



分组

班里有 N 个学生, 他们的编号为从 0 到 $N - 1$ 。每天, 老师都有一些项目需要学生去完成。每个项目都需要由一组学生在一天内完成。项目的难度可能不同。对于每个项目, 老师知道应该选择由多少学生组成的小组去完成。

不同的学生对小组的规模有不同的喜好。更准确地说, 对学生 i 而言, 他只愿意在小组规模介于 $A[i]$ 和 $B[i]$ 之间 (含 $A[i]$ 和 $B[i]$) 的小组工作。每一天, 一个学生最多只能被分配到一个小组工作。有些学生可能未被分配到任何小组中。每个小组只负责一个项目。

老师已选择好接下来 Q 天中每一天的项目。对于每一天, 现需要判断是否有一种分配学生的方案, 使得每个项目都有一个小组负责。

样例

设现有 $N = 4$ 个学生, 且有 $Q = 2$ 天。每个学生对于小组规模的喜好如下表:

学生	0	1	2	3
A	1	2	2	2
B	2	3	3	4

第一天共有 $M = 2$ 个项目。它们所需要的小组规模 (小组人数) 为 $K[0] = 1$ 及 $K[1] = 3$ 。这时可以将学生分为两个小组, 学生 0 被分配到小组规模为 1 的那个组, 而其余 3 个学生则被分配到小组规模为 3 的那个小组。

第二天共有 $M = 2$ 个项目, 但这一次它们所需要的小组规模分别为 $K[0] = 1$ 及 $K[1] = 1$ 。在这种情况下, 没有一种合适的学生分组方案, 因为只有一位同学愿意参加小组规模为 1 的小组。

任务

给定对所有学生的描述: N , A 及 B , 同时也给定了 Q 个问题的序列 — 每天一个问题。每个问题包含当天要完成的 M 个项目, 同时含有一个长度为 M 的序列 K , $K[i]$ 表示项目 i 所需的小组规模。对于每一个问题, 你的程序必须返回是否存在一种小组分配的方案, 可以完成当天的所有项目。

你需要实现两个函数, 它们分别是 `init` 和 `can` :

- `init(N, A, B)` — `grader` 在开始时将会调用这个函数恰好一次。
 - N : 学生的数目。
 - A : 一个长度为 N 的数组: $A[i]$ 是学生 i 愿意加入的小组的最小规模。
 - B : 一个长度为 N 的数组: $B[i]$ 是学生 i 愿意加入的小组的最大规模。

- 这个函数没有返回值。
- 对于 $i = 0, \dots, N-1$, 你可以假设 $1 \leq A[i] \leq B[i] \leq N$ 。
- $\text{can}(M, K)$ — 当调用完 init 一次之后, grader 将会连续调用本函数 Q 次, 每次对应于一天的数据。
 - M : 当天要完成的项目的数目。
 - K : 一个长度为 M 的数组, $K[i]$ 表示项目 i 所需的小组规模。
 - 若可以完成分组去完成当天所有的项目, 本函数返回 1, 否则, 应返回 0。
 - 你可以假设 $1 \leq M \leq N$, 且对于每个 $i = 0, \dots, M-1$ 都有 $1 \leq K[i] \leq N$ 。注意: $K[i]$ 之总和有可能大于 N 。

子任务

假设 S 代表所有调用 $\text{can}(M, K)$ 时的 M 的值的总和。

子任务	得分	N	Q	附加的限制条件
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	没有
2	13	$1 \leq N \leq 100,000$	$Q = 1$	没有
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

Sample grader

Sample grader 将以下面的格式读入相关数据:

- 第 1 行: N
- 第 2, ..., $N+1$ 行: $A[i] \ B[i]$
- 第 $N+2$ 行: Q
- 第 $N+3, \dots, N+Q+2$ 行: $M \cdot K[0] \cdot K[1] \cdot \dots \cdot K[M-1]$

对于每个问题, Sample grader 都会输出函数 can 的返回值。