C. 月

定义 $n = x_{max}$。

算法一

考虑每次暴力 dp，由于只需要枚举不超过 $\min\{\sqrt{x}, w\}$ 个转移点，所以复杂度 $O(qn\min\{\sqrt{n}, w\})$，期望得分 35 分。

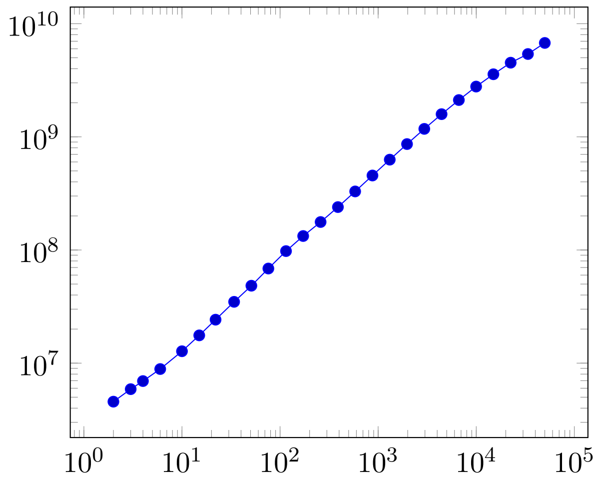
算法二

考虑这是一个 DAG 最短路模型，而答案不会很大，所以可以暴力地更新，也就是修改了一个位置的代价，然后暴力修改所有其能到达位置的答案，由于每次至少变少 1，所以每个位置不会被访问超过初始值次。

设 $p(x)$ 代表 x 到 0 的初始最小代价。

所以直接暴力每次 dfs 地更新，每次枚举除掉的数，可以更新到最多 $\min\{n/x, w\}^2$ 个数。

那么总次数就是 $\sum_x p(x)\min\{n/x, w\}^2$，可以编程计算这个值，下图是其关于 w 的函数（$n = 5\times 10^4$ 时）。



至于这东西的渐进复杂度分析，由于 w 比较大的时候很容易每次走一个质数，这东西就是 $\log_w x$ 量级的，然后求和就是：

$\frac{1}{\log w}\sum_x \log x \min\{n/x, w\}^2$

$= O\left(\log_w n\left(\sum_{i=1}^{n/w} w^2 + \sum_{i=n/w}^{n}(n/i)^2\right)\right)$

前面的直接算，后面的积分，$= O\left(\log_w n\left(nw + nw\right)\right)$

$= O(nw\log_w n)$。

而只有前几个 w 取到的因子个数较多，影响不大。

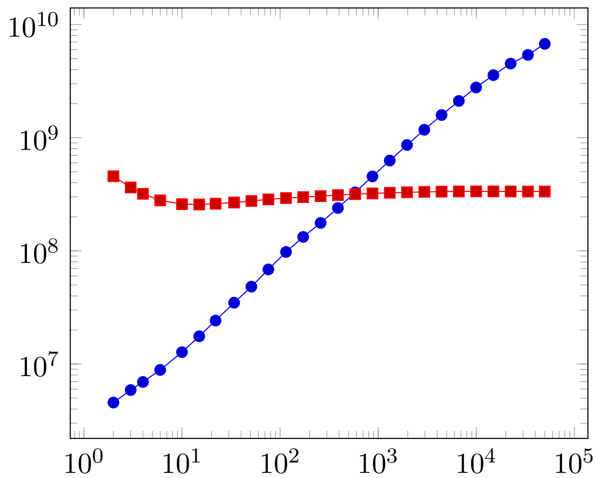
还有一个问题是如何求初始值，可以使用各种算法，如 $O(n\sqrt{n})$ 的暴力等，均不是瓶颈。

注意暴力更新的最内侧循环只有一个判断，所以实际上跑的极其快，期望得分 80~100 分。

算法三

考虑每次不暴力更新，而是枚举除掉的数是 x，那么能转移到的是一个区间，可以使用线段树维护区间最大值和最大值点来维护，考虑这样做的复杂度。

由于每次枚举除掉的数肯定不超过 n/x，所以最多 $\sum p(x)\min\{w, n/x\}$ 次线段树操作，如果把线段树的 $\log n$ 设为 100 的话，下图是两种算法的效率。



出题人试图分析了一下渐进复杂度，如下：

$\sum_{i=1}^{n} p(i)n/i \le n\sum_{i=1}^{n} \frac{1}{i}\sum_j d_0\left(\left\lfloor \frac{i}{w^j}\right\rfloor\right)$

$= O\left(n\sum_t d_0(t)\sum_{k=1}^{\log_w(n/t)}\sum_{p=w^k t}^{w^k(t+1)-1} 1/p\right)$

$= O\left(n\sum_t d_0(t)\sum_{k=1}^{\log_w(n/t)}\log\left(\frac{w^k(t+1)}{w^k t}\right)\right)$

$= O\left(n\sum_t d_0(t)\sum_{k=1}^{\log_w(n/t)}\log\left(\frac{t+1}{t}\right)\right)$

$= O\left(n\sum_t d_0(t)\log_w(n/t)\log(1+1/t)\right)$

$= O\left(n\log_w n\sum_i\sum_{j=1}^{n/i}\log(1+1/ij))\right)$

由于 $\log(1+1/x) = O(1/x)$，则

$= O(n\log_w n\sum_i 1/i\sum_{j=1}^{n/i} 1/j)) = O(n\log^3 n/\log w)$

次线段树操作，总复杂度不超过 $O(n\log^4 n/\log w)$，但似乎十分不满。

发现可以在时间限制内通过，期望得分 100 分。

注意线段树在 w 很小的时候表现比其他地方慢两倍，可以选择在适当的数据范围下分别使用两种算法的较快者，加快速度，标算并没有使用这个优化。

如果有能够严谨分析出这两个算法的渐进复杂度的欢迎洛谷联系 dengyao-triangle 赐教。