

P1824 进击的奶牛

题目描述

Farmer John 建造了一个有 N ($2 \leq N \leq 100000$) 个隔间的牛棚，这些隔间分布在一条直线上，坐标是 x_1, \dots, x_N ($0 \leq x_i \leq 1000000000$)。

他的 C ($2 \leq C \leq N$) 头牛不满于隔间的位置分布，它们为牛棚里其他的牛的存在而愤怒。为了防止牛之间的互相打斗，Farmer John 想把这些牛安置在指定的隔间，所有牛中相邻两头的最近距离越大越好。那么，这个最大的最近距离是多少呢？

输入格式

第 1 行：两个用空格隔开的数字 N 和 C 。

第 2~ $N+1$ 行：每行一个整数，表示每个隔间的坐标。

输出格式

输出只有一行，即相邻两头牛最大的最近距离。

输入样例

```
5 3
1
2
8
4
9
```

输出样例

```
3
```

分析

根据题目中给出的示例，3 头牛分散在 1，4，9 这三个距离下是最合适的，其中最大的最小间距是 3，超过这个间距就无法放入奶牛了。

解题思路就是不断的枚举间距，在某个间距下最多能容纳的奶牛数量是否满足需求。

注意：第一头奶牛必然是在最左边。

编码

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

```

using namespace std;

//定义全局最大数
const int MaxNum = 1e6 + 1;

int n, target; //数组长度和目标数字
int a[MaxNum]; //原始的数目高度

//计算能容纳的牛的数量
long long Count(int gap) {
    //第一头牛安放的位置肯定是最边上的格子
    long long sum = 1;
    int last = a[0];
    for (int i = 1; i < n; ++i) {
        //如果两个牛舍之间的间距不小于目标
        if (a[i] - last >= gap) {
            sum++;
            last = a[i];
        }
    }
    return sum;
}

//nums 待搜索数组
//target 目标数字
//length 数组长度
//return 目标所在索引位置
int BinarySearchRight(int left, int right) {
    long long sum = 0;
    int mid = 0;
    //开始循环搜索
    while (left <= right) {
        mid = left + (right - left) / 2;
        sum = Count(mid);
        if (sum == target) {
            //持续向右搜索,左边界增加
            left = mid + 1;
        }
        //小于目标, 间距太大了, 边界左移
        else if (sum < target) {
            right = mid - 1;
        }
        //大于目标, 间距太小了, 边界右移
        else if (sum > target) {
            left = mid + 1;
        }
    }
    return right;
}

```

```
int main() {  
    //输入初始数据  
    cin >> n >> target;  
    for (int i = 0; i < n; ++i) {  
        cin >> a[i];  
    }  
    //将数据进行排序  
    sort(a, a + n);  
    //右边界搜索  
    int gap = BinarySearchRight(a[0], a[n - 1]);  
    //输出满足条件的高度  
    cout << gap;  
}
```

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。

