

信奥算法

动态规划

Dynamic Programming

DP实战步骤

定义状态

优化状态

手动
填写
数组
所有
格子

再总结状态转移方程

优
化
决
策

再总结边界状态

DP第1步：定义状态

自然定义法：仿照原题定义状态

答案为 $f[n][m]$

01背包

$f[i][j]$:前 i 件物品用载重 j 的包最多装的价值

正整数拆分

$f[i][j]$:将 i 做 j 拆分的方案数

状态具体化：比原题描述更详细

答案为 $\max_i \{f[i]\}$

最大子段和

$f[i]$:以 i 号数字结尾的最大子段和

LIS

$f[i]$:以 i 号数字结尾的上升子序列最长长度

乘积最大

一个长度为 N 的数字串，要求使用 K 个乘号将它分成 $K+1$ 个部分，找出一种分法，使得这 $K+1$ 个部分的乘积最大。

$1 \leq K < N \leq 10$

输入样例

4 2

1231

输出样例

62

请写出

DP状态定义

自然定义法

抄原题大法

$f[N][K]$: 前 N 个字符用 K 个乘号能得到的最大乘积

$f[i][p]$: 前 i 个字符用 p 个乘号能得到的最大乘积

答案直接输出 $f[N][K]$

DP第2步： 状态转移方程

答案为 $f[n][m]$

01背包

$f[i][j]$:前 i 件物品用载重 j 的包最多装的价值

方程1

$$f[i][j] = \max\{f[i-1][j], f[i-1][j-w_i] + v_i | j \geq w_i\}$$

方程2

若 $j \geq w_i$

$$f[i][j] = \max\{f[i-1][j], f[i-1][j-w_i] + v_i\}$$

若 $j < w_i$

$$f[i][j] = f[i-1][j]$$

每一格只依赖上一行

正上方邻居,左上方某格

代码和
方程相同

```
if(j >= w[i]) f[i][j] = max(f[i-1][j], f[i-1][j-w[i]] + v[i]);  
else f[i][j] = f[i-1][j];
```

DP第2步： 状态转移方程

正整数
拆分

$f[i][j]$:将*i*做*j*拆分的方案数

答案为 $f[n][m]$

方程1

$$f[i][j] = f[i-j][1] + f[i-j][2] + \dots + f[i-j][j]$$

方程2

$$f[i][j] = f[i-1][j-1] + f[i-j][j]$$

每一格只依赖左一列

左上方邻居,正上方某格

代码和
方程相同

$$f[i][j] = f[i-1][j-1] + f[i-j][j];$$

快
kkcoding

DP第2步： 状态转移方程

最大子
段和

$f[i]$:以i号数字结尾的最大子段和

答案为
 $\max_i \{f[i]\}$

$$f[i] = \max\{f[i - 1], 0\} + x[i]$$

每一格只依赖左侧邻居

代码和
方程相同

$f[i] = \max(f[i-1], 0) + x[i];$

DP第2步： 状态转移方程

答案为 $\max_i \{f[i]\}$

LIS

$f[i]$:以i号数字结尾的上升子序列最长长度

易错写法: $f[i] = \max_{j < i} \{f[j] + 1 | x[j] < x[i]\}$

$$f[i] = \max_{j < i} \{f[j] | x[j] < x[i]\} + 1$$

左侧若干格 挑选条件最大值

打擂台求最值

首位擂主是几

1

代码和
方程不同

```
f[i]=1;
```

```
for(int j=0;j<i;j++)
```

```
if(x[j]<x[i]) f[i]=max(f[i],f[j]+1);
```

易错点：若干数求最值
打擂台的擂主初始化错误

快快
kkcoding.net

摆花

花店门口要摆上一排花，共 m 盆($0 < m \leq 100$)。小明列出了顾客最喜欢的 n ($1 < n \leq 100$)种花，从1到 n 标号。规定第 i 种花不能超过 a_i 盆，同一种花放在一起，且不同种类的花需按标号从小到大摆列，一共有多少种不同的摆花方案？

总数对1000007取模。

取模有哪些易错点

输入样例

2 4

3 2

输出样例

2

请写出
DP状态定义

自然定义法

抄原题大法

$f[n][m]$:前 n 种花排 m 盆有几种方案

$f[i][j]$:前 i 种花排 j 盆有几种方案

答案直接
输出 $f[n][m]$

摆花

$f[i][j]$: 前*i*种花排*j*盆有几种方案

答案 $f[n][m]$

输入样例

2 4

3 2

2种花要排4盆

第1种有3盆

第2种有2盆

| $f[i][j]$ | $j=0$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ |
|-----------|-------|-------|-------|-------|-------|
| $i=0$ | 1 | 0 | 0 | 0 | 0 |
| $i=1$ | 1 | 1 | 1 | 1 | 0 |
| $i=2$ | 1 | 2 | 3 | 3 | 2 |

请总结出状态转移方程
 $f[i][j]$ 依赖哪些格子

思考决策是什么

尝试理解 $f[i][j]$
 左/上方邻居格子的含义
 能否为 $f[i][j]$ 计算做贡献

摆花

$f[i][j]$: 前*i*种花排*j*盆有几种方案

答案 $f[n][m]$

$$f[i][j] = \sum_{k=0,1,\dots,\min(j,a[i])} f[i-1][j-k]$$

决策

选择题

枚举第*i*种花
摆*k*盆

剩下*j-k*盆

时间复杂度 $O(nm^2)$

能否再优化?

