#### 改进算法★★

解决该题方法有多种，一般对付这种小数据范围的题，最容易想到的是穷举的方法。

我们分三步走：

（１）选出K种面值的邮票；

（２）在（１）得到的邮票中，枚举每种情况中每种邮票的张数，求得其能构成的所有面值；

（３）查找它最大连续的面值，即MAX。

对于（１）：其本质就是组合。我们先根据数据范围估计一张邮票最大的面值不超过55。该步骤相当于55选K的组合问题。

对于（２）：递归搜索各邮票可取的张数，构造出所有可取的面值，存入哈希表。

对于（３）：步骤（２）已经得到一种组合所能构成的所有面值，循环一遍便可得到这种组合的MAX，比较是否最优，存入答案。

　　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  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81 | //邮票面值问题──改进  #include <stdio.h>  #include <stdlib.h>  FILE \*in=fopen("Stamp.in","r"),\*out=fopen("Stamp.out","w");    #define M 50 // N+K≤40  int N,K;  int k[M],n[M]; //k储存选中的面值组合，n[i]表示选定的k[i]面值的张数  bool line[300]; //line[i]表示能否得到面值i，能为1，不能为0，估计MAX不超过300  int ANS; //最优MAX  int ans[M]; //储存最优组合  void finish()//步骤（３）计算MAX，并判断是否最优解  {  int i,MAX;  for(i=1;line[i];i++)  line[i]=0;  MAX=i-1; //记录当前能取得的MAX  for(;i<300;i++)  line[i]=0; //初始化line数组    if(MAX>ANS) //如果当前MAX为最优，储存之  {  ANS=MAX;  for(i=2;i<=K;i++)  ans[i]=k[i];  }  }  void toline()  {  int i,s=0;  for(i=1;i<=K;i++)  s+=k[i]\*n[i];  //计算当前构造出的面值  if(s<300) //防止爆出line，干扰其他数据  line[s]=1;  }  void nn(int dep,int s)  {//步骤（２）穷举张数 n[dep]:面值k[dep]的邮票的张数;s:剩余可贴的张数  int i;  for(i=0;i<=s;i++) //k[dep]可取0到s张  {  n[dep]=i;  if(s-i<0)  return;  else if(dep==K)  toline(); //将当前拼出的面值记入line中  else if(dep<K)  nn(dep+1,s-i);  }  }  void kk(int dep)//步骤（１）.面值组合,55选K  {  int i;  for(i=k[dep-1]+1;i<=55-K+dep;i++)  {  k[dep]=i;  if(dep==K) //完成步骤（１），确定一种组合  {  nn(1,N); //b.穷举选中的各面值邮票的张数  finish(); //c.确定该组合的MAX，并判断该面值组合是否最佳  }  else  kk(dep+1);  }  }  int main()  {  fscanf(in,"%d %d",&N,&K);  k[1]=1;ans[1]=1;  kk(2); //k[1]必为1，从2开始递归组合  for(int i=1;i<=K;i++)//输出  fprintf(out,"%d ",ans[i]);  fprintf(out,"%d\n",ANS);  return 0;  } |