

```
    return 0;
}
```

I 最大子段和

难度	考点
5	贪心、动态规划

题目分析

由题意可知，我们要找出数列中**连续且非空**的一段，使其和最大，如果不看数据范围的话，我们可以很容易想到一种暴力的解法：直接枚举数列中所有存在的**连续且非空**的子列，将其和分别求出来，再用 `ans` 记录它们的最大值并输出，代码大致如下。

```
#include<stdio.h>
int main()
{
    int n,a[200005],ans=-10000000;
    scanf("%d",&n);
    for(int i=1;i<=n;i++) scanf("%d",&a[i]);
    for(int i=1;i<=n;i++)//枚举数列的左边界
    {
        for(int j=i;j<=n;j++)//枚举数列的右边界
        {
            int sum=0;
            for(int k=i;k<=j;k++)
            {
                sum+=a[k];//求和
            }
            if(sum>ans) ans=sum;//记录最大子段和
        }
    }
    printf("%d",ans);
    return 0;
}
```

但我们会发现这种思路只能通过30%的数据点，因为对于 n 在 10^5 的数量级时会时间超限（题目背景有说 `ddl` 快到了），所以要想一个更优的做法。

首先，我们用一个数组 `dp[i]` 表示数列中以 a_i 结尾的子列的最大子段和，之后顺序遍历整个数列。可以发现，对于 `dp[i]` 的取值分为两种可能：一种可能是将 a_i 与前面的子列合并，另一种是只取 a_i （这是因为如果前面的子段和加上 a_i 的值比 a_i 小，还不如不合并，只要 a_i ），因此可以得到如下状态转移方程：

$$dp[1] = a[1] \quad (i = 1)$$

$$dp[i] = \max(dp[i-1] + a[i], a[i]) \quad (2 \leq i \leq n)$$

由于数列可能分出很多段子列，最后要遍历 `dp[i]` 找出其中的最大值即为整个数列的最大子段和 `ans`。

示例代码

```
#include<stdio.h>
int main()
{
    int n;
    int a[200005],dp[200005];
    scanf("%d",&n);
    for(int i=1;i<=n;i++)
    {
        scanf("%d",&a[i]);
    }
    int ans=dp[1]=a[1];
    for(int i=2;i<=n;i++)
    {
        //按状态转移方程求dp[i]的值
        if(dp[i-1]+a[i]>a[i])
        {
            dp[i]=dp[i-1]+a[i];
        }
        else
        {
            dp[i]=a[i];
        }

        //ans始终为dp[i]中最大值
        if(dp[i]>ans)
        {
            ans=dp[i];
        }
    }
    printf("%d",ans);
    return 0;
}
```