历届 NOIP 搜索算法全集

摘要：算法分析：如果采用贪心法，则先取价值最大的 10，消耗了容积 8，下面只能取容积为 4 的 .... [数据结构] time,price 数组分别用来存入时间和价值，count 来存入背包的价值。...

关键词：算法,数据结构

类别：专题技术

来源：牛档搜索（Niudown.COM）

历届 NOIP 搜索算法全集

转自：oifans

用动态规划来解背包问题

在历届 NOIP 竞赛中，有 4 道初赛题和 5 道复赛题均涉及到背包问题，所谓的背包问题，可以描述如下：一个小偷打劫一个保 险箱，发现柜子里有 N 类不同大小与价值的物品，但小偷只有一个容积为 M 的背包来装东西，背包问题就是要找出一个小偷选择所偷物品的组合，以使偷走的物品总 价值最大。

如有 4 件物品，容积分别为： 3 4 5 8

对应的价值分别为： 4 5 7 10

小偷背包的载重量为：12

则取编号为 1 2 3 的物品，得到最大价值为 16。

算法分析：如果采用贪心法，则先取价值最大的 10，消耗了容积 8，下面只能取容积为 4 的物品，得到价值 5，这样总价值是 15，这不是最优解，因此贪心法是不正确的。

采用穷举法，用一个 B 数组来表示取数的标记，当 B=0 时表示第 i 件物品不取，当 B=1 时表示第 i 件物品已取，初始化全部取 0，以下算法是从后面的物品开始取起，通过 B 数组的取值把 15 种取法全部穷举出来，价值 MAX 初始化为 0。

B[0] B[1] B[2] B[3] B[4]

0 0 0 0 0 {初始化}

0 0 0 0 1 {取第 4 件物品，容积为 8，不超，价值为 10，将 MAX 替换为 10}

0 0 0 1 0 {取物品 3，容积为 5，不超，价值为 7，不换}

0 0 0 1 1 {取物品 3、4，容积为 13，超}

0 0 1 0 0 {取物品 2，容积为 4，不超，价值为 5，不换}

0 0 1 0 1

0 0 1 1 0

0 0 1 1 1

……

0 1 1 1 0 {这是最佳方案}

0 1 1 1 1

1 0 0 0 0 {当 B〔0〕＝1 时停止，B〔0〕称为哨兵}

生成 B 数组中数据的方法如下：

```
fillchar(b,sizeof(b),0);
while b[0]=0 do
begin j:=n;
while b[j]=1 do dec(j);
b[j]:=1;
for i:=j+1 to n do
b:=0;
end;
```

小结：以上每件物品只能取 1 件，所以取法只有 0 和 1 两种情况，我们称之为 0、1 背包，算法的时间复杂度为 O（2N），在 1 秒内 N 只能做到 20。

例 1：选数（NOIP2002 初中组复赛第 2 题）

[问题描述]:已知 n 个整数 x1,x2,…,xn，以及一个整数 k（k＜n）。从 n 个整数中任选 k 个整数相加，可分别得到一系列的和。例如当 n=4，k＝3，4 个整数分别为 3，7，12，19 时，

可得全部的组合与它们的和为：
3＋7＋12=22 3＋7＋19＝29 7＋12＋19＝38 3＋12＋19＝34。
现在，要求你计算出和为素数共有多少种。
例如上例，只有一种的和为素数：3＋7＋19＝29。
[输入]：
键盘输入，格式为：
n , k （1<=n<=20，k＜n）
x1,x2，…,xn （1<=xi<=5000000）
n[输出]：
屏幕输出，格式为：
一个整数（满足条件的种数）。
[输入输出样例]:
输入：
4 3
3 7 12 19
输出：
1
[算法分析]:本题应用背包问题中取数的方法进行穷举，在取数的过程中，当 B 数组中有 K 个 1 的时候将对应的 K 个数相加，再判断是不是素数。
主要程序段如下：

```
readln(n,k); sum:=0;
for i:=1 to n do read(a);
fillchar(b,sizeof(b),0);
while b[0]=0 do
begin j:=n;
while b[j]=1 do dec(j);
b[j]:=1;
for i:=j+1 to n do b:=0;
m:=0;
for i:=1 to n do
if b=1 then m:=m+1; {统计 1 的个数}
if m=k then
begin  计算此种取数方法得到的和 S；
if S  是素数  then sum:=sum+1;
end;
end;
```

例 2：采药(NOIP2005 初中组复赛第 3 题)
【问题描述】
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？

【输入文件】

输 入文件 medic.in 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。

【输出文件】

输出文件 medic.out 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】

70 3

71 100

69 1

1 2

【样例输出】

3

【数据规模】

对于 30%的数据，M <= 10；

对于全部的数据，M <= 100。

【算法分析】本题如果采用上述方法来解，只能将 M 算到 20，而这里 M〈=100，所以只能拿 30%的分数，比较好的算法是采用动态规划，为了能说清算法，现重新举一个例子，若输入：

10 3

3 4

4 5

5 6

表示背包的容量是 10，有 3 种物品。用一个数组用来表示背包容量与其最大价值的关系，上例中设置一个数组 count,用下标表示容量，初始化为 0。然后按物品的顺序一一来统计此时的最大价值，每种药品对应各种背包容量时得到的最大价值为：

对于是第 i 件物品，背包容量为 j 时的最大价值 Cmax(j)=MAX(Cmaxj,Pi+余下空间的最大价值 Cmax(j-i 物品所占的空间))，如上例中，根据物品的不断增加，各容量背包得到的最大价值不断替换：

容量 1 2 3 4 5 6 7 8 9 10

价值

序号 0 0 0 0 0 0 0 0 0 0

1 0 0 4 4 4 4 4 4 4 4

2 0 0 4 5 5 5 9 9 9 9

3 0 0 4 5 6 6 9 10 11 11

[数据结构] time,price 数组分别用来存入时间和价值，count 来存入背包的价值。

```
var
time,price:array[1..100] of longint;
t:longint; i,m,j:integer;
```

```
count:array[0..1000] of longint;
begin
assign(input,'medic.in');
assign(output,'medic.out');
reset(input);
rewrite(output);
readln(t,m);
for i:=1 to m do
readln(time,price);
fillchar(count,sizeof(count),0);
for i:=1 to m do
for j:=t downto 1 do
begin
if (j>=time) and (price+count[j-time]>count[j]) then
count[j]:=price+count[j-time];
end; {j>=time 表示当前的容量能放入背包，price+count[j-time]>count[j]表示第 i 件物品的价值加上第 i 件物品对于背包容量为 j 时余下空间的最大价值大于当前背包容量为 j 时的最大价值}
writeln(count[t]);
close(input);
close(output);
end.
```

例 3：开心的金明（NOIP2006 初中组复赛 2 题）

题目较长，省略，本题与例 2 相比，只是有时要先将价值乘以一个数，其余一样，但要注意的是：本题 N 的范围是<=26，用穷举法、1 背包穷举法在 1 秒内只能过 10 个点中的 8 个点。

总结：采用动态规划的时间复杂度为 O(n*m)，范围比穷举法大多了，但也有弱点，当数据不是整数时，就不可使用；如果还要求出具体的取法，那也相当麻烦。