跳石头比赛

【解题思路】
接下来描述一下二分的过程：
结果有可能是[0，L]的区间，可以用ans来存储二分的结果，ans初始化为0，就可以在[1，L]的区间去查找了。

1、先在1-L的区间计算出中间值mid。mid就是我们要去判断的是否符合的最小的最大距离。

2、怎么来判断，这就需要有个f函数。
先以起点为基点，如果从基点到第1块石头的距离小于mid（最小的最大距离），则移除第1块石头，再看接下来那块石头（原序号是第2块），如果还够不上mid，就继续移除。直至找到一块距离基点超过mid的石头，保留这块石头，并将它作为新的基点，再重复前面过程，逐一考察和移除在它之后的那些距离不足的石头，直至找到下一个基点予以保留。
当这个过程最终结束时，那些幸存下来的基点，彼此之间的距离肯定是大于当前设定的mid的。

跳石头比赛

3、
（1）这个时候要看一下被移除石头的总数，如果总数>M，则说明被移除的石头数量太多了（已超过上限值），进而说明当前设定的mid是过大的，应该移动右边界到mid-1，在左区间[1,r=mid-1]查找。
（2）如果总数≤M，则说明被移除的石头数量并未超过上限值，符合条件，当前值保存到ans里面，然后在右区间查找有没有更符合条件的值。

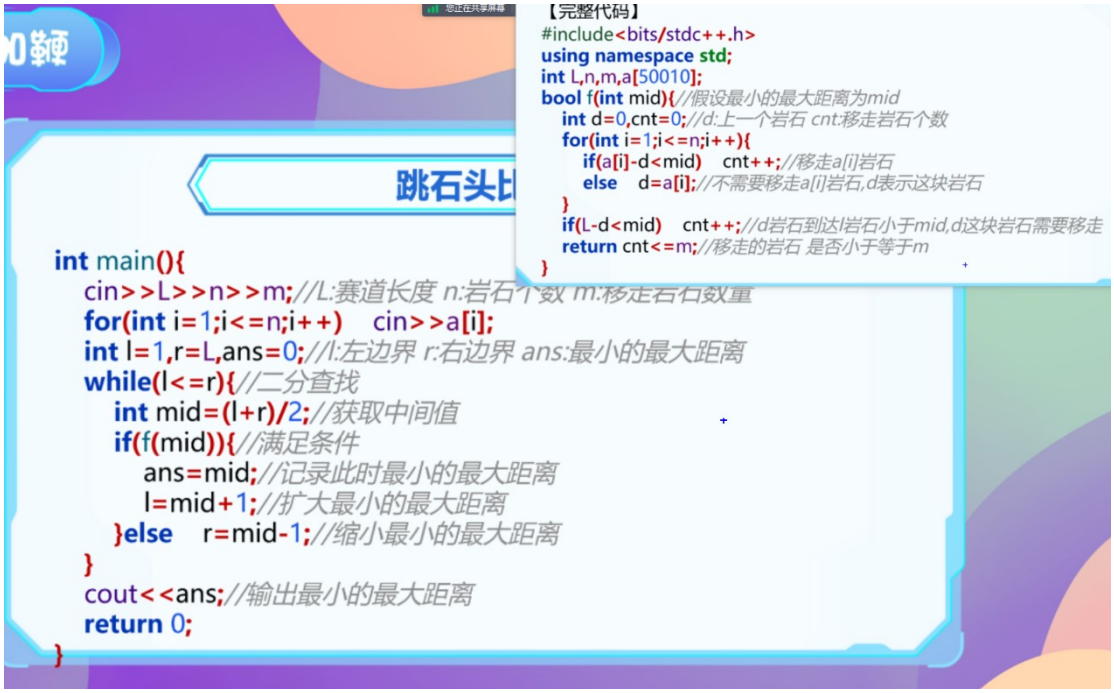
要点解析：题里面说不计算起始点和终点，但是还是要用终点减一下最后一块石头，求的是最后一块石头到终点的距离，这个距离不能大于mid。

