题目大意

有 n 个购买计划，第 i 个计划形如：1. 花费 a_i 的价钱购买一个物品；2. 花费 b_i 的价钱购买两个物品。两个选项不能同时选择，每个计划最多能执行一次。给定 q 次操作：

- `1 p x y`:$a_p \leftarrow x, b_p \leftarrow y$
- `2 l r k`:只能执行 $i \in [l, r]$ 中的计划，希望购买 k 个物品，花费的最少的价钱是多少

数据范围

$1 \le n, q \le 1 \times 10^5, 1 \le a_i < b_i \le 10^9, 0 \le l \le r < n, 1 \le k \le 2(r-l+1)$

子任务编号	特殊性质	分值
1	$1 \le n, q \le 14$	5
2	$1 \le n, q \le 300$	5
3	$1 \le n, q \le 2 \times 10^3$	15
4	$1 \le n, q \le 1 \times 10^5, l = 0$，保证没有 1 操作	15
5	保证没有 1 操作	10
6	$1 \le n, q \le 8 \times 10^4$	20
7	无	30

时间限制：$3s$

空间限制：$512MB$

解题过程

Subtask 1

对每个询问暴力枚举每个计划的选取状态，检查是否购买了 k 个更新答案。 时间复杂度 $O(q3^n)$.

Subtask 2

对每个询问进行一个暴力的背包 dp。时间复杂度 $O(qnk)$.

Subtask 3

考虑一个按照性价比购买的贪心。

对于第 i 个计划：

- $a \le b - a$

买第二个物品的时候更昂贵，是否买第二个物品和是否买第一个物品的决策独立。把它拆成两个大小为 1 的独立物品，代价分别为 $a, b-a$，性价比分别为 $a, b-a$.

- $a > b - a$

买第二个物品的时候更便宜，在最优性价比的购买策略中，假设购买了第一个物品，之后如果继续购买，那么一定会紧接着购买第二个物品。把它看成一个大小为 2 的组合物品，代价为 b，性价比为 $\frac{b}{2}$.

按照性价比排序购买，如果刚好能凑齐 k 个物品那么就是答案，否则能凑齐 $k+1$ 个物品，最后一个是大小为 2 的组合物品 s.

考虑已选集合和最优策略集合的差异，一定不会删除一个大小为 v 的物品去加入另外一个大小为 v 的物品。所以有 2 种策略：

- 从已选集合删除最贵的大小为 2 的组合物品，在未选集合中加入最便宜的大小为 1 的物品。
- 从已选集合删除最贵的大小为 1 的物品。

对于大小为 1 的物品，可以通过将组合物品拆分得到，即在选取组合物品的基础上减去 $b-a$.

对每组询问暴力排序之后贪心，时间复杂度 $O(qn\log n)$.

Subtask 4

Subtask 3 中贪心需要维护一个前缀 $\max$ 和后缀 $\min$，由于没有删除操作，可以在树状数组上维护。

查询时需要先在树状数组上二分出已选集合和未选集合的分界线，然后分类讨论即可。

时间复杂度 $O(n\log n)$.

Subtask 5

套用 Sub4 的做法，使用可持久化线段树维护。时间复杂度 $O(n\log n)$.

Subtask 6

大常数正解或者一些根号做法。

Subtask 7

考虑对性价比值域进行整体二分，假设当前的值域区间为 $[l,r]$，维护当前所有答案在 $[l,r]$ 内和加入/删除一个物品的性价比在 $[l,r]$ 的操作序列。每一层用线段树维护出值域在 $[l,mid]$ 的前缀 $\max$，值域在 $(mid,r]$ 的后缀 $\min$ 和值域在 $[l,mid]$ 的 sum/cnt，类似 Sub3 的贪心将询问分成两部分递归做就可以了。

因为带删除，所以在线段树每个节点要维护 $\max$ / $\min$ 维护两个可删堆。

考虑到常数问题，因为只涉及单点修改，可以将线段树改成非递归形式。

时间复杂度 $O(n\log^2 n)$.

参考资料

与李劲逸、宋浩然、鲜博宇同学的讨论。