重大发现

$f[i][j]$ 依赖上一行的连续和!

摆花

$f[i][j]$: 前 i 种花排 j 盆有几种方案

答案 $f[n][m]$

若 $j - 1 < a[i]$

$$f[i][j] = f[i - 1][j] + f[i][j - 1]$$

决策

是非题

不摆第 i 种
的方案数

摆第 i 种花
的方案数

若 $j - 1 \geq a[i]$

$$f[i][j] = f[i - 1][j] + f[i][j - 1] - f[i - 1][j - 1 - a[i]]$$



$f[i][j - 1]$ 种方案里恰好摆了
 $a[i]$ 盆第 i 种花的方案数
要扣除掉

时间复杂度
 $O(nm)$

现场挑战

708

快快编程
kkcoding.net

请写出
DP状态定义

| | | | |
|---|-------|-----|-----------------|
| 自然定义法 | 抄原题大法 | 不行! | 不确定调整几次 |
| $f[i][j]$:对前 <i>i</i> 枚导弹最多用 <i>j</i> 次调整 最多拦截几枚? | | | 答案 $f[n][m]$ |

| | | |
|---|----------|--|
| 状态具体化 | 比原题描述更详细 | 明确调整次数 |
| $f[i][j]$:对前 <i>i</i> 枚导弹恰好用 <i>j</i> 次调整 最多拦截几枚? | | 答案 $\max_{0 \leq j \leq m} \{f[n][j]\}$ |

明确当前方向

| | |
|-------|----------|
| 状态具体化 | 比原题描述更详细 |
|-------|----------|

| |
|---|
| $f[i][j]$:对前 <i>i</i> 枚导弹恰好用 <i>j</i> 次调整 最多拦截几枚? |
|---|

| | |
|--------|-------------|
| 明确当前方向 | 请问当前方向是F/B? |
|--------|-------------|

| | |
|-------------|---|
| 截止目前共用过1次调整 | B |
| 截止目前共用过2次调整 | F |
| 截止目前共用过3次调整 | B |
| 截止目前共用过4次调整 | F |

| | |
|-------|----------|
| 状态具体化 | 比原题描述更详细 |
|-------|----------|

| |
|---|
| $f[i][j]$:对前 <i>i</i> 枚导弹恰好用 <i>j</i> 次调整 最多拦截几枚? |
|---|

| | |
|--------|-------------|
| 明确当前方向 | 请问当前方向是F/B? |
|--------|-------------|

| | |
|--------|-----------|
| $j\%2$ | 0代表F,1代表B |
|--------|-----------|

```

10      cin>>n>>m>>s;
11      for(int i=1;i<=n;i++)
12          a[i]=(s[i-1]=='F')?0:1;

```

| | |
|------------|-----------|
| 当前导弹方向a[i] | 0代表F,1代表B |
|------------|-----------|

| | |
|---|--------------|
| 对前 <i>i</i> 枚导弹恰好用 <i>j</i> 次调整,能否拦截a[i]? | $a[i]==j\%2$ |
|---|--------------|

$f[i][j]$:对前*i*枚导弹恰好用*j*次调整
最多拦截几枚?

答案
 $\max_{0 \leq j \leq m} \{f[n][j]\}$

输入样例
7 2
BFFBBFF

输出样例
6

| $f[i][j]$ | $j=0$ | $j=1$ | $j=2$ |
|-----------|-------|-------|-------|
| $i=0$ | 0 | 0 | 0 |
| $i=1$ | 0 | 1 | 0 |
| $i=2$ | 1 | 1 | 2 |
| $i=3$ | 2 | 1 | 3 |
| $i=4$ | 2 | 3 | 3 |
| $i=5$ | 2 | 4 | 3 |
| $i=6$ | 3 | 4 | 5 |
| $i=7$ | 4 | 4 | 6 |

请总结出状态转移方程
 $f[i][j]$ 依赖哪些格子

尝试理解 $f[i][j]$
左/上方邻居格子的含义
能否为 $f[i][j]$ 计算做贡献

思考决策是什么

$f[i][j]$:对前*i*枚导弹恰好用*j*次调整
最多拦截几枚?