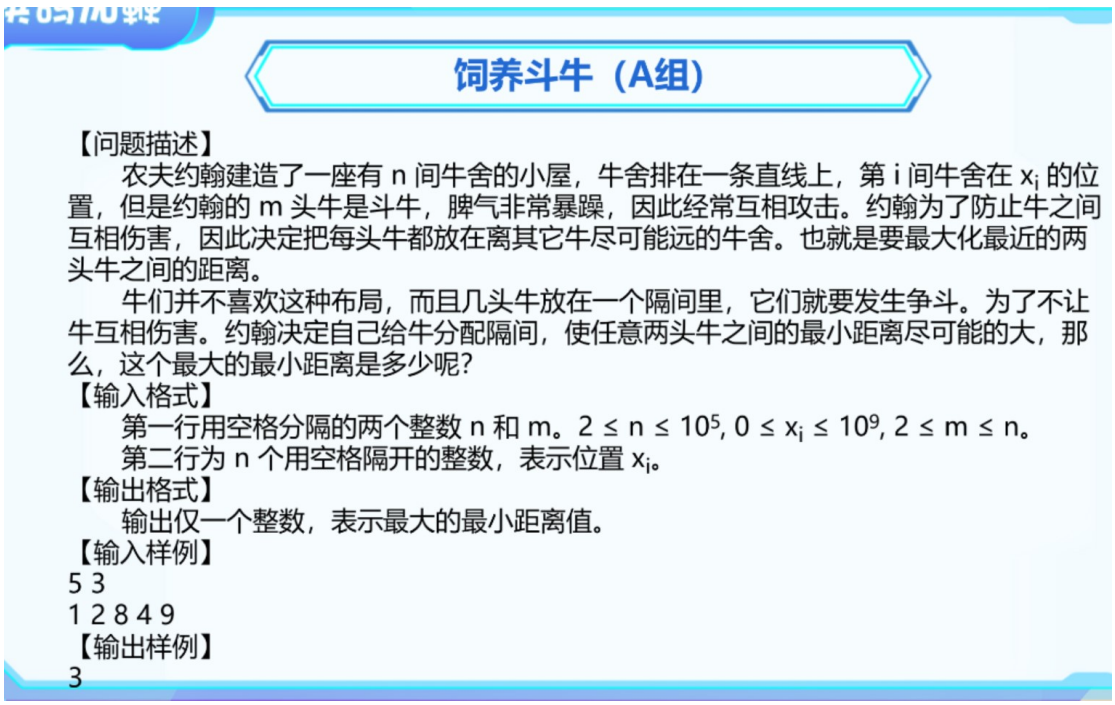
跳石头比赛

【样例解析】
L=25表示比赛河道长度为25　n=5表示河道内有5块岩石　m=2表示移动2块岩石

开始二分：
左边界l=1,右边界r=L=25。
中间值mid=(l+r)/2=(1+25)/2;

mid值是13





饲养斗牛（A组）

【问题描述】

农夫约翰建造了一座有 n 间牛舍的小屋，牛舍排在一条直线上，第 i 间牛舍在 x_i 的位置，但是约翰的 m 头牛是斗牛，脾气非常暴躁，因此经常互相攻击。约翰为了防止牛之间互相伤害，因此决定把每头牛都放在离其它牛尽可能远的牛舍。也就是要最大化最近的两头牛之间的距离。

牛们并不喜欢这种布局，而且几头牛放在一个隔间里，它们就要发生争斗。为了不让牛互相伤害。约翰决定自己给牛分配隔间，使任意两头牛之间的最小距离尽可能的大，那么，这个最大的最小距离是多少呢?

