$ 满足 $a_k > a_v$ 且至少存在一个 $p \in (u, v)$ 满足 $a_u < a_p < a_v$ 。

这相当于 u 需要在 v 的限制区间内，即 v 是 u 的祖先。不妨令 p 取到最小的满足条件的位置，那么上述不合法的 (u, v) 满足：u 是 v 的非父亲祖先或者 v 是 u 的非父亲祖先。直接预处理即可，同样可以使用倍增的思想，复杂度 $\mathcal{O}(n \log n)$

歧路万千 (path)

解法 1

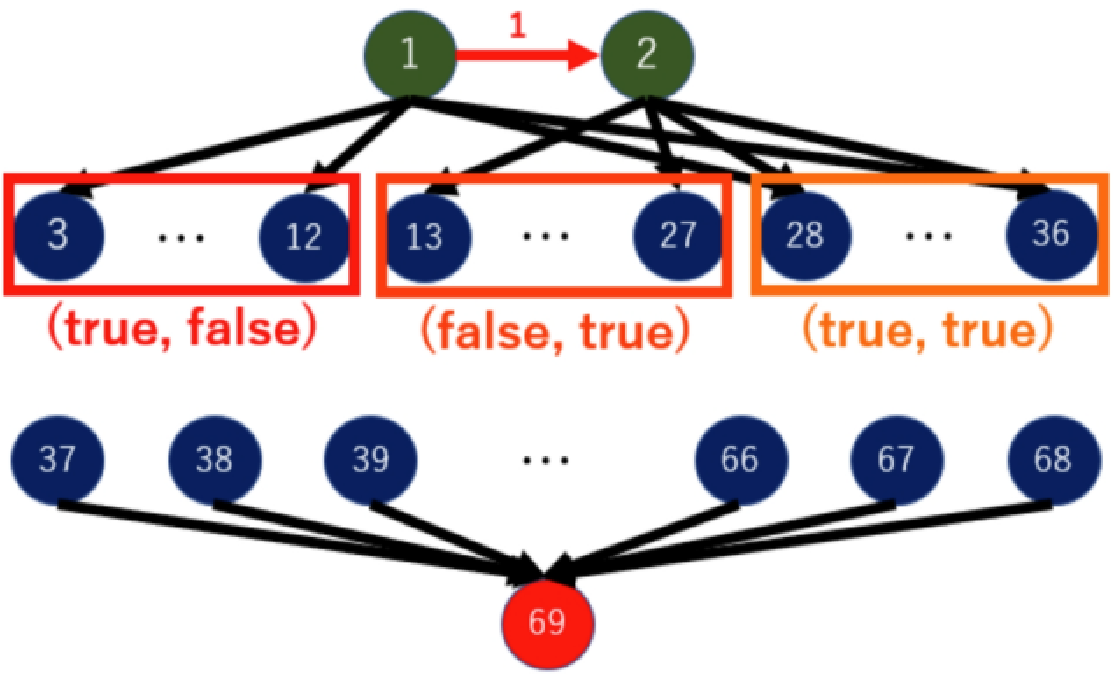


如图所示，中间至多$45^2=2025$条边，出现了就加一条即可。

期望得分 10。

解法 2

从解法 1 的图开始改，考虑将中间的边缩至 1000 条。具体来说这么搞：



将两个相邻的距离绑在一起 $(1,2),(3,4)\dots$。每一组相邻的距离是否出现可以用一个 2 位二进制数表示。共四种情况，其中两个数字都不出现的情况无需考虑，因此只有三种情况。

按照上图进行构造，对于每一对权值 $(2i-1,2i)$。根据在 P 数组种出现的情况进行划分，如果对应该类上一个点到下一层的所有点距离都已经使用了，就新建一个点 x，否则使用该类点中到下一层还有边没用的点 x。然后从 x 连向下一层一条权值为 $2i-1$ 的边即可。

例如：对于数对 $(99, 100)$，假设两个距离都需要出现。首先查询第二层三类点中有没有和下一层的点还没有边相连的。如果有，就使用那个点，否则就在第二层新建一个点 x。从 1 向 x 连边，距离为0，从 2 向 x 连边，距离为 0。从 x 向下一层一个未与 x 连边的点，距离为 99。

最后至多需要：$2 + \lfloor \frac{1000}{32} \rfloor + (3-1) + 32 + 1 = 69$ 个点。期望得分 30 分。

解法 3

按二分组不够优秀，我们可以考虑更换分组方式。

分组方式	顶点数
2×1000	$2 + 32 + (3-1) + 32 + 1 = 69$
3×667	$3 + 26 + (7-1) + 26 + 1 = 62$
4×500	$4 + 22 + (15-1) + 23 + 1 = 65$
5×400	$5 + 20 + (31-1) + 20 + 1 = 76$

期望得分 52。

解法 4

接下来我们优化 4×500 的方案，我们注意到 $(15-1)$ 这个非常耀眼，这是这个方案非常累赘的地方，如果优化好了可能可以超过 3×667。

我们注意到 $(0, 1, 1, 1)$ 等以 0 开头的分组是完全没有必要的，完全可以往后推到 $(1, 1, 1, *)$ 这样考虑，因此有效的分组结果成为了 8 种。

优化到了 $4 + 22 + (8-1) + 23 + 1 = 57$，期望得分 76 分。

解法 5

第 57 号节点不是必要的，所有下层节点全部往最后一个下层节点连边即可，无需新建节点，优 化到了 56 个节点，期望得分 82 分。

解法 6

我们在最上层部分连了 $1 \to 2 \to 3 \to 4$，边权都为 1，接下来尝试将这 4 个节点压缩到只剩下一个节点。

我们假设所有的 8 种类型都是存在的，那么一定存在 1001，1010 ，1100 ，对应的上层节点， 我们让 1 分别给这些点连上边权为 $3, 2, 1$ 的边，剩下的状态通过包含这些节点的方式（如 1111 包含了 1001 ，1010 ，1100，让这些点去向 1111 连接长度为 0 的边）整出来，这样上面的 4 个节点只剩下 1 个了。

优化到了 53 个节点，期望得分 100 分。