

B. 蛋糕 (cakes)

时间限制: 2 秒

空间限制: 1024 MiB

Liliana 生日到了, 她邀请了所有好朋友来庆祝! 为了让派对更特别, 她打算准备多个蛋糕, 每个蛋糕上都装饰着各种配料, 比如草莓、杏仁或果仁糖。Liliana 有 N 种配料, 每种配料 i 有 a_i 个。

蛋糕的“美味度”由其上出现次数最多的配料的次数决定。比如:

- 蛋糕配料为 $\{1, 1, 2, 2, 2\}$ 时, 美味度为 3, 因为配料 2 出现了三次。
- 蛋糕配料为 $\{0, 0, 1, 1, 2\}$ 时, 美味度为 2, 因为配料 0 和 1 都出现了两次, 且没有其他配料出现次数更多。

Liliana 想烤几个美味度相同的蛋糕, 并且要用完所有的配料, 不能有剩余。她还没决定要烤多少个蛋糕。她正在考虑 Q 种方案, 每种方案指定了一个具体的蛋糕数量 K_j 。对于每种方案, 请判断是否能把所有配料分摊到 K_j 个蛋糕中, 且每个蛋糕的美味度都相同。蛋糕上的配料数量可以不同, 但每个蛋糕至少需要有一份配料。请注意, 不同的蛋糕可以包含不同数量的配料种类。

输入

第一行包含两个整数 N 和 Q , 分别代表配料种类数和方案数。第二行包含 N 个整数 a_0, a_1, \dots, a_{N-1} , 其中 a_i 表示配料 i 的数量。接下来的 Q 行, 每行包含一个整数 K_j , 指定了第 j 个方案所需的蛋糕数量。

输出

输出 Q 行。如果能将所有配料分摊到 K_j 个美味度相同的蛋糕中, 第 j 行输出 YES, 否则输出 NO。

约束条件

- $1 \leq N, Q \leq 100\,000$.
- $1 \leq a_i \leq 100\,000$.
- $1 \leq K_j \leq 10^{18}$.

评分方式

你的程序将在分成若干子任务的测试数据上进行测试。要获得某个子任务的分数, 你必须正确解出该子任务中所有的测试数据。

- 子任务 0 [0 分]: 样例。
- 子任务 1 [9 分]: $N = 1$ 。
- 子任务 2 [22 分]: $Q = 1$ 且 $K_j = 2$ 。
- 子任务 3 [24 分]: $Q \leq 5, N \leq 1000, a_i \leq 1000$ 。
- 子任务 4 [24 分]: $Q \leq 5$ 。
- 子任务 5 [21 分]: 没有额外的约束条件。

样例

stdin	stdout
4 5 2 5 1 1 1 2 3 4 5	YES NO YES NO YES
1 1 4 2	YES
5 3 1 1 1 1 1 1 10000000000000000000 5	YES NO YES

在第一个样例中，Liliana 有四种配料：两个 0 号配料（绿色三角形表示），五个 1 号配料（黄色星星表示），一个 2 号配料（橙色圆形表示），以及一个 3 号配料（蓝色正方形表示）。

当 $K = 1$ 时，Liliana 可以做一个美味度为 5 的蛋糕，把所有配料都放在一个蛋糕上：

- 蛋糕 1： $\{0, 0, 1, 1, 1, 1, 2, 3\}$ （配料 1 出现了五次）。



图 1 $K = 1$ 时的分配示例。

当 $K = 2$ 时，Liliana 不可能将所有配料分配到两个美味度相同的蛋糕中。

当 $K = 3$ 时，Liliana 可以做 3 个蛋糕，每个美味度为 2，分配如下：

- 蛋糕 1： $\{0, 0, 1\}$ （配料 0 出现了两次）。
- 蛋糕 2： $\{1, 1, 2\}$ （配料 1 出现了两次）。
- 蛋糕 3： $\{1, 1, 3\}$ （配料 1 出现了两次）。



图 2 $K = 3$ 时的分配示例。

当 $K = 4$ 时，Liliana 不可能将所有配料分配到四个美味度相同的蛋糕中。

当 $K = 5$ 时, Liliana 可以做 5 个蛋糕, 每个美味度为 1, 分配如下:

- 蛋糕 1: $\{0, 1\}$ (配料 0 和 1 各出现一次)。
- 蛋糕 2: $\{0, 1\}$ (配料 0 和 1 各出现一次)。
- 蛋糕 3: $\{1\}$ (配料 1 出现一次)。
- 蛋糕 4: $\{1, 2\}$ (配料 1 和 2 各出现一次)。
- 蛋糕 5: $\{1, 3\}$ (配料 1 和 3 各出现一次)。



图 3 $K = 5$ 时的分配示例。