

资源分配类动态规划

主讲老师：党东



背包类型动态规划是动态规划中的经典问题，在近几年联赛中，经常出现，甚至在NOIP2006的普及组和提高组中同时出现。

对于一般的情况，如果我们给出了某种资源，那么我们只要考虑将资源如何分配给每个阶段即可。





最大乘积类DP



【例1】乘积最大（NOIP2000普及） --1216



Description

今年是国际数学联盟确定的“2000——世界数学年”，又恰逢我国著名数学家华罗庚先生诞辰90周年。在华罗庚先生的家乡江苏金坛，组织了一场别开生面的数学智力竞赛的活动，你的一个好朋友XZ也有幸得以参加。活动中，主持人给所有参加活动的选手出了这样一道题目：

设有一个长度N的数字串，要求选手使用K个乘号将它分成K+1个部分，找出一种分法，使得这K+1个部分的乘积能够为最大。

同时，为了帮助选手能够正确理解题意，主持人还举了如下的一个例子：有一个数字串：312，当N=3，K=1时会有以下两种分法：

1) $3*12=36$

2) $31*2=62$

这时，符合题目要求的结果是： $31*2=62$

现在，请你帮助你的好朋友XZ设计一个程序，求得正确的答案。

【例1】乘积最大 (NOIP2000普及) --1216



Input

输入共有两行:

第一行共有2个自然数 N, K ($6 \leq N \leq 40$, $1 \leq K \leq 6$)

第二行是一个 K 度为 N 的数字串。

Output

输出所求得的最大乘积 (一个自然数) 。

Sample Input

4 2

1231

Sample Output

62

【例1】乘积最大（NOIP2000普及） --1216



【题目分析】

方法一：贪心法。尽可能平均分配各段，这样最终的数值将会尽可能大。但有反例。如191919分成3段： $19*19*19=6859$ ；但 $191*91*9=156429$ ，显然乘积更大。

将一个数分成若干段乘积后比该数小，因为输入数不超过20位，因此不需高精度运算。

证明：假设AB分成A和B,且 $A, B < 10$ ，则有 $AB = 10*A + B > A*B$ (相当于B个A相加)，同理可证明A, B为任意位也成立。

【例1】乘积最大 (NOIP2000普及) --1216



【题目分析】

方法二：动态规划

1.阶段和状态：

$\text{sum}[i][j]$ ：表示在s串中从i开始到j结束的数值；

$f[i][j]$ ：表示前j个字符中插入i个乘号所得的最大乘积；

2.状态转移方程：

初始化： $f[0][i]=\text{sum}[1][i]$;

状态转移方程： $f[i][j]=\max\{f[i-1][t]*\text{sum}[t+1][j]\}$

$(1 \leq i \leq k, i+1 \leq j \leq n, i \leq t \leq j-1)$

$\text{answer}=f[k][n]$

【例1】乘积最大（NOIP2000普及） --1216



巴蜀中學
BASHU SECONDARY SCHOOL

【核心代码】

```
int f[7][41],sum[41][41]={0},a[41],n,k;  
int figure(int x,int y)  
{ int i;  
    for(i=x;i<=y;i++) sum[x][y]=sum[x][y]*10+a[i];  
}
```



【例1】乘积最大 (NOIP2000普及) --1216



【核心代码】

```
int main()
{ int i,j,t,maxx;
  char c;
  cin>>n>>k; //n个数, k个乘号
  for(i=1;i<=n;i++){cin>>c;a[i]=c-'0';} //转换为数值
  for(i=1;i<=n;i++)
    for(j=i;j<=n;j++)figure(i,j);
  for(i=1;i<=n;i++)f[0][i]=sum[1][i]; //初始化
  for(i=1;i<=k;i++) //阶段: 枚举乘号数目
    for(j=i+1;j<=n;j++) //状态: 数字个数
    { maxx=0;
      for(t=i;t<=j-1;t++) //决策: 前t个数字中添加j-1个乘号
        if(f[i-1][t]*sum[t+1][j]>maxx)maxx=f[i-1][t]*sum[t+1][j];
      f[i][j]=maxx;
    }
  cout<<f[k][n]<<endl;
}
```

【例2】 机器分配 --1354



Description

总公司拥有高效设备M台，准备分给下属的N个分公司。各分公司若获得这些设备，可以为国家提供一定的盈利。问：如何分配这M台设备才能使国家得到的盈利最大？求出最大盈利值。其中 $M \leq 15$ ， $N \leq 10$ 。分配原则：每个公司有权获得任意数目的设备，但总台数不超过设备数M。

Input

第一行有两个数，第一个数是分公司数N，第二个数是设备台数M；
接下来是一个 $N \times M$ 的矩阵，表明了第I个公司分配J台机器的盈利。

Output

第一行输出最大盈利值；
接下N行，每行有2个数，即分公司编号和该分公司获得设备台数。
(多种方案中，输出靠后公司分配较多的方案)

