联测缺首（study）

如果上课不如自学，那么把所有上课时间换成自学可行且更优，所以将所有 A_i 变为 $\max(A_i, B_i)$。之后的问题里，保证 $A_i \geq B_i$。

要求 n 门课的最小理解程度最大，这是有单调性的，所以考虑二分答案 x，问题转化为如何判定是否能达到每门课的理解程度都 $\geq x$。

对于每门课，由于 $A_i \geq B_i$，优先考虑上课。第 i 门课所需的上课课时为 $C_i = \lceil \frac{x}{A_i} \rceil$。如果 $C_i > m$，那么还需额外的 $D_i = \lceil \frac{x - m \cdot A_i}{B_i} \rceil$ 个自学课时；如果 $C_i < m$，则能挤出 $m - C_i$ 个额外的自学课时。能达到每门课的理解程度都 $\geq x$，当且仅当能挤出的自学课时数总和大于等于需要的自学课时的总和。注意如果需要的自学课时总和 $> N \times M$，那么需要直接返回不能，否则可能会爆 `long long`。

结合二分答案，我们就以 $O(N \log V)$ 的时间复杂度解决了这个问题。

联测缺额（exam）

子任务 3 的缝合操作不会改变题目爽度。所以直接做完全背包求出最大的变量数再乘以爽度就是答案。

观察 1：不需要考虑每种题目只有 m 个变种的限制。因为每种题目的变量数都是正整数，而最终总变量数限制为 m，所以只要满足了最终限制，必然满足了前面的限制。所以每种题目可以当成有无数个变种做。

此时我们进行二维完全背包，求出所有可能的缝合后题目，计算出它们的变量数和开心度，直接再求一个完全背包就可以得到答案。这个做法的时间复杂度为 $O(n \cdot m^2)$，瓶颈在于第一次的二维背包。它可以通过子任务 $1, 2$。

观察 2：同一个变量数的缝合后题目只需要保留开心度最大的一个。由于是完全背包，这是显然的。

所以我们在第一步时把关键变量数当成价值，做完全背包求出缝合后每个变量数对应的最大关键变量数，就可以将第一次背包的复杂度优化至 $O(n \cdot m)$，可以通过所有子任务。

联测缺颈（team）

核心思路是：当前有至少两项能力取到全局最大值的人一定不会被选入队伍。

我们不断删掉这些一定不会被选的人，同时维护当前剩余的所有人三项能力值分别的最大值 (a', b', c')。

这可以直接用 `std::multiset` 完成；也可以离散化之后维护每项能力的每个值的计数器（即在当前所有人中该项能力拥有该值的人数），并将当前的最大值暴力递减直到最大值的计数器不为 0。修改完全局最大值三元组后，将对应新增的至少有两项取到最大值的人加入删除队列。

一直删除直到不存在满足删除条件的人（删除队列为空），此时选择三个三项能力分别取到最大值的人并输出其能力和即可。若删空则说明无解，输出 `-1`。

复杂度 $O(n \log n)$ 或 $O(n)$。

联测缺尾（memory）

考虑使用线段树维护。每个节点维护一个堆，表示那些对于这个区间对应的整个节点的插入（相当于放在这个点上的，不止一个的永久化的懒标记）；以及一个 $maxval$，表示这个区间对应的子树内目前的最大值。

对于操作 1，直接找到对应的 $O(\log n)$ 个区间插入即可。

对于操作 3，直接找到对应的 $O(\log n)$ 个区间查询即可。

对于操作 2，首先执行一次操作 3 找到对应的数 T，接下来讨论：

- 如果区间代表的子树内都没有 T，不管。
- 如果区间代表的子树内有 T，但是区间维护的堆里没有 T，递归两棵子树即可。
- 否则，区间维护的堆里有 T，这个 T 可能会被拆分（与修改区间交的部分的 T 删除，另一部分的保留），递归下传下去这个懒标记即可。

复杂度容易分析出来是 $O(n\log^2 n)$。