

题意

给定 n, m ($n \leq 1e6$)

给 n 个数的序列

求选取 m 个不重叠子段，能得到最大的子段和是多少

分析

设 $dp[i][j]$ 表示第 j 个数在第 i 子段能得到的最大子段和

$dp[i][j] = \max(dp[i][j-1], \max(dp[i-1][k]) + a[j])$;

对于第 j 个数字，把它放到第 i 个子序列，有两种情况，一种是第 $j-1$ 个数字也是第 i 序列的，此时子段和为 $dp[i][j-1] + a[j]$ ，另一种是作为第 i 个序列的第一个数字，此时子段和为 $dp[i-1][k] + a[j]$, $k < j$ ， $dp[i][j]$ 就等于上述情况的最大子段和

这道题没有给出 m 的范围（毒瘤），可以看出， $dp[i][j]$ 只和本行左边一个数据以及上一行左边的数据有关，所以状态压缩 $dp[i][j]$ 为一维 $dp[j]$

$\max(dp[i-1][k])$ 可以在求上一行的时候就用数组存下来

卡时限过题，不知道有没有更简单的方法

代码

```
1 #include<iostream>
2 #include<cstring>
3 #include<algorithm>
4 #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
5 #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
6 #define Memset(a,b); memset((a),(b),sizeof((a)));
7 #define Cin(a); scanf("%d",&(a));
8 #define Cinc(a); scanf(" %c",&(a));
9 #define Cins(a); scanf("%s",(a));
10 #define Cout(a,b); printf("%d",(a));printf(b);
11 #define Coutc(a,b); printf("%c",(a));printf(b);
12 #define Couts(a,b); printf("%s",(a));printf(b);
13 using namespace std;
14 typedef long long LL;
15 typedef unsigned long long ULL;
16 typedef long double LDB;
17 inline int readint() {int x;cin>>x;return x;}
18 const int N = 1e6+5;
19 const long long INF = 922337203685477580;
```

```
20 int n,m;
21 LL a[N],lastmax[N],dp[N];
22 int main()
23 {
24     while(scanf("%d%d",&m,&n)!=EOF)
25     {
26         For(i,1,n)
27         {
28             scanf("%lld",&a[i]);
29             lastmax[i] = 0;
30         }
31         dp[0] = 0;
32         LL Max;
33         For(i,1,m)
34         {
35             Max = -INF;
36             For(j,i,n)
37             {
38                 dp[j] = max(dp[j-1],lastmax[j-1])+a[j];
39                 lastmax[j-1] = Max;
40                 Max = max(Max,dp[j]);
41             }
42         }
43         printf("%lld\n",Max);
44     }
45 }
```