基础算法选讲-解题报告

徐沐杰

南京大学

2023 年 7 月 9 日

取舍

- 10pts 因为提供的精力值总是大于等于零，于是贪心选取所有正金钱价值的物品。

取舍

- 10pts 因为提供的精力值总是大于等于零，于是贪心选取所有正金钱价值的物品。
- 30pts 因为 n 很小，直接搜索所有情况即可。

取舍

- 10pts 因为提供的精力值总是大于等于零，于是贪心选取所有正金钱价值的物品。
- 30pts 因为 n 很小，直接搜索所有情况即可。
- 100pts 考虑折半搜索。各枚举出一半物品的所有情况，把每种情况得到的精力记录下来。

取舍

- 10pts 因为提供的精力值总是大于等于零，于是贪心选取所有正金钱价值的物品。
- 30pts 因为 n 很小，直接搜索所有情况即可。
- 100pts 考虑折半搜索。各枚举出一半物品的所有情况，把每种情况得到的精力记录下来。
  考虑组合这两半物品的情况，需要满足 $W_1 + W_2 \geq -W$，对两半情况的某一半做一次关于 W 的后缀最大值，对于另一半逐个二分/单调指针查找即可。

取舍

- 10pts 因为提供的精力值总是大于等于零，于是贪心选取所有正金钱价值的物品。
- 30pts 因为 n 很小，直接搜索所有情况即可。
- 100pts 考虑折半搜索。各枚举出一半物品的所有情况，把每种情况得到的精力记录下来。
  考虑组合这两半物品的情况，需要满足 $W_1 + W_2 \geq -W$，对两半情况的某一半做一次关于 W 的后缀最大值，对于另一半逐个二分/单调指针查找即可。
  最优复杂度 $O(2^{\frac{n}{2}})$。

建筑

- 按列（即有 k 列）考虑最终的长方形。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。
- 于是可以把每种建筑材料想象为一条长为 u_i ，可以按整数任意分割的木板。如果最终的答案是 d，考虑构造一个可行解。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。
- 于是可以把每种建筑材料想象为一条长为 u_i，可以按整数任意分割的木板。如果最终的答案是 d，考虑构造一个可行解。
- 在这种情境下，相当于用木板按序填满有 k 个长度为 d 的空槽。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。
- 于是可以把每种建筑材料想象为一条长为 u_i，可以按整数任意分割的木板。如果最终的答案是 d，考虑构造一个可行解。
- 在这种情境下，相当于用木板按序填满有 k 个长度为 d 的空槽。
- 对于长大于等于 d 的木板，只有其中长度为 d 的部分可用。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。
- 于是可以把每种建筑材料想象为一条长为 u_i，可以按整数任意分割的木板。如果最终的答案是 d，考虑构造一个可行解。
- 在这种情境下，相当于用木板按序填满有 k 个长度为 d 的空槽。
- 对于长大于等于 d 的木板，只有其中长度为 d 的部分可用。
- 现在相当于只有长小于等于 d 的木板。按序考虑空槽，如果放得下就放，如果放不下就割下多出的部分放在新的空槽里即可。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。
- 于是可以把每种建筑材料想象为一条长为 u_i，可以按整数任意分割的木板。如果最终的答案是 d，考虑构造一个可行解。
- 在这种情境下，相当于用木板按序填满有 k 个长度为 d 的空槽。
- 对于长大于等于 d 的木板，只有其中长度为 d 的部分可用。
- 现在相当于只有长小于等于 d 的木板。按序考虑空槽，如果放得下就放，如果放不下就割下多出的部分放在新的空槽里即可。
- 可以证明一定不会产生冲突。因为木板的长度不超过 d，所以一定不会有一个行截面出现重复的材料。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。
- 于是可以把每种建筑材料想象为一条长为 u_i，可以按整数任意分割的木板。如果最终的答案是 d，考虑构造一个可行解。
- 在这种情境下，相当于用木板按序填满有 k 个长度为 d 的空槽。
- 对于长大于等于 d 的木板，只有其中长度为 d 的部分可用。
- 现在相当于只有长小于等于 d 的木板。按序考虑空槽，如果放得下就放，如果放不下就割下多出的部分放在新的空槽里即可。
- 可以证明一定不会产生冲突。因为木板的长度不超过 d，所以一定不会有一个行截面出现重复的材料。
- 于是检查是否有 $\sum_{i\in[n]} \min\{d, u_i\} \geq kd$ 即可。

