题目分析

孙耀峰

2018年x月x日

ZYB建围墙

ZYB和售货机

ZYB玩字符串

题目大意

- 求在六边形网格里围住至少 N 个格子的最短围墙长度。

20分

- 结合样例和图示猜一猜。
- 效率：$O(1)$。

40分

- 推知一些可能的规律，继续手算。
- 效率：$O(1)$。

第80%的数据

- 观察这个等式，你会发现：
  房子数量恰好能填充满一个正六边形。
- 我们不加证明地断言：此时正六边形外面围一圈必然最优。
- 直接计算一下边界的长度即可。
- 效率：$O(\log\log N)$ （开根复杂度）

100分

- 把上述思路拓展一下即可。
- 假设我们已经围住了一个满的六边形。
- 如果要再多围一个点，怎么办?
- 画一画就会发现，我们只需多围长度为 1 的围墙，就可以把一条边“扩展”一层。
- 再画一下，假设目前六边形的边长是e，连续“扩展” 6 次，每次能多围的增量分别是 $e-1, e, e, e, e, e+1$。而连续增量 6 次后，正好又是下一个边长六边形了。
- 我们只需暴力枚举六边形的边长，等到接近 N 时，再一层一层贪心拓展，找到刚刚 $\geq N$ 时的答案即可。
- 时间效率是 $O(\sqrt{N})$。

ZYB建围墙

ZYB和售货机

ZYB玩字符串

题目大意

- 一台售货机有 N 个物品。若编号为 i 的物品至少有 1 个，可以花费 c_i 的代价，购买 1 个编号为 p_i 的物品。再给出每个物品的个数 a_i 和市场价 d_i，问一番捣鼓后能挣多少钱。
- $N \le 10^5$，$1 \le a_i, f_i, c_i, d_i \le 10^6$

10分

- 怎么暴力怎么来。
- 比如直接枚举按钮 i 按几次。
- check一下合法性更新一下答案即可。
- 复杂度 $O(\prod A_i \times ?)$

30分

- 注意到，每一个按钮 i 只可能有三种状态：
  1. 买 a_i 次。
  2. 买到只剩下 1 个（使 f_i 能被选到）。
  3. 一次也不买。
- 所以三进制枚举一下即可。
- 复杂度 $O(\prod 3^N \times ?)$

第60%

- $f_i = i$。
- 这个点是送分的。
- 直接统计一下答案即可。
- 复杂度 $O(N)$

第70%

- $f_i \le i$。
- 如果我们从 i 到 f_i 连边。
- 这是一个由根是自环的树组成的森林。
- 考虑一条有向边 (x, y) 的意义：
- 如果物品 x 还存在，那么物品 y 可以被选，且收益给定。

第70%

- $f_i \le i$。
- 如果我们从 i 到 f_i 连边。
- 这是一个由根是自环的树组成的森林。
- 考虑一条有向边 (x, y) 的意义：
- 如果物品 x 还存在，那么物品 y 可以被选，且收益给定。
- 而且**这棵树的边全是朝向根的方向！！！**

第70%

- $f_i \leq i$。
- 如果我们从 i 到 f_i 连边。
- 这是一个由根是自环的树组成的森林。
- 考虑一条有向边 (x, y) 的意义：
- 如果物品 x 还存在，那么物品 y 可以被选，且收益给定。
- **而且这棵树的边全是朝向根的方向！！！**
- 如果我们从根开始往子树考虑，你会发现：
  每一个物品其实都能被取空。
- 直接统计一下答案即可。
- 复杂度 $O(N)$

100%

- 一般情况下，我们连成的有向图是**基环内向树**。

100%

- 一般情况下，我们连成的有向图是**基环内向树**。
- 现在我们考虑这个环。
- 如果存在环上的一个点 i，按下这个按钮后，获得的收益（c_i）不如另外一条连向 f_i 的树边，那么我们可以直接把这条环边断开。
- 这样就化成了上述树的做法，统计一遍即可。

