可能题：投资问题

g[i][j]代表第i个工程被分配j资源对应的利润表，

换句说g数组就是保存每个工程对应j资源的利润表

f[i][j]储存的数值代表前i项工程分配了j资源的最大利润值

d[i][j]储存的是在前i项工程被分配j资源的最大利润已经求出的情况下对应的第i项的资源分配额度。

k[i]代表第i个工程被分配的资源额度，

也就是说k数组保存最优解对应的每项工程的分配额度

Invest(m,n,f[n][m],g[n][m])

{ for(j=0;j<=m;j++)

{ f[1][j]=g[1][j];d[1][j]=j; }

上面这行很重要，f[1][j]=g[1][j]是先对第1个工程的所有分配额度进行保存最大利润情况，因为是第一个工程不涉及其他工程，所以分配多少资源，考虑第一个工程的最大利润都是对应的资源分配利润表g[1][j]，

d[1][j]=j也是一个道理，保存第一个工程被分配j资源所有最大利润情况下的 分配额。这两个步骤都是对第一个工程的情况进行预处理，因为是自底向上求解，所以目的是为了给后面的递推公式提供数值计算。

for(i=2;i<=n;i++) 这行开始进行第2项到第n项的所有情况演算了

for(j=0;j<=m;j++)这行是进行所有分配额的情况演算

{ f[i][j]=0;这行目的是对项目的最大利润进行初始化为0

方便计算，如果不初始化，那么项目初始的最大利润没法进行比较会出现问题，详细看下面

for(k=0;k<=j;k++) 变量K的作用是动态变换所有分配组合 { s=f[i-1][j-k]+g[i][k];

上面这行变量s就是保存计算将前i-1项被分配j-k最大利润与当前分配的利润相加组合，k值的作用就是保存所有分配组合都发生

if(s>f[i][j]) { f[i][j]=s; d[i][j]=k; }

将每一次组合情况的最大利润都与f[i][j]进行比较，利润大的保留下来，进行下次比较，并且保存当前第i项工程的分配额度，重点是由于f[i][j]是被初始化为0，所以第一次会直接保存第一种s情况，然后在对所有情况比较留下最优组合，如果不初始化，数组值是垃圾数值，会出现无限大，那就无法有效进行计算，这就是前面初始化的原因所在

}

}

到这里每次递增算工程的时候，前i项工程的最优解已被算出，重复这个循环过程知道i达到n，那么总体最优解就算好了，只不过这个最优解的组合情况为被提取出来，只是保存在d数组中需要下面的往回寻找来筛选出来。

}

s=m; k[n]=d[n][m]；之前的s是保存最大利润，这里变量复用，赋值工程总资源，k[n]保存计算到最后一个工程的最优分配额，这个应该毋庸置疑的，一直算到第n个工程资源m被分配完时候这最后一项工程n的分配额算完了，然后结束了，不会因为下一项工程的组合分配造成d数组内数值的变动

for(i=n-1；i>0；i--)

{

s = s-k[i+1]；将m总资源数-最后一项被分配的资源数，留下前n-1项资源数，以此循环直到减到最后一组

k[i] = d[i][s]；对k[n-1]保存第n-1项目的最优资源分配额，关键在于这个s,s代表前n项被最优解被分配的总资源数，不知道这个无法筛选出那时候当前项目的分配额度，

}通过这样的操作，将最优解的组合分配额度，完全复制到k数组中去了

output k[i]；输出每个项目最优情况下被分配的额度，算法解析完毕！

第二题：N皇后问题

x数组保存的是第i个皇后所在i行的列的位置

k变量控制第几个皇后

Nqueen( )

{ x[1]=1; k=1;

do { if(x[k]<=8)//规定的是8\*8棋盘，所以列数最大上限为8

{ die=0//初始化为可放置皇后

for (i=1; i<k; i++)//开始放置第一个皇后

{ if (x[i]=x[k]) { die=1; break }

//上面是判断寻找中的第k行皇后与前面k行已找到的皇后列数未知进行比对，查询是否出现同列，如果同列则不能放置皇后，判断死亡

if (x[k]-x[i])=abs(k-i))

//上面是判断两个皇后是否在同一斜线上，由于斜率为1，所以横坐标差值==纵坐标差值，如果在斜线上则判断死亡

{die=1; break;}

}

if(die=1) x[k]=x[k]+1;//如果死亡就递增移到下一列位置继续循环判断

else{k=k+1; x[k]=1}//没死亡则找到此皇后的位置，递增k值开始找下一个皇后，并初始化的它起始列的位置

}

else { x[k]=1; k=k-1; x[k]=x[k]+1; }

//上面是如果皇后的列数位置一直搜索到8都没找到则进行回溯回滚至前一个皇后进行移动查找另一种可行情况

} while(k<=8 and k>0)//一个可行解找到的结束条件

if(k>8) output x[i]..x[8] 输出每个行皇后的列位置

else output no solve；否则输出无解

}

图M着色一样的