[POJ 2566：Bound Found（Two pointers）](https://www.cnblogs.com/forever97/p/poj2566.html)

【题目链接】 <http://poj.org/problem?id=2566>

【题目大意】

　　给出一个序列，求一个子段和，使得其绝对值最接近给出值，  
　　输出这个区间的左右端点和区间和。

【题解】

　　因为原序列的前缀和不具有单调性，难以处理，  
　　因此我们对前缀和进行排序，同时保留前缀和的右端点做标识作用，  
　　题目要求区段和的绝对值最接近目标，因此排序不会造成前后顺序变化造成的影响  
　　现在题目转化为在一个有序数列中，求一个数对，使得差值最接近给出数，  
　　利用单调性，可以尺取解决问题。

【代码】

1. #include <cstdio>
2. #include <utility>
3. #include <algorithm>
4. #include <climits>
5. using namespace std;
6. const int N=100010,INF=INT\_MAX;
7. typedef pair<int,int> P;
8. int n,m,s,t,ans,ansl,ansr;
9. P p[N];
10. int main(){
11. while(scanf("%d%d",&n,&m),n&&m){
12. p[0]=P(s=0,0);
13. for(int i=1;i<=n;i++)scanf("%d",&t),p[i]=P((s+=t),i);
14. sort(p,p+n+1);
15. while(m--){
16. scanf("%d",&t);
17. int l=0,r=1,Min=INF;
18. while(l<=n&&r<=n){
19. int tmp=p[r].first-p[l].first;
20. if(abs(tmp-t)<Min){
21. Min=abs(tmp-t); ans=tmp;
22. ansl=p[l].second; ansr=p[r].second;
23. }if(tmp>t)l++;else if(tmp<t)r++;else break;
24. if(l==r)r++;
25. }if(ansl>ansr)swap(ansl,ansr);
26. printf("%d %d %d\n",ans,ansl+1,ansr);
27. }
28. }return 0;
29. }