**算法训练 装箱问题**

时间限制：1.0s   内存限制：256.0MB

**问题描述**  
　　有一个箱子容量为V（正整数，0＜＝V＜＝20000），同时有n个物品（0＜n＜＝30），每个物品有一个体积（正整数）。  
　　要求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

　　第一行为一个整数，表示箱子容量；  
　　第二行为一个整数，表示有n个物品；  
　　接下来n行，每行一个整数表示这n个物品的各自体积。

输出格式

　　一个整数，表示箱子剩余空间。  
　　**样例**输入  
　　24  
　　6  
　　8  
　　3  
　　12  
　　7  
　　9  
　　7

样例输出

0

锦囊1

动态规划。

锦囊2

用F[i,j]表示前i个物品装j的容量是否可以，则F[i,j]=F[i-1,j-a[i]]||F[i-1,j]。其中a[i]表示第i个物品的体积。

本题的C++参考代码如下：

#include<iostream>

#include<cstdio>

#include<cstdlib>

#include<cstring>

using namespace std;

bool v[21000];

int main()

{

//memset(v,0,sizeof(v));

int i,j,m,n,a;

scanf("%d%d",&m,&n);

v[0]=1;

for(i=1;i<=n;i++)

{

scanf("%d",&a);

for(j=m;j>=a;j--)

v[j]=v[j]||v[j-a];

}

for(j=m;j>=0;j--)

{

if(v[j])

{

printf("%d\n",m-j);

return 0;

}

}

}

本题的C参考代码如下：

#include <stdio.h>

#include <string.h>

#include <malloc.h>

#define V 20001

#define N 30

int main()

{

int f[V];

int i,j;

int n,v;

int T[V];

memset(f,0,sizeof(f));

memset(T,0,sizeof(T));

scanf("%d%d",&n,&v);

for(i=1;i<=v;i++)

scanf("%d",&f[i]);

for(i=1;i<=v;i++)

{

for(j=n;j>=f[i];j--)

{

if(f[i]<=j&&f[i]+T[j-f[i]]>T[j])

T[j] = f[i]+T[j-f[i]];

}

}

printf("%d\n",n-T[n]);

return 0;

}

本题的Java参考代码如下：

import java.io.\*;

import java.util.\*;

public class Main {

private static int v[][]=new int[31][20001];

public static void main (String args[])throws Exception{

//Scanner sc=new Scanner(System.in);

BufferedReader bf=new BufferedReader(new InputStreamReader(System.in));

//int n=sc.nextInt();

//int m=sc.nextInt();

int n=Integer.parseInt(bf.readLine());

int m=Integer.parseInt(bf.readLine());

int a[]=new int[m+1];

for(int i=1;i<=m;i++)

a[i]=Integer.parseInt(bf.readLine());

//a[i]=sc.nextInt();

for(int i=1;i<=m;i++){

for(int j=1;j<=n;j++){

if(j<a[i])

v[i][j]=v[i-1][j];

else

v[i][j]=Math.max(v[i-1][j], v[i-1][j-a[i]]+a[i]);

}

}

System.out.println(n-v[m][n]);

}

}