1102 采药 2005年NOIP全国联赛普及组

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？

输入描述 Input Description

输入第一行有两个整数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。

输出描述 Output Description

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。

定义数组 int f[101][1001] f[i][t]存储采第i个药并且用时恰好为t，取得的最大价值

定义数组 int t[i], v[i] 分别存储第i个药采摘所需的时间和价值

初始化：

#include <bits/stdc++.h>

using namespace std;

int a[10000],b[10000],f[10000];

int main() {

int t,n;

cin>>t>>n;

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

cin>>a[i]>>b[i];

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

for(int j=t; j>=a[i]; j--)

f[j]=max(f[j-a[i]]+b[i],f[j]);

cout<<f[t];

}

M—药的数目，T规定时限

f[0][0]=0;

for(int i=1;i<=T;i++) f[0][i]=-1 初始化第零阶段

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

for(int j=0;j<=T;j++)

{ int temp;//计算f[i][j]

temp=j-t[i];

if(temp<0 ) {f[i][j]=f[i-1][j];continue;}

if(f[i-1][temp]==-1) {f[i][j]=f[i-1][j];continue;}

if(f[i-1][j]==-1) {f[i][j]=f[i-1][temp]+v[i]};continue;}

f[i][j]= f[i-1][j]>f[i-1][temp]+vi? f[i-1][j]:f[i-1][temp]+v[i];

}

答案在f[M][0,…,T]中找

2102 石子归并 2

在一个园形操场的四周摆放N堆石子,现要将石子有次序地合并成一堆.规定每次只能选相邻的2堆合并成新的一堆，并将新的一堆的石子数，记为该次合并的得分。

试设计出1个算法,计算出将N堆石子合并成1堆的最小得分和最大得分.

输入描述 Input Description

数据的第1行试正整数N,1≤N≤100,表示有N堆石子.第2行有N个数,分别表示每堆石子的个数.

输出描述 Output Description

输出共2行,第1行为最小得分,第2行为最大得分.

#include<iostream>

#include<cstrng>

using namespace std;

int stone[100],a[100];

int f[100][100],g[100][100];

int n;

int main()

{ cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

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

memset(g,0,sizeof(g));

for(int i=0;i<n;i++) g[1][i]=a[i];

for(int i=2;i<=n;i++)

for(int j=0;j<n;j++)

g[i][j]=g[i-1][j]+a[(i+j-1)%n];//从j堆开始，连续i堆的和

for(int i=2;i<=n;i++) //阶段i

for(int j=0;j<n;j++)//起始位置

{ int min=0x7fffffff;

int temp;

for(int k=1;k<=i-1;k++)

{ temp=f[k][j]+f[i-k][(j+k)%n]+g[i][j];

min=temp<min?temp:min;

}

f[i][j]=min;

}

int min=0x7fffffff;

for(int i=0;i<n;i++)

min=f[n][i]<min?f[n][i]:min;

cout<<min<<endl;

//-------------------

for(int i=2;i<=n;i++) //阶段i

for(int j=0;j<n;j++)//起始位置

{ int max=-1;

int temp;

for(int k=1;k<=i-1;k++)

{ temp=f[k][j]+f[i-k][(j+k)%n]+g[i][j];

max=temp>max?temp:max;

}

f[i][j]=max;

}

int max=-1;

for(int i=0;i<n;i++)

max=f[n][i]>max?f[n][i]:max;

cout<<max<<endl;

}