算法设计与分析

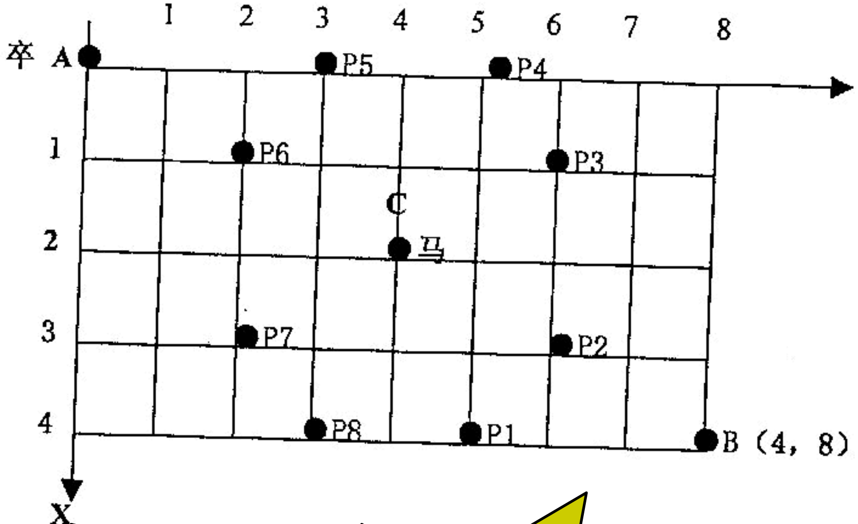
动态规划2

例题六.　马拦过河卒

实例

[问题描述]:

如图，**A** 点有一个过河卒，需要走到目标 **B** 点。卒行走规则：可以向下、或者向右。同时在棋盘上的任一点有一个对方的马（如上图的C点），该马所在的点和所有跳跃一步可达的点称为对方马的控制点。例如上图 **C** 点上的马可以控制 **9** 个点（图中的**P1，P2 … P8** 和 **C**）。卒不能通过对方马的控制点。



为什么不暴力?

棋盘用坐标表示，A 点（0，0）、B 点（n,m）(n,m 为不超过 20 的整数，并由键盘输入)，同样马的位置坐标是需要给出的（约定: C<>A，同时C<>B）。现在要求你计算出卒从 A 点能够到达 B 点的路径的条数。

[输入]:

键盘输入

B点的坐标（n,m）以及对方马的坐标（X,Y）

[输出]:

屏幕输出

一个整数（路径的条数）。

[输入输出样例]:

输入：

6 6 3 2

输出：

17

分析：阶段：棋盘上的每个可走的点；

每个阶段的求解；

步骤1：用$F(X,Y)$表示到棋盘上每个阶段(X,Y)的路径条数；

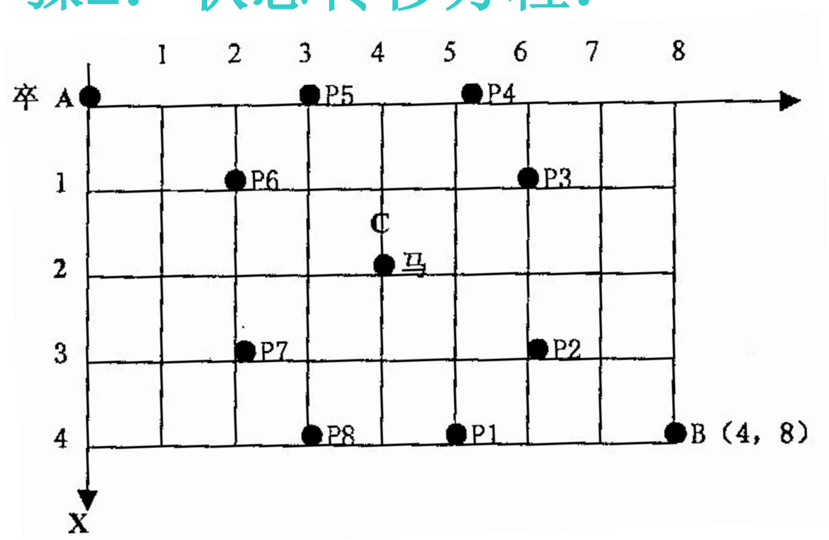
步骤2：状态转移方程：

$$F(X,Y) = F(X-1,Y) + F(X,Y-1)$$

其中，$F(0,0) = 1$

$$F(0,Y) = 1$$

$$F(X,0) = 1$$



马的影响怎么体现？

步骤3：以自底向上的方法来计算最优解

例题七：数字三角形问题

1．问题描述

设有一个三角形的数塔，顶点结点称为根结点，每个结点有一个整数数值。从顶点出发，可以向左走，也可以向右走。如图10一1所示。

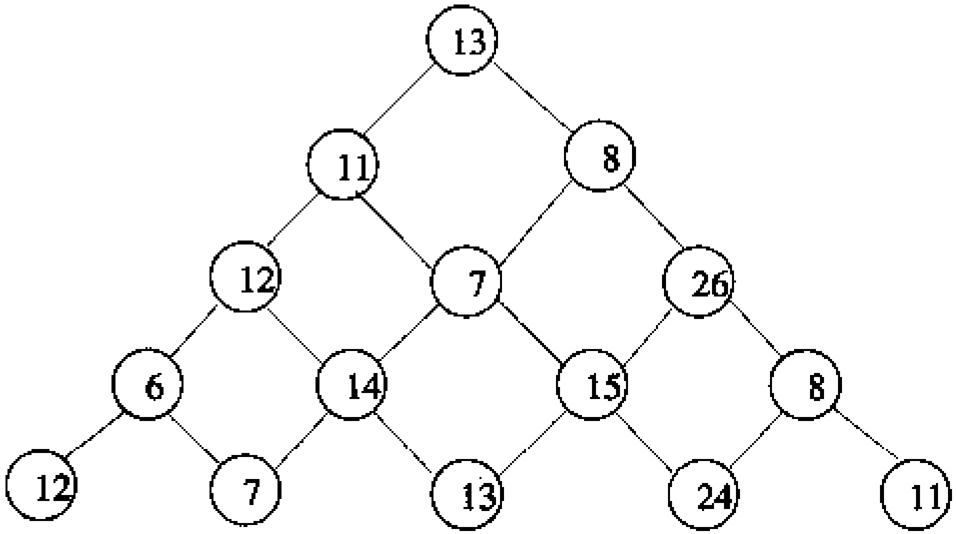


图 10－1

问题：当三角形数塔给出之后，找出一条从第一层到达底层的路径，使路径的值最大。若这样的路径存在多条，任意给出一条即可。

步骤1

二维数组 $D(x,y)$描述问题，$D(x,y)$表示从顶层到达第x层第y个位置的最小路径得分。

阶段分析：$D(1,1)=13$

到第x层的第y个位置有两种可能，要么走右分支 得到，要么走左分支得到。

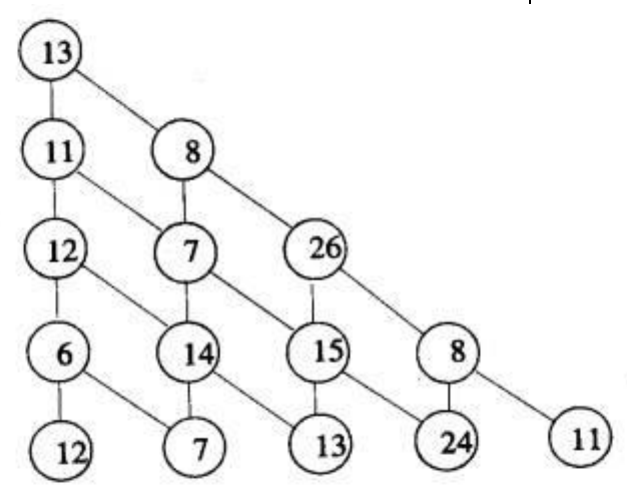


图 10 - 4

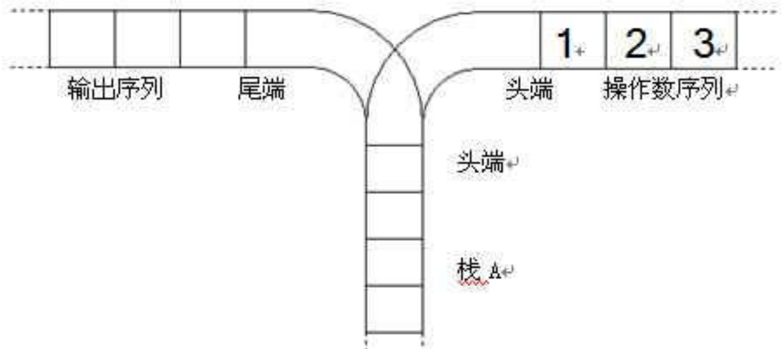
步骤2：状态转移方程

- $D(x,y)=\min\{D(x-1,y)，D(x-1,y-1\}+a(x,y)$
- D(1, 1)＝a(1, 1)

