这道题如果要使用暴力搜索直接求解会严重超时。实际上，我们可以发现，这个所谓的最短跳跃距离显然不能超过一个范围，而这个范围题目上已经给了出来。也就是说，答案是有一个确定的范围限制的，我们就可以考虑一种另外的方法去解决——枚举答案，并去验证答案是否可行。

实际上，枚举答案有时候也会超时。这就好比说你要从一本英汉词典上查一个单词，你从头到尾一页一页的翻着找，这样找可以保证一定能找到，但是最坏情况你要把整本词典都翻一遍，那就麻烦了。

有什么改进的方法吗？当然有。

考虑把这个词典从中间分开，看一下中间那一页的主要单词都是啥，然后去判断我要找的单词应该在左半部分还是右半部分，再去那一部分考虑怎么找就好了。同样的，在另一部分也是要进行划分并且判断的操作。这样一直进行下去，便能很快的找到答案，而且根本不需要翻过整个词典来。

可以证明，如果一页一页的找，最多要找n次，但是用这个方法，最多找floor(log2n)次。

我们把这个方法叫做“二分答案”。顾名思义，它用二分的方法枚举答案，并且枚举时判断这个答案是否可行。但是，二分并不是在所有情况下都是可用的，使用二分需要满足两个条件。一个是有界，一个是单调。

二分答案应该是在一个单调闭区间上进行的。也就是说，二分答案最后得到的答案应该是一个确定值，而不是像搜索那样会出现多解。二分一般用来解决最优解问题。刚才我们说单调性，那么这个单调性应该体现在哪里呢？

可以这样想，在一个区间上，有很多数，这些数可能是我们这些问题的解，换句话说，这里有很多不合法的解，也有很多合法的解。我们只考虑合法解，并称之为可行解。考虑所有可行解，我们肯定是要从这些可行解中找到一个最好的作为我们的答案， 这个答案我们称之为最优解。

最优解一定可行，但可行解不一定最优。我们假设整个序列具有单调性，且一个数x为可行解，那么一般的，所有的x'(x'<x)都是可行解。并且，如果有一个数y是非法解，那么一般的，所有的y'(y'>y)都是非法解。

那么什么时候适用二分答案呢？注意到题面：使得选手们在比赛过程中的最短跳跃距离尽可能长。如果题目规定了有“最大值最小”或者“最小值最大”的东西，那么这个东西应该就满足二分答案的有界性（显然）和单调性（能看出来）。

那就好办了。我们二分跳跃距离，然后把这个跳跃距离“认为”是最短的跳跃距离，然后去以这个距离为标准移石头。使用一个judge判断这个解是不是可行解。如果这个解是可行解，那么有可能会有比这更优的解，那么我们就去它的右边二分。为什么去右边？答案是，这个区间是**递增**的 ，而我们求的是最短跳跃距离的**最大值**，显然再右边的值肯定比左边大，那么我们就有可能找到比这更优的解，直到找不到，那么最后找到的解就有理由认为是区间内最优解。反过来，如果二分到的这个解是一个非法解，我们就不可能再去右边找了。因为性质，右边的解一定全都是非法解。那么我们就应该去左边找解。整个过程看起来很像递归，实际上，这个过程可以递归写， 也可以写成非递归形式，我个人比较喜欢使用非递归形式。

下一个问题，这个judge怎么实现呢？judge函数每个题有每个题的写法，但大体上的思想应该都是一样的——想办法检测这个解是不是合法。拿这个题来说，我们去判断如果以这个距离为最短跳跃距离需要移走多少块石头，先不必考虑限制移走多少块，等全部拿完再把拿走的数量和限制进行比对，如果超出限制，那么这就是一个非法解，反之就是一个合法解，很好理解吧。

可以去模拟这个跳石头的过程。开始你在i(i=0)位置，我在跳下一步的时候去判断我这个当前跳跃的距离，如果这个跳跃距离比二分出来的mid小，\*\*那这就是一个不合法的石头，应该移走。\*\*为什么？我们二分的是最短跳跃距离，已经是最短了，如果跳跃距离比最短更短岂不是显然不合法，是这样的吧。移走之后要怎么做？先把计数器加上1，再考虑向前跳啊。去看移走之后的下一块石头，再次判断跳过去的距离，如果这次的跳跃距离比最短的长，那么这样跳是完全可以的，我们就跳过去，继续判断，如果跳过去的距离不合法就再拿走，这样不断进行这个操作，直到i = n+1，为啥是n+1？河中间有n块石头，显然终点在n+1处。（这里千万要注意不要把n认为是终点，实际上从n还要跳一步才能到终点）。

模拟完这个过程，我们查看计数器的值，这个值代表的含义是我们以mid作为答案需要移走的石头数量，然后判断这个数量 是不是超了就行。如果超了就返回false，不超就返回true。

整道题我已经说完了。。。如果实在难以理解，请看代码。