

P1873 砍树

题目描述

伐木工人 Mirko 需要砍 M 米长的木材。对 Mirko 来说这是很简单的工作，因为他有一个漂亮的新伐木机，可以如野火一般砍伐森林。不过，Mirko 只被允许砍伐一排树。

Mirko 的伐木机工作流程如下：Mirko 设置一个高度参数 H（米），伐木机升起一个巨大的锯片到高度 H，并锯掉所有树比 H 高的部分（当然，树木不高于 H 米的部分保持不变）。Mirko 就得到树木被锯下的部分。例如，如果一排树的高度分别为 20, 15, 10 和 17，Mirko 把锯片升到 15 米的高度，切割后树木剩下的高度将是 15, 15, 10 和 15，而 Mirko 将从第 1 棵树得到 5 米，从第 4 棵树得到 2 米，共得到 7 米木材。

Mirko 非常关注生态保护，所以他不会砍掉过多的木材。这也是他尽可能高地设定伐木机锯片的原因。请帮助 Mirko 找到伐木机锯片的最大的整数高度 H，使得他能得到的木材至少为 M 米。换句话说，如果再升高 1 米，他将得不到 M 米木材。

输入格式

第 1 行 2 个整数 N 和 M，N 表示树木的数量，M 表示需要的木材总长度。

第 2 行 N 个整数表示每棵树的高度。

输出格式

1 个整数，表示锯片的最高高度。

输入样例

```
4 7
20 15 10 17
```

输出样例

```
15
```

解析

不断的枚举砍伐的高度，直到找到最高的那个结果，注意计算结果比较大，需要使用 long long 类型。

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

//计算砍掉的总高度
long long Count(int high) {
    long long sum = 0;
    for (int i = 0; i < n; ++i) {
        if (a[i] > high) {
            int cut = a[i] - high;
            sum += cut;
        }
    }
    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,高度太大了,边界左移
        else if (sum < target) {
            right = mid - 1;
        }
        //大于目标,高度太矮了,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);  
//有边界搜索  
int high = BinarySearchRight(a[0], a[n - 1]);  
//输出满足条件的高度  
cout << high;  
}
```

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。

