

尺取法

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。在这个过程中枚举的区间的子区间的集合即是序列本来的所有子区间的集合，确保了间接枚举了每个区间，但与直接枚举并不同，一般在只关注符合特征的最短的区间的问题中这样做是正确的。

看几个例题

poj 3061: 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。

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<iostream>
#include<cstdlib>
#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()
{
    freopen("input.in","r",stdin);
    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[right];
            right++;
        }
    }
}
```

```
        sum-=a[left++];
    }
    else sum+=a[++right];
}
if(length==100005)
    length=0;
printf("%d\n",length);
}
return 0;
}
```

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

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:no_morning_training:%E5%B0%BA%E5%8F%96%E6%B3%95&rev=1588939440 

Last update: 2020/05/08 20:04