【例2】 机器分配 --1354



Sample Input

3 3 //3个分公司分3台机器

30 40 50

20 30 50

20 25 30

Sample Output

70 //最大盈利值为70

1 1 //第一分公司分1台

2 1 //第二分公司分1台

3 1 //第三分公司分1台

【例2】 机器分配 --1354



【题目分析】

1、阶段和状态：

这是一个典型的动态规划试题。用公司数来做状态，数组 $f[i][j]$ 表示前 i 个公司分配 j 台机器的最大盈利。

下标 i 公司数表示阶段， j 机器数表示状态。

题目要求的是： **n 个公司分配 m 台机器的最大盈利 $f[n][m]$;**

2、状态转移方程：

$$f[i][j] = \max\{f[i-1][k] + a[i][j-k]\} \quad (1 \leq i \leq n, 1 \leq j \leq m, 0 \leq k \leq j)$$

初始值: $f[0][0] = 0$

时间复杂度 $O(n \cdot m^2)$

【例2】 机器分配 --1354



【核心代码】

公司作为阶段，机器数作为状态，

状态转移方程为： $f[i,j]=\max\{f[i-1,k]+v[i,j-k]\}$

$(1 \leq i \leq n, 1 \leq j \leq m, 0 \leq k \leq j)$

```
#include<iostream>
```

```
using namespace std;
```

```
int n,m;
```

```
int f[25][25]; //f[i,j]表示前i个公司分配j台机器的最大盈利。
```

```
int v[25][25]; //v[i,j]表示第i个公司分配j台机器的盈利
```

```
void print(int i,int j){
```

```
    for(int k=j;k>=0;k--)
```

```
        if(f[i][j]==f[i-1][j-k]+v[i][k]){
```

```
            if(i>1)print(i-1,j-k);
```

```
            cout<<i<<" "<<k<<endl; return ;
```

```
        }
```

```
}
```


【例2】 机器分配 --1354



【核心代码】

```
cin>>n>>m;
for(int i=1;i<=n;i++)
    for(int j=1;j<=m;j++)
        cin>>v[i][j];
f[1][1]=v[1][1];
for(int i=1;i<=n;i++)
{
    f[i][1]=f[i-1][1];
    for(int j=1;j<=m;j++)
        for(int k=0;k<=j;k++)
            f[i][j]=max(f[i][j],f[i-1][j-k]+v[i][k]);
}
cout<<f[n][m]<<endl;
print(n,m);
```

【练习】邮局问题 --1379



Description

一些村庄建在一条笔直的高速公路边上，我们用一条坐标轴来描述这条公路，每个村庄的坐标都是整数，没有两个村庄的坐标相同。两个村庄的距离定义为坐标之差的绝对值。我们需要在某些村庄建立邮局。使每个村庄使用与它距离最近的邮局，建立邮局的原则是：所有村庄到各自使用的邮局的距离总和最小。

Input

输入共计2行

第一行: n m {表示有 n 个村庄，建立 m 个邮局}

第二行: a_1 a_2 a_3 .. a_n {表示 n 个村庄的坐标}

Output

输出只有1行

第一行: l { l 表示最小距离总和}

Sample Input

10 5

1 2 3 6 7 9 11 22 44 50

Sample Output

9

【练习】邮局问题 --1379



【思路点拨】

- ①先读入 n 个村庄坐标，按照递增依次排列；
- ②求 $g[i][j]$ (表示在第 $a[i]$ 村庄到 $a[j]$ 村庄之间建立一个邮局的最小距离和， $g[i][j]=|a[k]-a[p]|$, $(1 \leq i \leq n, i \leq k \leq j, p=(i+j)/2)$);
- ③设 $f[i][j]$: 表示在前 j 个村庄建立 i 个邮局所得的最小距离和。
 $f[1][i]=g[1][i]$;
 $f[i][j]=\min\{f[i-1][k]+g[k+1][j], 2 \leq i \leq m, i \leq j \leq n, i-1 \leq k \leq j-1\}$



【练习】邮局问题 --1379



【参考代码】

```
int a[1005],n,m,sum[1005],g[1005][1005],f[1005][1005];
void Init()
{ int i,j,k,L;
  cin>>n>>m;
  for(i=1;i<=n;i++)cin>>a[i];
  qsort(1,n); //先按照距离快排
  for(i=1;i<=n-1;i++) //初始化g[i][j]
    for(j=i+1;j<=n;j++)
      { L=(i+j+1)/2;
        for(k=i;k<=j;k++)g[i][j]+=abs(a[k]-a[L]);
      }
}
```

【练习】邮局问题 --1379



【参考代码】

```
void DP()
{ int i,j,k,Min;
  for(i=1;i<=n;i++)f[1][i]=g[1][i];
  for(i=2;i<=m;i++) //阶段：邮局
    for(j=i;j<=n;j++) //状态：村庄
    { Min=0x7fffffff/2;
      for(k=i-1;k<=j-1;k++)
        if(Min>f[i-1][k]+g[k+1][j])Min=f[i-1][k]+g[k+1][j];
      f[i][j]=Min;
    }
  cout<<f[m][n]; }

int main()
{ Init();
  DP();
}
```