【输入格式】

第一行用空格分隔的两个整数 n 和 m。$2 \le n \le 10^5$, $0 \le x_i \le 10^9$, $2 \le m \le n$。

第二行为 n 个用空格隔开的整数，表示位置 x_i。

【输出格式】

输出仅一个整数，表示最大的最小距离值。

【输入样例】

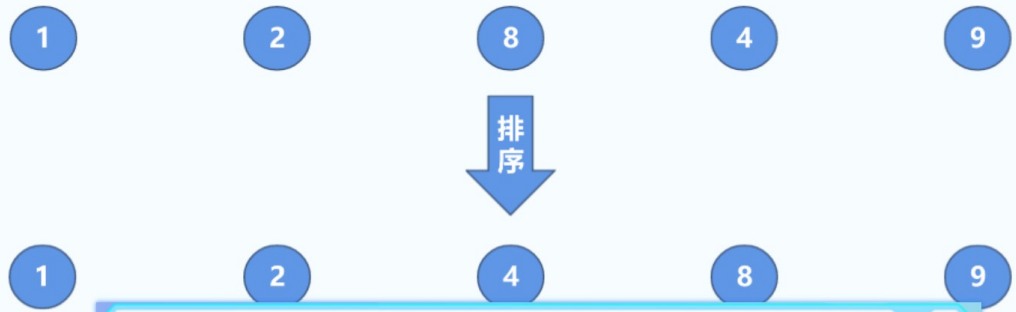
5 3
1 2 8 4 9

【输出样例】

3

饲养斗牛

【样例解析】