处理边界！

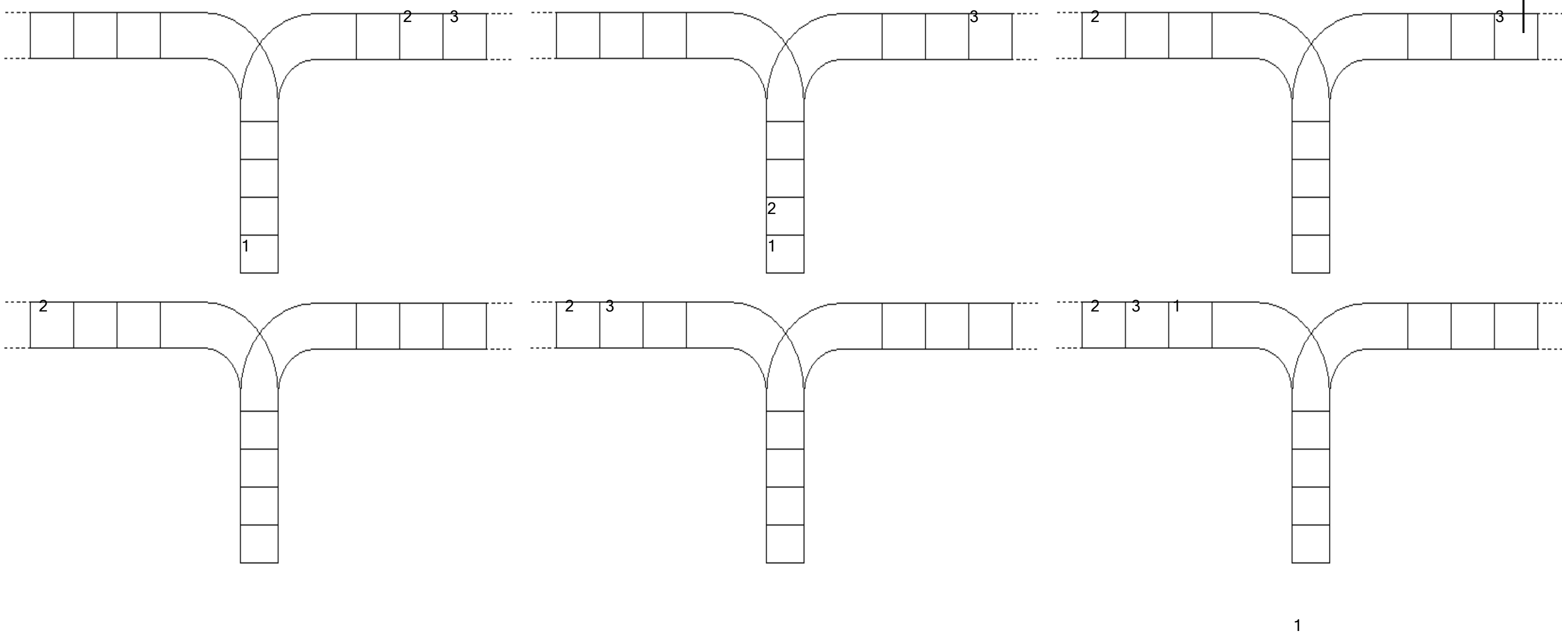
拓展：栈（vijos 1122）

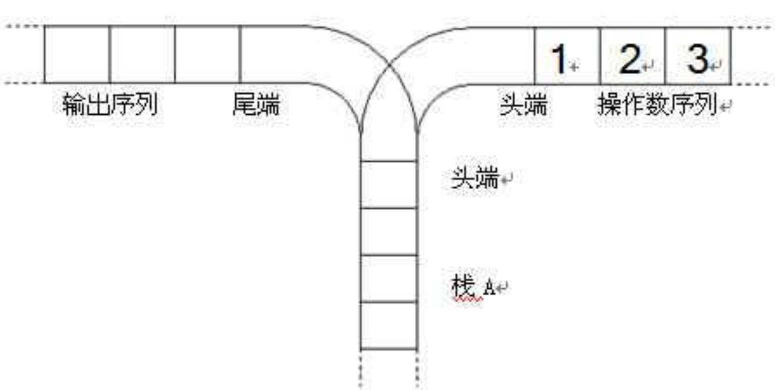
【问题背景】栈是计算机中经典的数据结构，简单的说，栈就是限制在一端进行插入删除操作的线性表。
栈有两种最重要的操作，即pop（从栈顶弹出一个元素）和push（将一个元素进栈）。



