

解题报告
参考答案
<pre>#include<iostream> #define N 105 using namespace std; int i,n,m,x[N]; int main(){ cin>>n>>m; for(i=0;i<n;i++) cin>>x[i]; int cnt=1,sum=0; for(i=0;i<n;i++){ if(x[i]>m)break; sum+=x[i]; if(sum>m){ cnt++; sum=x[i]; } } if(i<n)cout<<0<<endl; else cout<<cnt<<endl; return 0; }</pre>
题目分析
题目考察贪心算法的理解和运用，需要注意的是：若单个罪犯智商超过 m，则无解。
提示
<p>题目是课件例题“臭皮匠 1”的变化，可参考课件第 4 页的程序。</p> <p>贪心法枚举每个罪犯的智商：如果某个罪犯 i 自己的智商就超过 m，则停止枚举，直接输出 0（无解）；否则，把该罪犯 i 的智商累加到分组智商和 sum，如果 sum 超过 m，另开一个新组，分组数量加 1，sum 初始化为该罪犯 i 智商。迭代下一罪犯。</p>
易错点
另开一个新组时，该组智商和 sum 不应初始化为 0，而是让上一组超出 m 的那个罪犯的智商。

解题报告
参考答案
<pre> #include<iostream> #include<algorithm> #define N 105 using namespace std; int i,n,k,x[N]; bool OK(int m){//让各组和都小于等于 m 时，分组数能否控制在 k 以内 int cnt=1,sum=0; for(i=0;i<n;i++){ if(x[i]>m)return 0; if((sum+=x[i])>m)cnt++,sum=x[i]; } return cnt<=k; } int main(){//最小化最大和 cin>>n>>k; for(i=0;i<n;i++)cin>>x[i]; int r=0; for(i=0;i<n;i++)r+=x[i]; int l=*max_element(x,x+n); int ans=r; while(l<=r){ int mid=l+(r-l)/2; if(OK(mid)) ans=mid,r=mid-1; else l=mid+1; } cout<<ans<<endl; return 0; } </pre>
题目分析
<p>题目考察整数答案的二分查找框架的理解和运用，需要注意的是：若单个罪犯智商超过 m，则此种情形不能成立。</p>
提示
<p>题目是课件例题“臭皮匠 2”的变化，可参考课件第 9 页程序及第 12 页最小优化思路。查找范围的左端点 l 初始为所有罪犯的智商最大值，右端点 r 初始为所有罪犯的智商和。只要 l 和 r 之间还有没搜索到的整数，则一直循环：计算中间值 mid 为 l 和 r 的平均；如果每组智商均不超过 mid 是可行的（具体算法可参考“高智商罪犯 1”），则记录答案为 mid，并试图在低的一半范围内寻找更小的优化解；否则在高的一半范围内寻找可行解。</p>
易错点
<ol style="list-style-type: none"> 1) 计算 mid 时不能直接使用 $(l+r)/2$，因为求和可能会溢出。 2) 判断每组的智商和是否均小于 m 时，忽略了某罪犯自己的智商就超过 m 的情形。

解题报告
参考答案
<pre> #include<iostream> #include<algorithm> #define N 505 using namespace std; int i,n,k,d[N]; bool OK(int m){//让间隔都大于等于 m 时，能否安排 n 人 int pos=d[0],cnt=1; for(i=1;i<k;i++) if(d[i]-pos>=m) cnt++,pos=d[i]; return cnt>=n; } int main(){//最大化相邻最小间隔 cin>>n>>k; for(i=0;i<k;i++)cin>>d[i]; sort(d,d+k); int l=1,r=1001,ans=1; while(l<=r){ int mid=l+(r-l)/2; if(OK(mid)) ans=mid,l=mid+1; else r=mid-1; } cout<<ans<<endl; return 0; } </pre>
题目分析
<p>题目考察整数答案的二分查找框架的理解和运用，需要注意的是：让高智商罪犯的牢房之间尽可能地远，可以转换为从大到小地试探对于某个整数 m，是否可以安排 n 名罪犯，彼此间隔都大于等于 m。但是贪心法暴力枚举 m 速度较慢，使用二分查找框架更好。</p>
提示
<p>题目是课件例题“臭皮匠 2”的变化，可参考课件第 9 页程序及第 13 页最大优化思路。查找范围的左端点 l 初始为 1（最左牢房编号），右端点 r 初始为 1001（最右牢房编号），答案 ans 初始为 1。只要 l 和 r 之间还有没搜索到的整数，则一直循环：计算中间值 mid 为 l 和 r 的平均；如果所有罪犯相距至少 mid 是可行的，则记录答案为 mid，并试图在高的一半范围内寻找更大的优化解；否则在低的一半范围内寻找可行解。</p>
易错点
<ol style="list-style-type: none"> 1) 计算 mid 时不能直接使用 $(l+r)/2$，因为求和可能会溢出。 2) 空牢房编号初始时要先排序。