n=5表示有5个牛舍　m=3表示有3头斗牛

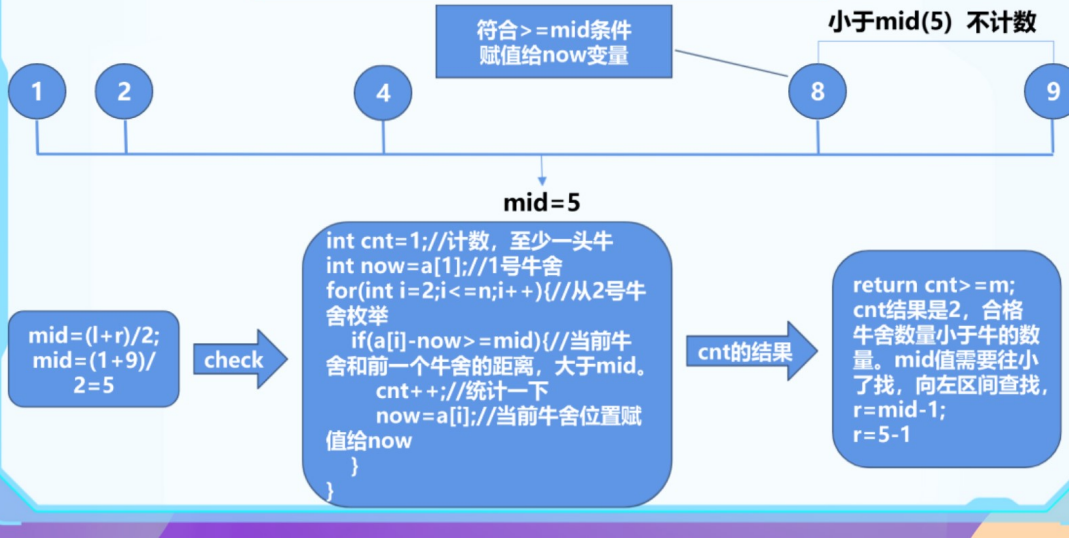


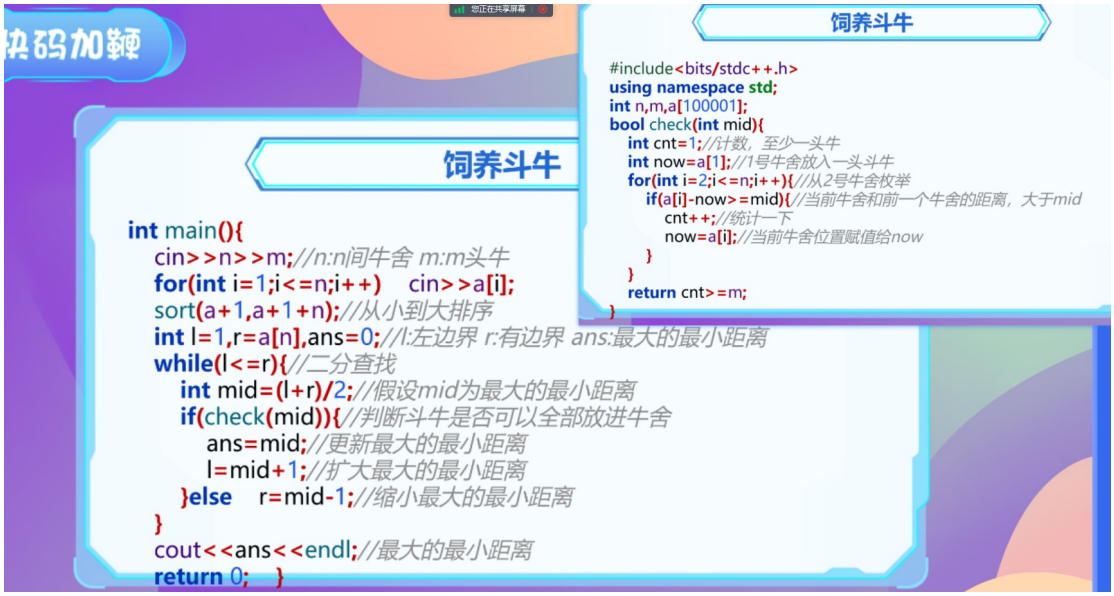
饲养斗牛

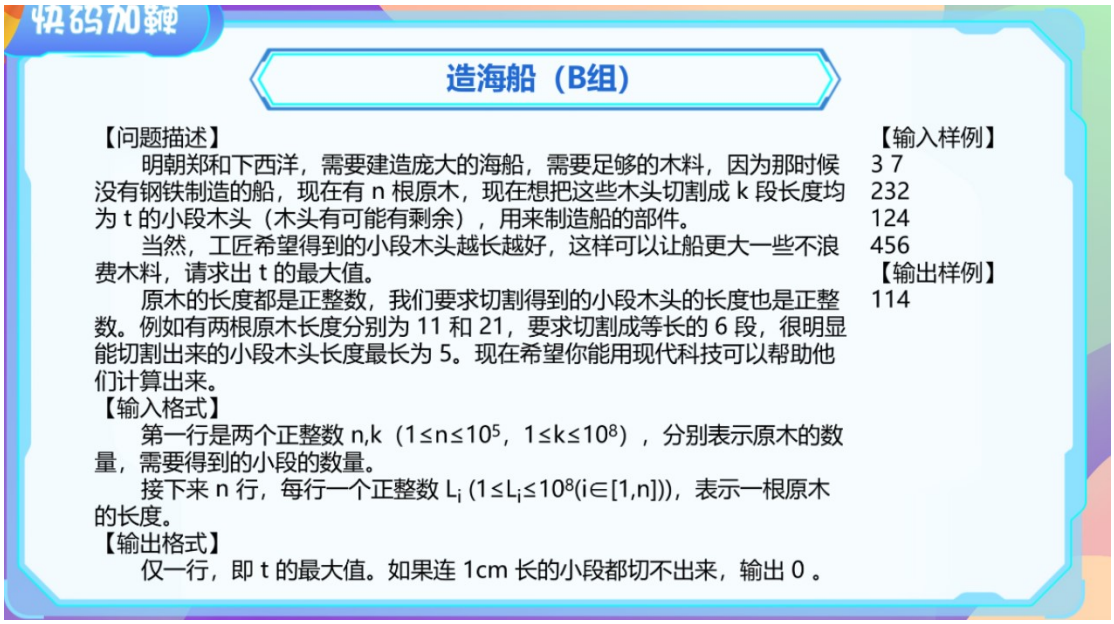
【问题分析】
题目要求所有的两头牛之间，最大的最小距离是多少，实际上就是求最小的最大距离。分析下这句话，让所有距离中的最小距离尽可能的大。
数据范围很大，枚举超时，同样采用二分来解决。

