

P3853 [TJOI2007]路标设置

题目描述

现在政府决定在公路上增设一些路标，使得公路的“空旷指数”最小。他们请求你设计一个程序计算能达到的最小值是多少。请注意，公路的起点和终点保证已设有路标，公路的长度为整数，并且原有路标和新设路标都必须距起点整数个单位距离。

输入格式

第 1 行包括三个数 L 、 N 、 K ，分别表示公路的长度，原有路标的数量，以及最多可增设的路标数量。

第 2 行包括递增排列的 N 个整数，分别表示原有的 N 个路标的位置。路标的位置用距起点的距离表示，且一定位于区间 $[0, L]$ 内。

输出格式

输出 1 行，包含一个整数，表示增设路标后能达到的最小“空旷指数”值。

输入样例

```
101 2 1
0 101
```

输出样例

```
51
```

解析

本题与《河中跳房子》有异曲同工之妙。大致思路如下：

不断的枚举“空旷指数”，对比在当前指数下所需要添加的路标数，是否等于目标。

如果多余目标，则说明间距太小，则左边界右增。

如果少于目标，则说明间距太大，则右边界左减。

等于目标时，我们需要更小的间距，因此也是有边界左减。

编码

```
#include<bits/stdc++.h>

using namespace std;
```

```

int const MaxN = 1e7 + 5;
long long n, m, a[MaxN];
long long len;

long long CountNum(long long x) {
    long long cnt = 0;
    //尝试放入标记牌
    for (int i = 2; i <= n; i++) {
        long long d = a[i - 1];
        //只要两个标志间距大于目标值，就放入一个
        while (a[i] - d > x) {
            cnt++;
            //将放入的标记作为新的起点
            d += x;
        }
    }
    return cnt;
}

long long Search(long long left, long long right) {
    while (left <= right) {
        long long mid = left + (right - left) / 2;
        long long num = CountNum(mid);
        //数量合适，但是要寻找尽可能短的距离，因此右边界左移
        if (num == m) {
            right = mid - 1;
        }
        //增加的路标数大于目标数，说明间距太小，应该增加间距
        else if (num > m) {
            left = mid + 1;
        }
        //增加的路标数小于目标数，说明间距太大，应该减少间距
        else if (num < m) {
            right = mid - 1;
        }
    }
    return left;
}

int main() {
    cin >> len >> n >> m;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    //边界从1到最长
    cout << Search(1, a[n]);
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

