【题目描述】2.1 简单背包问题（Backpack）

有一个背包，可以放入的最大重量为S，现有N件物品，重量均为正整数，分别为W1，W2，W3，…，Wn，现从N块物品中挑选若干块，使得放入的背包的重量之和正好为S。若成功，则输出放入背包的物品重量，否则输出“Failed!”。

【输入格式】

第一行两个整数即S和N（S＜1 000，N＜32）。第二行为N个整数，即N件物品的重量。

【输出格式】

若成功（答案非唯一），则输出放入背包的物品，一个物品一行，否则输出“Failed!”。

【输入样例】

10 5

1 2 3 4 5

【输出样例】

1

4

5

|  |  |
| --- | --- |
|  | 简单背包问题无需使用动规思想来解决，放在此处只是为了与后面要讲到的0/1背包等算法做比较。 |

容易想到的方法是将物品逐个放入背包内试验，设布尔函数Bag（s，n）表示剩下的n个物品中装满剩下重量为s的背包，如果有解，返回1，否则返回0。

从取最后一个物品开始：

（1）取最后一个物品Wn，调用Bag（s，n）；

（2）如 Wn＝s，结束程序，输出结果；

（3）如 Wn＜s，且n＞1，则求Bag（s－Wn，n－1）；

（4）如 Wn＞s，且n＞1，删除Wn，从剩下n－1中继续找，即Bag（s，n－1）。

递归结束的条件为：

（1）Wn＝s　（正好放入的物品重量等于背包能装的重量）

（2）Wn≠s　 （无解）

（3）n≤0　　（再没有物品可试）

但实际上问题并不是这么简单，因为所选取并放入的物品Wn很可能导致无法获得正确结果。例如s＝10，物品重量分别为1，6，2，7，5，如果第一次选择Wn＝5放入背包后，则后面再怎么选择也不可能成功，正确的做法是排除Wn＝5，从Wn＝7开始才可能有正确答案，即7＋2＋1＝10。

因此Wn是否有效还要看后续的Bag（s－Wn，n－1）是否有解，如果无解，说明先前取的Wn不合适，就要放弃Wn，在剩余物品中重新开始挑选，即Bag（s，n－1）。

参考代码如下所示。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | //简单背包问题 — 递归算法  #include <bits/stdc++.h>  **using** **namespace** std**;**  int W**[**40**];** //各物品重量  int Bag**(**int s**,**int n**)** //s为剩余重量，n为剩余可选物品数  **{**  **if(**s**==**0**)** //如果正好装满  **return** 1**;**  **if(**s**<**0 **||** **(**s**>** 0 **&&** n**<**1**))** //如s<0或n<1则不能完成  **return** 0**;**  **if(**Bag**(**s**-**W**[**n**],**n**-**1**))** //从后往前装，装上W[n]后，若剩余物品仍有解  **{**  cout**<<**W**[**n**]<<**"\n"**;** //则装进第n个包，并输出  **return** 1**;**  **}**  **return** Bag**(**s**,**n**-**1**);** //如装了第n个后无解则删除，尝试第n－1个  **}**  int main**()**  **{**  int S**,**N**;**  scanf**(**"%d%d"**,&**S**,&**N**);**  **for(**int i**=**1**;** i**<=**N**;** **++**i**)**  scanf**(**"%d"**,&**W**[**i**]);**  **if(!**Bag**(**S**,**N**))**  printf**(**"Failed!\n"**);**  **return** 0**;**  **}** |

#### 枚举算法

该问题可以转化为某个物品取或不取的问题，以1代表取，0代表不取，枚举出所有的可能性，输出总重量＝S的组合即可。

参考程序如下所示。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | //普通背包问题 — 枚举算法  #include <bits/stdc++.h>  **using** **namespace** std**;**  int N**,**S**;**  int W**[**40**],**use**[**40**];**  int main**()**  **{**  cin**>>**S**>>**N**;**  **for(**int i**=**0**;** i**<**N**;** i**++)**  cin**>>**W**[**i**];**  **for(**int num**=**0**;** num**<(**1**<<**N**);** num**++)** //位运算法枚举2的N次方的组合  **{**  int sum**=**0**;**  memset**(**use**,**0**,sizeof(**use**));**  **for(**int i**=**0**;** i**<**N**;** i**++)**  **if((**1**<<**i**)** **&** num**)** //如果当前位是1  **{**  sum**+=**W**[**i**];**  use**[**i**]=**1**;**  **}**  **if(**sum**==**S**)** //打印方案  **{**  **for(**int i**=**0**;** i**<**N**;** i**++)**  **if(**use**[**i**])**  cout**<<**W**[**i**]<<**endl**;**  exit**(**0**);**  **}**  **}**  cout**<<**"Failed!\n"**;**  **return** 0**;**  **}** |