题解

给定n个数，分为m个区间，保证m个区间的元素数目都是一样并且尽可能多，剩余的无法均分的则舍去，忽略。

网上见到很多人这道题都是二分+RMQ写得，但是，实际上是有问题的，因为并不是段数越多，值就越大，比如说：3 4 55 55 2 3，如果分为三段，结果4+55+3=62，而如果分为两段，55+55=110了，所以说这道题需要枚举+RMQ，可是这里如果不剪枝会特别悬，很有可能TLE，这里说到一个优化就是如果分为x段得到的长度len等于分成x-1段得到的长度len\_，那么只要把前一次的结果加上这一次第x段的最大值即可。

严谨的做法是枚举

但这个代码是二分的

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<cmath>

#include<algorithm>

#define inf 0x3f3f3f3f

using namespace std;

int n,k;

int dp[200005][20];

inline void init()

{

for(int j=1;j<=log2(n+1);++j)

for(int i=1;i<=n+1-(1<<j);++i)

dp[i][j]=max(dp[i][j-1],dp[i+(1<<(j-1))][j-1]);

}

int rmq(int l,int r)

{

int k=log2(r-l+1);

return max(dp[l][k],dp[r-(1<<k)+1][k]);

}

bool check(int m)

{

int i,sum=0,l=n/m;

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

{

sum+=rmq(l\*(i-1)+1,l\*i);

}

return sum>k;

}

int main()

{

int i,j;

while(scanf("%d%d",&n,&k))

{

if(n<0&&k<0)break;

int minn=inf,maxn=0,sum=0;

for(i=1;i<=n;++i)

{

scanf("%d",&dp[i][0]);

minn=min(minn,dp[i][0]);

maxn=max(maxn,dp[i][0]);

sum+=dp[i][0];

}

if(maxn==0)maxn=1;

if(minn==0)minn=1;

if(maxn>=k)

{

printf("1\n");

continue;

}

if(sum<k)

{

printf("-1\n");

continue;

}

init();

int l=k/maxn;

int r=min(k/minn+1,n);

int ans;

while(l<=r)

{

int mid=(l+r)>>1;

if(check(mid))

r=mid-1;

else

l=mid+1;

}

printf("%d\n",l);

}

return 0;

}