答案

$$\max_{0 \leq j \leq m} \{f[n][j]\}$$

决策

对*i*号导弹是否调整方向

$$f[i][j] = h[i][j] + \max\{f[i-1][j], f[i-1][j-1]\}$$

$$h[i][j] = (a[i] == j \% 2)$$

$h[i][j]$ 代表调整*j*次后
*i*号导弹是否能拦截

$f[i][j]$:对前*i*枚导弹恰好用*j*次调整
最多拦截几枚?

答案

$\max_{0 \leq j \leq m} \{f[n][j]\}$

决策

对*i*号导弹是否调整方向

$$f[i][j] = h[i][j] + \max\{f[i-1][j], f[i-1][j-1]\}$$

```
10  cin>>n>>m>>s;
11  for(int i=1;i<=n;i++)
12      a[i]=(s[i-1]=='F')?0:1;
13  for(int j=0;j<=m;j++)
14      for(int i=j;i<=n;i++)
15          h[i][j]=(a[i]==j%2);
16  for(int i=1;i<=n;i++)
17      f[i][0]=h[i][0]+f[i-1][0];
18  for(int j=1;j<=m;j++)
19      for(int i=j;i<=n;i++)
20          f[i][j]=h[i][j]+max(f[i-1][j],f[i-1][j-1]);
21  cout<<*max_element(f[n],f[n]+m+1)<<endl;
```

快快编程
kkcoding.net

现场挑战

664

快快编程
kkcoding.net

动态规划

状态
定义

自然定义法/抄原题定义法

$f[n][m]$ 代表 n 个人里选 m 人
最小的不齐程度

$f[i][j]$ 代表前 i 个人里选 j 人
最小的不齐程度

输出
答案

$f[n][m]$

此方法推导转移方程时会有困难!

因为选了谁没有写明,无法计算差
距值

动态规划

状态
定义

状态具体化: 提供更详细信息

$f[i][j]$ 代表选中 i 号作结尾
共选 j 人时, 最小的不齐程度

输出
答案

$\min\{f[m][m], f[m+1][m], \dots, f[n][m]\}$

此方法推导转移方程时会方便些

决策
变量

结尾是 i 号时
枚举 i 号左边邻居是几号

$f[i][j]$ 代表用 i 号结尾共选 j 人时最小的不齐程度

$n=7, m=4$

| $y[i]$ | $f[i][j]$ | $j=0$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ |
|--------|-----------|-------|-------|-------|-------|-------|
| | $i=0$ | | | | | |
| 6 | $i=1$ | | 0 | | | |
| 5 | $i=2$ | | 0 | 1 | | |
| 9 | $i=3$ | | 0 | 3 | 5 | |
| 3 | $i=4$ | | 0 | 2 | 3 | 11 |
| 7 | $i=5$ | | 0 | 1 | 3 | 7 |
| 4 | $i=6$ | | 0 | 1 | 2 | 4 |
| 9 | $i=7$ | | 0 | 0 | 3 | 5 |

请设计
填表顺序

每格依赖
左侧一列
左上方
若干格

结尾是 i 号时
枚举 i 号左边邻居是几号

代码

```
11 for(int i=1;i<=n;i++)f[i][1]=0;
12 for(int j=2;j<=m;j++)
13     for(int i=j;i<=n;i++){
14         f[i][j]=INF;
15         for(int k=j-1;k<i;k++)
16             f[i][j]=min(f[i][j],f[k][j-1]+abs(y[i]-y[k]));
17     }
18 int ans=INF;
19 for(int i=m;i<=n;i++) ans=min(ans,f[i][m]);
20 cout<<ans<<endl;
```

枚举j代表选中人数

枚举i代表结尾编号

结尾是i号时枚举k
代表i号左边邻居是几号

状态变量
i和j

决策变量
k

查错方法

```
for(int i=0;i<=n;i++,cout<<endl)
    for(int j=0;j<=m;j++)cout<<f[i][j]<<"\t";
```

| | |
|------------|------------|
| 打印整张 表格 | 核对手算 结果 |
|------------|------------|

调试查错



作业要求

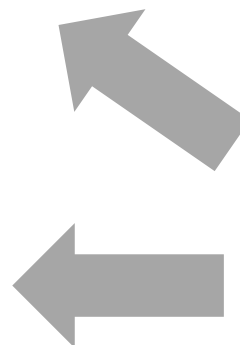
写程序前请写明：

1. 二维数组每一格的含义
2. 手算样例对应表格

查错方法：

1. 打印二维数组
2. 和手算表格对比找不同

```
1  /*
2  f[i][j]代表
3      j=0,1,2,3,4,5
4  i=0
5  i=1
6  i=2
7  i=3
8  i=4
9  i=5
10 */
```



模版格式
供参考

快快编程作业

48

708

710

664

拓展题

203,685,709