建筑

- 按列（即有 k 列）考虑最终的长方形。
- 我们需要让每列的最短长度最长，且列间对应位置的材料种类不同。
- 于是可以把每种建筑材料想象为一条长为 u_i，可以按整数任意分割的木板。如果最终的答案是 d，考虑构造一个可行解。
- 在这种情境下，相当于用木板按序填满有 k 个长度为 d 的空槽。
- 对于长大于等于 d 的木板，只有其中长度为 d 的部分可用。
- 现在相当于只有长小于等于 d 的木板。按序考虑空槽，如果放得下就放，如果放不下就割下多出的部分放在新的空槽里即可。
- 可以证明一定不会产生冲突。因为木板的长度不超过 d，所以一定不会有一个行截面出现重复的材料。
- 于是检查是否有 $\sum_{i\in[n]} \min\{d, u_i\} \geq kd$ 即可。
- 利用单调性进行二分答案，可得复杂度 $O(n \log \max\{u_i\})$。

RMQ 问题

- 15pts 对于每个询问，直接计算区间的最小值即可，复杂度 $O(nq)$。

RMQ 问题

- 15pts 对于每个询问，直接计算区间的最小值即可，复杂度 $O(nq)$。
- 50pts ST 表模板题，复杂度 $O(n \log n + q)$。

RMQ 问题

- 15pts 对于每个询问，直接计算区间的最小值即可，复杂度 $O(nq)$。
- 50pts ST 表模板题，复杂度 $O(n\log n+q)$。
- 25pts(bonus) 因为询问的区间必定大于序列长度的一半，所以询问肯定跨过区间的中点。所以在区间中点前做后缀最大值，在区间中点后做前缀最大值即可，复杂度 $O(n+q)$。

RMQ 问题

- 15pts 对于每个询问，直接计算区间的最小值即可，复杂度 $O(nq)$。
- 50pts ST 表模板题，复杂度 $O(n\log n+q)$。
- 25pts(bonus) 因为询问的区间必定大于序列长度的一半，所以询问肯定跨过区间的中点。所以在区间中点前做后缀最大值，在区间中点后做前缀最大值即可，复杂度 $O(n+q)$。
- 25pts(bonus) 由于数据范围小有很多种做法。假设不会 ST 表，可以把每个位置前的每种数的最迟出现位置做一次关于数大小的前缀最大值，然后每次询问需要二分查找最大值，复杂度 $O(n\max(|a_i|)+q\log\max(|a_i|))$。

RMQ 问题

- 15pts 对于每个询问，直接计算区间的最小值即可，复杂度 $O(nq)$。
- 50pts ST 表模板题，复杂度 $O(n\log n+q)$。
- 25pts(bonus) 因为询问的区间必定大于序列长度的一半，所以询问肯定跨过区间的中点。所以在区间中点前做后缀最大值，在区间中点后做前缀最大值即可，复杂度 $O(n+q)$。
- 25pts(bonus) 由于数据范围小有很多种做法。假设不会 ST 表，可以把每个位置前的每种数的最迟出现位置做一次关于数大小的前缀最大值，然后每次询问需要二分查找最大值，复杂度 $O(n\max(|a_i|)+q\log\max(|a_i|))$。
  这实际上是在模拟明天要学的单调栈。很容易发现此前缀最大值只需要维护一些“突变点”就可以让复杂度降到均摊 $O(n+q\log n)$，这个做法聊作启发，明天大家将更深入学习。