宁宁考虑的是这样一个问题：一个操作数序列，从1，2，一直到n（图示为1到3的情况），栈A的深度大于n。
现在可以进行两种操作，
1.将一个数，从操作数序列的头端移到栈的头端（对应数据结构栈的push操作）
2. 将一个数，从栈的头端移到输出序列的尾端（对应数据结构栈的pop操作）

使用这两种操作，由一个操作数序列就可以得到一系列的输出序列，下图所示为由1 2 3生成序列2 3 1的过程。（原始状态如上图所示）





3 1 2怎么生成？

你的程序将对给定的n，计算并输出由操作数序列1，2，…，n经过操作可能得到的输出序列的总数。

【输入格式】

输入文件只含一个整数n（1≤n≤18）

【输出格式】

输出文件只有一行，即可能输出序列的总数目

【输入样例】

3

【输出样例】

5

例题八：最长公共子序列

- 一个给定序列的子序列是在该序列中删去若干元素后得到的序列。
- 给定两个序列X和Y，当另一序列Z既是X的子序列又是Y的子序列时，称Z是序列X和Y的公共子序列。
- 最长公共子序列:公共子序列中长度最长的子序列。
- 最长公共子序列问题

  给定两个序列$X=\{x_1,x_2,\ldots,x_m\}$和$Y=\{y_1,y_2,\ldots, y_n\}$，找出X和Y的一个最长公共子序列。
- 例如: X = (A, B, C, B, D, A, B)

  X的子序列:
  - 所有X的子集(集合中元素按原来在X中的顺序排列)

    (A, B, D), (B, C, D, B), 等等.

例子

X = (A, B, C, B, D, A, B)　　X = (A, B, C, B, D, A, B)

Y = (B, D, C, A, B, A)　　Y = (B, D, C, A, B, A)

- (B, C, B, A)和(B, D, A, B)都是X和Y 的最长公共子序列(长度为4)

- 但是,(B, C, A)就不是X和Y的最长公共子序列

穷举法

- 对于每一个X_m的子序列,验证它是否是Y_n的子序列.
- X_m有2^m个子序列
- 每个子序列需要o(n)的时间来验证它是否是Y_n的子序列.
  - 从Y_n的第一个字母开始扫描下去,如果不是则从第二个开始
- 运行时间: o($n2^m$)

分析:

- 给定一个序列$X_m = (x_1, x_2, \ldots, x_m)$,我们定义$X_m$的第$i$个前缀为:

步骤1 $X_i = (x_1, x_2, \ldots, x_i)$ $i = 0, 1, 2, \ldots, m$

$c[i, j]$为序列$X_i = (x_1, x_2, \ldots, x_i)$和$Y_j = (y_1, y_2, \ldots, y_j)$的最长公共子序列的长度.

步骤2

最优子结构性质:

设序列$X_m=\{x_1,x_2,\ldots,x_m\}$和$Y_n=\{y_1,y_2,\ldots,y_n\}$的一个最长公共子序列为$Z_k=\{z_1,z_2,\ldots,z_k\}$，则

