

P2678 跳石头

题目描述

这项比赛将在一条笔直的河道中进行，河道中分布着一些巨大岩石。组委会已经选择好了两块岩石作为比赛起点和终点。在起点和终点之间，有 N 块岩石（不含起点和终点的岩石）。在比赛过程中，选手们将从起点出发，每一步跳向相邻的岩石，直至到达终点。

为了提高比赛难度，组委会计划移走一些岩石，使得选手们在比赛过程中的最短跳跃距离尽可能长。由于预算限制，组委会至多从起点和终点之间移走 M 块岩石（不能移走起点和终点的岩石）。

输入格式

第一行包含三个整数 L, N, M ，分别表示起点到终点的距离，起点和终点之间的岩石数，以及组委会至多移走的岩石数。保证 $L \geq 1$ 且 $N \geq M \geq 0$ 。

接下来 N 行，每行一个整数，第 i 行的整数 D_i ($0 < D_i < L$)，表示第 i 块岩石与起点的距离。这些岩石按与起点距离从小到大的顺序给出，且不会有两个岩石出现在同一个位置。

输出格式

一个整数，即最短跳跃距离的最大值。

输入样例

```
25 5 2
2
11
14
17
21
```

输出样例

```
4
```

解析

题解参考一本通的《河中跳房子》，需要注意的是本题的数据较大，需要使用 `long long` 类型，数组大小也需要注意。

编码

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

```

using namespace std;
long long n, m, a[50005];

long long CountDistance(long long x) {
    long long d = 0, cnt = 0;
    //依次比较每两块石头的间距，别忘了终点
    for (int i = 1; i <= n + 1; i++) {
        long long gap = a[i] - d;
        //两块石头的间距小于目标值，则必须搬掉一块
        if (gap < x) {
            cnt++;
        }
        //否则将新的石头作为起点
        else {
            d = a[i];
        }
    }
    //搬的石头大于目标数，说明间距太大，应该缩小间距
    return cnt;
}

long long Search(long long left, long long right) {
    while (left <= right) {
        long long mid = left + (right - left) / 2;
        long long num = CountDistance(mid);
        //数量合适，但是要寻找尽可能长的距离，因此左边界右移
        if (num == m) {
            left = mid + 1;
        }
        //搬的石头大于目标数，说明间距太大，应该缩小间距
        else if (num > m) {
            right = mid - 1;
        }
        //搬的石头小于目标数，说明间距太小，应该增加间距
        else if (num < m) {
            left = mid + 1;
        }
    }
    return right;
}

long long l = 0, r = 0;
long long len;

int main() {
    cin >> len >> n >> m;
    for (int i = 1; i <= n; i++) {

```

```
        cin >> a[i];
    }
    l = 0;    //左边界需要从0开始
    //别忘了将终点也要存入数组，并将其设为最大值
    r = a[n + 1] = len;
    cout << Search(l, r);
    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。