滑窗

- 麻烦：gcd 可以结合，但是无法消去。

滑窗

- 麻烦：gcd 可以结合，但是无法消去。
- 受 RMQ 问题的前后缀最大值方法的启发，可以在第一个区间中选取任意一点（为了方便，可选择右端点）为“分界点”。

滑窗

- 麻烦：gcd 可以结合，但是无法消去。
- 受 RMQ 问题的前后缀最大值方法的启发，可以在第一个区间中选取任意一点（为了方便，可选择右端点）为"分界点"。
- 我们从该点往左一直做后缀 gcd 到区间的左端点。同时维护从交界点到右端点的 gcd，每次的答案就是这两个 gcd 的 gcd。

滑窗

- 麻烦：gcd 可以结合，但是无法消去。
- 受 RMQ 问题的前后缀最大值方法的启发，可以在第一个区间中选取任意一点（为了方便，可选择右端点）为“分界点”。
- 我们从该点往左一直做后缀 gcd 到区间的左端点。同时维护从交界点到右端点的 gcd，每次的答案就是这两个 gcd 的 gcd。
- 此后若右端点需要右移，只需要更新一次 gcd。

滑窗

- 麻烦：gcd 可以结合，但是无法消去。
- 受 RMQ 问题的前后缀最大值方法的启发，可以在第一个区间中选取任意一点（为了方便，可选择右端点）为“分界点”。
- 我们从该点往左一直做后缀 gcd 到区间的左端点。同时维护从交界点到右端点的 gcd，每次的答案就是这两个 gcd 的 gcd。
- 此后若右端点需要右移，只需要更新一次 gcd。
- 若左端点需要右移，如果没超过交界点，只需丢弃一个后缀和。

滑窗

- 麻烦：gcd 可以结合，但是无法消去。
- 受 RMQ 问题的前后缀最大值方法的启发，可以在第一个区间中选取任意一点（为了方便，可选择右端点）为“分界点”。
- 我们从该点往左一直做后缀 gcd 到区间的左端点。同时维护从交界点到右端点的 gcd，每次的答案就是这两个 gcd 的 gcd。
- 此后若右端点需要右移，只需要更新一次 gcd。
- 若左端点需要右移，如果没超过交界点，只需丢弃一个后缀和。
- 如果超过了交界点，可以将交界点直接移到当前区间的右端点，并重新计算后缀和。可以证明每个数只会参与一次后缀和计算，复杂度 $O(n\log\max\{a_i\}+q\log\max\{a_i\})$。

滑窗

- 遐想：ST 表可不可以直接做?

滑窗

- 遐想：ST 表可不可以直接做?
- 倍增计算 gcd，由于固定一个端点时 gcd 迭代次数有限，每个点均摊复杂度上界是 $O(\max\{\log\max\{a_i\}, \log n\})$，那么复杂度也是 $O(n\log\max\{a_i\} + q\log\max\{a_i\})$。

滑窗

- 遐想：ST 表可不可以直接做?
- 倍增计算 gcd，由于固定一个端点时 gcd 迭代次数有限，每个点均摊复杂度上界是 $O(\max\{\log\max\{a_i\},\log n\})$，那么复杂度也是 $O(n\log\max\{a_i\}+q\log\max\{a_i\})$。
- 甚至不需要滑窗的性质!

滑窗

- 遐想：ST 表可不可以直接做?
- 倍增计算 gcd，由于固定一个端点时 gcd 迭代次数有限，每个点均摊复杂度上界是 $O(\max\{\log\max\{a_i\}, \log n\})$，那么复杂度也是 $O(n\log\max\{a_i\} + q\log\max\{a_i\})$。
- 甚至不需要滑窗的性质!
- 但是面对矩阵乘法等运算时失效。

滑窗

- 遐想：ST 表可不可以直接做?
- 倍增计算 gcd，由于固定一个端点时 gcd 迭代次数有限，每个点均摊复杂度上界是 $O(\max\{\log\max\{a_i\},\log n\})$，那么复杂度也是 $O(n\log\max\{a_i\}+q\log\max\{a_i\})$。
- 甚至不需要滑窗的性质!
- 但是面对矩阵乘法等运算时失效。
- Anyway，数论算法超纲了，仅作为可选解法。
  ~~别想了，我卡了你们内存……~~

提问 || 完结

Any Question?

Any Question?

Thank you for listening!