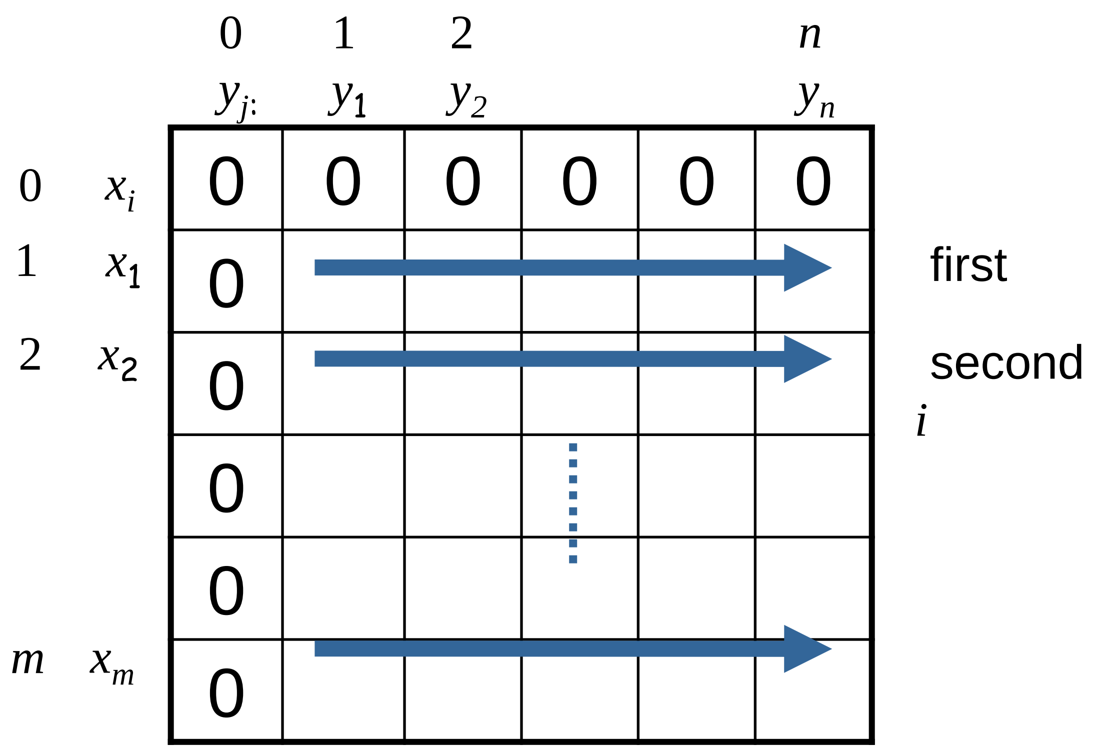
1.若$x_m=y_n$，则$z_k=x_m=y_n$，且Z_{k-1}是X_{m-1}和Y_{n-1}的最长公共子序列。

2.若$x_m\neq y_n$，且$z_k\neq x_m$，则Z_k是X_{m-1}和Y_n的最长公共子序列。

3.若$x_m\neq y_n$，且$z_k\neq y_n$，则Z_k是X_m和Y_{n-1}的最长公共子序列。

状态转移方程

步骤3



例题9：0-1背包问题

- 小偷有一个可承受W的背包
- 有n件物品: 第i个物品价值v_i 且重w_i
- 目标:
  - 找到x_i使得对于所有的$x_i = \{0, 1\}$, i = 1, 2, .., n

$$\sum w_i x_i \le W \text{ 并且} \sum x_i v_i \text{ 最大}$$

部分背包问题

最优子结构

- 考虑最多重W的物品且价值最高
- 如果我们把j物品从背包中拿出来

$\Rightarrow$ 剩下的装载一定是取自n-1个物品使得不超过载重量$W-w_j$并且所装物品价值最高的装载

0-1背包问题的动态规划

步骤1

***P*(*i*, *w*) –考虑前*i*件物品所能获得的最高价值,其中*w*是背包的承受力**

阶段分析：

对于每一个物品i，都有两种情况需要考虑

第1种情况:物品*i*的重量w_i<=*w*，小偷对物品*i*可拿或者不拿

$$P[i,w] = \max\{P[i-1, w], P[i-1,w-w_i] + v_i\}$$

第2种情况:物品*i*的重量w_i>*w*,即小偷不拿物品*i*

$$P(i, w) = P(i - 1, w)$$

步骤2

$$P(i,w)=\begin{cases} P(i-1,w) & \text{当wi> w （不够装不装）} \\ \max\begin{cases} P(i-1,w) & \text{够装但不装} \\ p(i-1,w-wi)+vi & \text{够装而且装} \end{cases} \end{cases}$$

$$F(i,w) = \max\{F(i-1, w-kw[i]) + kv[i]\}，k = 0,1$$

Knapsack (S,W)

for $w \leftarrow 0$ **to** $w_1 - 1$ **do** $P[1, w] \leftarrow 0$;
for $w \leftarrow w_1$ **to** W **do** $P[1, w] \leftarrow v_1$;
for $i \leftarrow 2$ **to** n **do**
    for $w \leftarrow 0$ **to** W **do**
        if $w_i > w$ **then**
            $P[i,w] \leftarrow P[i-1, w]$;
        else
            $P[i,w] \leftarrow \max\{P[i-1, w], P[i-1,w-w_i] + v_i\}$

