尺取法

简介&思想

通常是具有单调特征序列、寻找连续子区间接近目标值的问题中应用。
关注左右两端的指针，当选定的区间中所关注的值大于目标值时将左侧指针左移，当选定的区间中所关注的值小于目标值时将右指针右移，直至终点。枚举区间的复杂度为O(n)，计算区间的值的复杂度通常是O(1)，不排除有例外。
在这个过程中间接枚举了每个区间，但与直接枚举并不同，在尺取法应用的问题中一般对区间有要求，较短的区间往往无法符合要求因此将它们包含的较大的区间枚举到即可。

看几个例题

---

poj 3061 Subsequence

给定一个正数序列，使得其和大于或等于S，求最短的子序列长度。ouo

这道题比较基础，就充当模板了。（尺取法似乎没有特定的模板）

如果枚举两端的话复杂度就是O(n^2)，但是正数序列是单调的，也就是说扩大区间其和一定增大，减小区间其和一定减小；并且我们其实不关心每一个区间，我们关心的是和大于S的区间，而这样的区间由于单调性肯定是和越短和越小，因此对于同一个左/右位置我们关心的是使和尽可能小的但仍大于S的另一端。可能说得不是很明确，不过这道题从直觉上、一定的理性上，尺取法都是一个好的选择，而且将复杂度将至O(n)。

思路是，当选取的区间和大于S时就把左端点右移直至小于S，当区间和小于S时就将右端点右移直至大于S，等于的情况合理即可。

```
#include<cstdio>
using namespace std;
int n,s,t,a[100005];
int min(int x,int y)
{
    if(x==-1)
        return x;
    if(x>y)
        return y;
    return x;
}
int main()
{
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d%d",&n,&s);
        for(int i=1;i<=n;i++)
            scanf("%d",&a[i]);
```

```
        a[n+1]=0;
        int left=1,right=1,length=100005;
        long long sum=a[1];
        while(right<=n)
        {
            if(sum>=s)
            {
                length=min(length,right-left+1);
                sum-=a[left++];
            }
            else sum+=a[++right];
        }
        if(length==100005)
            length=0;
        printf("%d\n",length);
    }
    return 0;
}
```

---

poj 3320 Jessica's Reading Problem

一本书有 P 页，每页都有a[i]个知识点，知识点可能重复，求最少的连续页数来覆盖所有知识点。ouo

这个题虽然和上面的不同但是知识点种类仍然是单调的，随着区间的增缩而增缩，而且也是要连续页、最少页数，因此用尺取法可以解决。

代码略。

poj 2566 Bound Found

给定一个数组和一个值t求一个子区间使得其和的绝对值与t的差值最小。数组中的数可正可负。ouo

这个题看起来并不符合我们使用尺取法的条件。不过我们可以将其变形。

将前缀和排序，排序后每两个位置的前缀和之差代表一个原来的区间的和的绝对值(妙啊），同时在这个排序后的前缀和序列中随区间增缩所取的区间和的绝对值也增缩。这样运用尺取法是一个好的选择。

```
#include<cstdio>
#include<algorithm>
#define INF 0x3f3f3f3f
using namespace std;
int n,k,a;
typedef pair<int,int> unit;
unit sum[100005];
int main()
{
    while(scanf("%d%d",&n,&k)&&n)
```

```
    {
        for(int i=1;i<=n;i++)
        {
            scanf("%d",&a);
            sum[i]=unit(sum[i-1].first+a,i);
        }
        sort(sum+1,sum+1+n);
        for(int i=1;i<=k;i++)
        {
            scanf("%d",&a);
            int t=INF,l,r;
            for(int left=1,right=1;right<=n;right++)
            {
                while(sum[right].first-sum[left].first>a)
                {
                    if(sum[right].first-sum[left].first-a<t)
                    {
                        t=sum[right].first-sum[left].first-a;
                        l=sum[left].second;
                        r=sum[right].second;
                    }
                    left++;
                }
                if(a-sum[right].first+sum[left].first<t)
                {
                    t=a-sum[right].first+sum[left].first;
                    l=sum[left].second;
                    r=sum[right].second;
                }
            }
            printf("%d %d %d\n",sum[r].first-sum[l].first,l,r);
        }
    }
    return 0;
}
```

总结

运用尺取法一定要注意题目中序列的特征，因为尺取法并不能真正枚举每一个区间。

对序列进行变换也可能让尺取法可以应用。

参考

https://blog.csdn.net/lxt_lucia/article/details/81091597

From:
https://wiki.cvbbacm.com/ - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:no_morning_training:shaco:%E7%9F%A5%E8%AF%86%E7%82%B9:%E5%9F%BA%E7%A1%80:%E5%B0%BA%E5%8F%96%E6%B3%95&rev=1590914774

Last update: **2020/05/31 16:46**