100%

- 一般情况下，我们连成的有向图是**基环内向树**。
- 现在我们考虑这个环。
- 如果存在环上的一个点 i，按下这个按钮后，获得的收益（c_i）不如另外一条连向 f_i 的树边，那么我们可以直接把这条环边断开。
- 这样就化成了上述树的做法，统计一遍即可。
- 否则，环上的边一定都是最大的。
- 我们先把环上的物品先都取得只剩下 1 个。
  注意，此时一定有一个物品会取不到环边的价值。
- 那就找一个环边价值减去树边价值最小的点 i，强制它不能选环边。接下来统计一遍树的情况即可。
- 复杂度 $O(N)$

ZYB建围墙

ZYB和售货机

ZYB玩字符串

题目大意

- 初始 S 为空。每次往 S 里任意位置插入一个连续 p。
- 给出若干次操作后的 S，求可能的长度最小、字典序最小的 p。
- $|S| \leq 200$

20分

- 怎么暴力怎么来。
- 复杂度$O(?)$

40分

- 注意到，p必然是S的一个子串（考虑最后一次操作）。
- 首先，我们从小到大枚举 p 的长度 len。
- 在 $O(N)$ 枚举可能的 p。
- 验证的话，由于 $|S| \leq 20$，可以采取暴力验证。
- 每次找到一个连续的 p，暴力删掉，继续递归下去即可。
- 复杂度上界 $O(2^N \times N)$。

第70%

- 有一种很自然的想法是:
- 是不是可以贪心地去验证一个 p?
- 比如我们每次删除最开头的一个匹配的 p,并继续这个过程,直到 S 为空。
- 很可惜,这是错的,比如 $S = ababaa, p = aba$。
- 其实也是合法的,但第一步我们要删中间的 aba。
- 但是这一档数据是随机的,所以应该可以过。
- 复杂度 $\sum_{len|N} N^2$。

第80%

- 答案长度为 1 直接判断即可。
- 如果答案长度为 2，字符集必然是 2，相当于我们要验证 ab 的分解方式是否合法。
- 可以dp下，设$f_{i,j}$表示i j是否能消光，随意转一下即可。
- 否则，因为保证有解，长度必然为 3。根据字符出现次数，可以确定 a 和 b 的数量，大概判一下合法性即可。
- 复杂度$O(?)$

100%

- 注意到，虽然一块 p 可能被断成“若干截”，但夹在其中的每一段，必然是能独立消光的。
- 我们可以 $f_{i,j}$ 表示区间 $[i,j]$ 是否能合法。
- 如果 $j-i+1$ 是 len 的倍数，$f_{i,j}$ 就表示它能否消光。

100%

- 注意到，虽然一块 p 可能被断成“若干截”，但夹在其中的每一段，必然是能独立消光的。
- 我们可以 $f_{i,j}$ 表示区间 $[i,j]$ 是否能合法。
- 如果 $j-i+1$ 是 len 的倍数，$f_{i,j}$ 就表示它能否消光。
- 注意， $j-i+1$ 不要求是 len 的倍数。
- 如果有余数，我们要求多余的部分正好是 p 的前缀（这样 f 值才是1）。

100%

- 转移时考虑第 j 个字符。
  1. 它和之后的零碎字符会拼成一段，即 $f_{i,j}| = f_{i,j-1}\&a_j = prefix_{(j-i)\%len+1}$；
  2. 对于之后的零碎部分而言， j 是属于夹在它之间的整块：$f_{i,j}| = f_{i,j-k\times len}\&f_{j-k\times en+1,j}$。
- 第二步的转移复杂度上界是 $O(\frac{N}{len})$的。
- 最终复杂度约为 $\sum_{len|N}(N-len+1)\times N^2\times\frac{N}{len}=O(N^4)$。
- 看起来复杂度很满。
- 其实我们对每一个 p，可以先判断一下字符集合法性。
- 然后再对 p 记忆化一下。就跑得很快啦。