运行时间: $\Theta(nW)$

拓展1：装箱问题 （vijos 1133）

有一个箱子容量为v(正整数，o≤v≤20000)，同时有n个物品(o≤n≤30)，每个物品有一个体积 (正整数)。要求从 n 个物品中，任取若干个装入箱内，使箱子的剩余空间为最小。

输入格式 Input Format

第一行，一个整数，表示箱子容量；
第二行，一个整数，表示有n个物品；
接下来n行，分别表示这n个物品的各自体积。

输出格式 Output Format

一个整数，表示箱子剩余空间。

样例输入 Sample Input

24
6
8
3
12
7
9
7

样例输出 Sample Output

0

拓展2：采药（vijos1104)

辰辰是个天资聪颖的孩子，他的梦想是成为世界上最伟大的医师。为此，他想拜附近最有威望的医师为师。医师为了判断他的资质，给他出了一个难题。医师把他带到一个到处都是草药的山洞里对他说："孩子，这个山洞里有一些不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。我会给你一段时间，在这段时间里，你可以采到一些草药。如果你是一个聪明的孩子，你应该可以让采到的草药的总价值最大。"

如果你是辰辰，你能完成这个任务吗？

输入格式 Input Format

输入的第一行有两个整数T（1 <= T <= 1000）和M（1 <= M <= 100），用一个空格隔开，T代表总共能够用来采药的时间，M代表山洞里的草药的数目。接下来的M行每行包括两个在1到100之间（包括1和100）的整数，分别表示采摘某株草药的时间和这株草药的价值。

输出格式 Output Format

输出包括一行，这一行只包含一个整数，表示在规定的时间内，可以采到的草药的最大总价值。

样例输入 Sample Input

70 3
71 100
69 1
1 2

样例输出 Sample Output

3

拓展3：开心的金明 （vijos 1317）

金明今天很开心，家里购置的新房就要领钥匙了，新房里有一间他自己专用的很宽敞的房间。更让他高兴的是，妈妈昨天对他说："你的房间需要购买哪些物品，怎么布置，你说了算，只要不超过N 元钱就行"。今天一早金明就开始做预算，但是他想买的东西太多了，肯定会超过妈妈限定的N 元。于是，他把每件物品规定了一个重要度，分为5 等：用整数1~5 表示，第5 等最重要。他还从因特网上查到了每件物品的价格（都是整数元）。他希望在不超过N 元（可以等于N 元）的前提下，使每件物品的价格与重要度的乘积的总和最大。设第j 件物品的价格为v[j]，重要度为w[j]，共选中了k 件物品，编号依次为j1...jk，则所求的总和为：v[j1]*w[j1]+..+v[jk]*w[jk]请你帮助金明设计一个满足要求的购物单.

输入格式 Input Format

输入的第1 行，为两个正整数，用一个空格隔开：
N m
（其中N（<30000）表示总钱数，m(<25)为希望购买物品的个数。）
从第2 行到第m+1 行，第j 行给出了编号为j-1
的物品的基本数据，每行有2 个非负整数
v p
（其中v 表示该物品的价格（v≤10000），p 表示该物品的重要度（1~5））

输出格式 Output Format

输出只有一个正整数，为不超过总钱数的物品的价格与重要度乘积的总和的最大值（<100000000）

样例输入 Sample Input

1000 5
800 2
400 5
300 5
400 3
200 2

样例输出 Sample Output

3900

例题10：完全背包

描述：有N种物品和一个体积为W的背包，每种物品都有无限件可用。第i种物品的体积是w[i]，价值是v[i]。求解将哪些物品装入背包可使这些物品的体积总和不超过背包容量，且价值总和最大。

$F(i,w)$表示前i种物品放入容量为w的背包中的最大价值，

$$F(i,w)=\max\{F(i-1,w-kw[i])+kv[i]\}，kw[i]\le \mathrm{w}$$

或者：

$$F(i,w)=max\begin{cases}F(i-1,w),\text{不放}\\F(i,w-w[i])+\mathrm{v}[i],\text{至少放一件}\end{cases}$$

