Farmer John has built a new long barn, with N (2 <= N <= 100,000) stalls. The stalls are located along a straight line at positions x1,...,xN (0 <= xi <= 1,000,000,000).   
  
His C (2 <= C <= N) cows don't like this barn layout and become aggressive towards each other once put into a stall. To prevent the cows from hurting each other, FJ want to assign the cows to the stalls, such that the minimum distance between any two of them is as large as possible. What is the largest minimum distance?

Input

\* Line 1: Two space-separated integers: N and C   
  
\* Lines 2..N+1: Line i+1 contains an integer stall location, xi

Output

\* Line 1: One integer: the largest minimum distance

Sample Input

5 3

1

2

8

4

9

Sample Output

3

Hint

OUTPUT DETAILS:   
  
FJ can put his 3 cows in the stalls at positions 1, 4 and 8, resulting in a minimum distance of 3.   
  
Huge input data,scanf is recommended.

二分搜索值

#include <iostream>

#include <cstdio>

#include <cstring>

#include<algorithm>

#include<cmath>

#define INF 0x3f3f3f3f

using namespace std;

int n,m;

int a[100010];

bool judge(int val)

{

int last=0,crt;

for(int i=1;i<m;i++)

{

crt=last+1;

while(a[crt]-a[last]<val)

crt++;

if(crt>=n)return 0;

last=crt;

}

return 1;

}

int main()

{

scanf("%d%d",&n,&m);

for(int i=0;i<n;i++)

scanf("%d",&a[i]);

sort(a,a+n);

int l=0,r=INF;

while(l<=r)

{

int mid=(l+r)/2;

if(judge(mid))

l=mid+1;

else r=mid-1;

}

printf("%d\n",l-1);// 这一点我开始错误，误以为是mid ，最后一次循环肯定是left==right==mid, 但是mid 有可能不能放下c 只羊，我们能够确定left 左边的能够放下c 只羊，right 右边的不能放下c 只羊，可是mid 不能确定；如果mid 能够放下，则left 右移，left=right+1 ，显然不能放下，则left-1 是最大值；如果mid 不能放下，则left 不变，left 依然不能不能放下，left-1 是最大值。

//或者输出r也是对的，此时r就等于l-1

return 0;

}