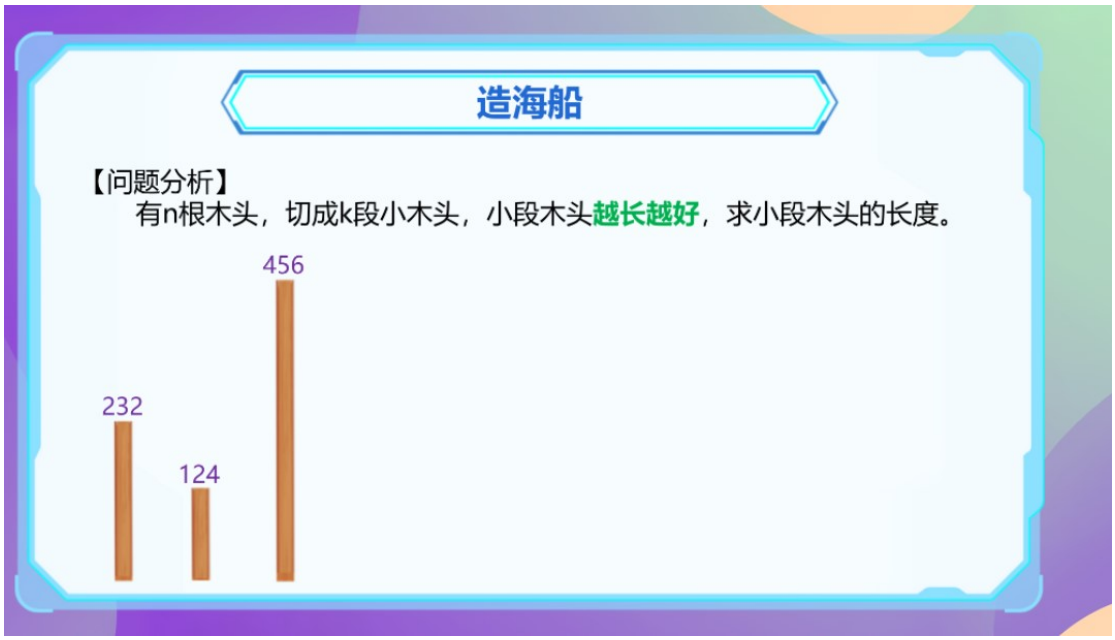
快码加鞭

饲养斗牛




饲养斗牛
int main(){
cin>>n>>m;//n:n间牛舍 m:m头牛
for(int i=1;i<=n;i++) cin>>a[i];
sort(a+1,a+1+n);//从小到大排序
int l=1,r=a[n],ans=0;//l:左边界 r:有边界 ans:最大的最小距离
while(l<=r){//二分查找
int mid=(l+r)/2;//假设mid为最大的最小距离
if(check(mid)){//判断斗牛是否可以全部放进牛舍
ans=mid;//更新最大的最小距离
l=mid+1;//扩大最大的最小距离
}else r=mid-1;//缩小最大的最小距离
}
cout<<ans<<endl;//最大的最小距离
return 0; }
饲养斗牛
#include<bits/stdc++.h>
using namespace std;
int n,m,a[100001];
bool check(int mid){
int cnt=1;//计数，至少一头牛
int now=a[1];//1号牛舍放入一头斗牛
for(int i=2;i<=n;i++){//从2号牛舍枚举
if(a[i]-now>=mid){//当前牛舍和前一个牛舍的距离，大于mid
cnt++;//统计一下
now=a[i];//当前牛舍位置赋值给now
}
}
return cnt>=m;
}


造海船（B组）
【问题描述】
明朝郑和下西洋，需要建造庞大的海船，需要足够的木料，因为那时候没有钢铁制造的船，现在有 n 根原木，现在想把这些木头切割成 k 段长度均为 t 的小段木头（木头有可能有剩余），用来制造船的部件。
当然，工匠希望得到的小段木头越长越好，这样可以让船更大一些不浪费木料，请求出 t 的最大值。
原木的长度都是正整数，我们要求切割得到的小段木头的长度也是正整数。例如有两根原木长度分别为 11 和 21，要求切割成等长的 6 段，很明显能切割出来的小段木头长度最长为 5。现在希望你能用现代科技可以帮助他们计算出来。
【输入格式】
第一行是两个正整数 n,k（1≤n≤10^5，1≤k≤10^8），分别表示原木的数量，需要得到的小段的数量。
接下来 n 行，每行一个正整数 Li (1≤Li≤10^8(i∈[1,n]))，表示一根原木的长度。
【输出格式】
仅一行，即 t 的最大值。如果连 1cm 长的小段都切不出来，输出 0 。
【输入样例】
3 7
232
124
456
【输出样例】
114



造海船

```
#include<iostream>
using namespace std;
int a[100001];//存储每个木头的长度
int l=0,r,n,k;//l:木头最小值 r:最长木头 n:原木的总数量 k:要切割成k段
int main(){
    cin>>n>>k;
    for(int i=0;i<n;i++){
        cin>>a[i];
        if(r<a[i])    r=a[i];
    }
    while(l+1<r){
        int mid=(l+r)/2;//中间值:切割长度
        int ans=0;//能切成的小木头数量
        for(int i=0;i<n;i++){
            ans=ans+a[i]/mid;//统计切成长度mid的木头数量
        }
        if(ans<k)//切割木头数量<k段木头r=mid;
        else //切割木头数量>=k段木头     l=mid;
    }
                                        cout<<l;
                                        return 0;
                                    }
```

作业：

1.程程的生活费

2.猜数字