例题：洛谷P1616，疯狂采药

可转化为01背包问题

例题11：多重背包

描述：有N种物品和一个容量为W的背包。第i种物品最多有p[i]件可用，每件体积是w[i]，价值是v[i]。求解将哪些物品装入背包可使这些物品的体积总和不超过背包容量，且价值总和最大。

$F(i,w)$表示前i种物品放入容量为w的背包中的最大价值，

$$F(i,w) = \max\{F(i-1, w-kw[i]) + kv[i]\}，0 \le k \le p[i]$$

例题：Luogu P1776 宝物筛选

可转化为01背包问题

例题12：混合背包

描述：01和完全背包混合，或者再加上多重背包

全部转化为01背包问题

例题：[Luogu P1833 樱花]

例题13：分组背包

描述：有N件物品和一个容量为W的背包。第i件物品的体积是w[i]，价值是v[i] 。这些物品被划分为若干组，每组中的物品互相冲突，最多选一件。求解将哪些物品装入背包可使这些物品的体积总和不超过背包容量，且价值总和最大。

$F(i,w)$表示前i组物品放入容量为w的背包中的最大价值，

$$F(i,w)=\max(F[i-1][w],f[i-1][w-w[k]]+v[k] \mid 物品k属于组i)$$

例题[Luogu 1757 通天之分组背包]

一组最多选一件

例题14：有依赖的背包

金明的预算方案

描述：金明今天很开心，家里购置的新房就要领钥匙了，新房里有一间金明自己专用的很宽敞的房间。更让他高兴的是，妈妈昨天对他说：“你的房间需要购买哪些物品，怎么布置，你说了算，只要不超过N元钱就行”。今天一早，金明就开始做预算了，他把想买的物品分为两类：主件与附件，附件是从属于某个主件的，下表就是一些主件与附件的例子：

主件 附件
电脑 打印机，扫描仪
书柜 图书
书桌 台灯，文具
工作椅 无

一组可以选多件

如果要买归类为附件的物品，必须先买该附件所属的主件。每个主件可以有0个、1个或2个附件。附件不再有从属于自己的附件。金明想买的东西很多，肯定会超过妈妈限定的N元。于是，他把每件物品规定了一个重要度，分为5等：用整数1~5表示，第5等最重要。他还从因特网上查到了每件物品的价格（都是10元的整数倍）。他希望在不超过N元（可以等于N元）的前提下，使每件物品的价格与重要度的乘积的总和最大。

设第j件物品的价格为v[j]，重要度为w[j]，共选中了k件物品，编号依次为j1，j2，……，jk，则所求的总和为：v[j1]*w[j1]+v[j2]*w[j2]+ …+v[jk]*w[jk]。（其中*为乘号）请你帮助金明设计一个满足要求的购物单。

输入格式 Input Format

输入文件的第1行，为两个正整数，用一个空格隔开：
N m
其中N（<32000）表示总钱数，m（<60）为希望购买物品的个数。）
从第2行到第m+1行，第j行给出了编号为j-1的物品的基本数据，每行有3个非负整数
v p q
（其中v表示该物品的价格（v<10000），p表示该物品的重要度（1~5），q表示该物品是主件还是附件。如果q=0，表示该物品为主件，如果q>0，表示该物品为附件，q是所属主件的编号）

输出格式 Output Format

输出文件只有一个正整数，为不超过总钱数的物品的价格与重要度乘积的总和的最大值
（<200000）。

样例输入 Sample Input

```
1000 5
800 2 0
400 5 1
300 5 1
400 3 0
500 2 0
```

样例输出 Sample Output

```
2200
```

例题15：二维费用背包

描述：对于每件物品，具有两种不同的费用；选择这件物品必须同时付出这两种代价；对于每种代价都有一个可付出的最大值（背包容量）。问怎样选择物品可以得到最大的价值。设第i件物品物品的价值为v[i]，所需的两种代价分别为w[i] 和g[i]。两种代价可付出的最大值（两种背包容量）分别为V和T。

设$F(i,w,g)$表示前i件物品付出两种代价分别为w和g时可获得的最大价值。

$$F(i,w,g) = max(f(i-1,w,g), f(i-1,w-w[i],w-g[i]) + v[i])$$

例题Luogu 1507 NASA的食物计划

继续题目类型整理