1006

**题意:**给定一个序列，求和最大的连续子序列

**思路：**

题目限定N最大不超过10000，所以o（n^2）的效率是10^8过不了，最差也要o(nlogn)，这里提供最优化的o(n)算法，其他算法参考百度。

sum 当前大和

Max 当前最大和

start，end 开始和结尾的下标

X 当前可能的开始下标//下面会讲到用处

从第一个数开始扫描，扫描到i个数的时候

1. 若之前的sum<0，说明这之后的元素不应当记录在这个序列中，因为前缀<0了.记录x为当前元素的下标，从这个元素开始重新计算序列和，x = i，记录可能的开始的位置
2. 若sum > 0，sum+=num[i]，
3. 每次都判断当前序列和是否大于Max，若大于Max，更新Max=sum和start=x，end=i（这里注意因为x不一定是最大的序列的开始，所以前面1的时候不能直接改变s的值）

**代码：**

#include <iostream>

#include <cstdio>

using namespace std;

int num[10008];

int main()

{

int n,start,end,sum,Max,x;

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

{

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

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

start=end=0;

sum = x =0;

Max = -1e9;

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

{

if(sum<0)

{

x=i;

sum=num[i];

}

else

{

sum+=num[i];

}

if(sum>Max)

{

start=x;

end=i;

Max=sum;

}

}

printf("%d %d %d\n",Max, start, end);

}

return 0;

}