【算法分析】

这里的取模运算无论是正数还是负数，对10取模的结果均为非负值。在解法上首先破环为链，如图16.2所示：

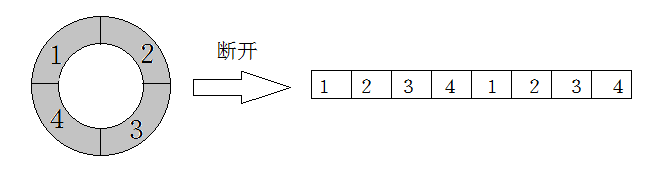


图16.2

然后应用动态规划，设 sum[i]表示前i个整数的和，设f[i][j]表示前i个数分成j个部分所能取得的最大值，设g[i][j]表示前i个数分成j个部分所能取得的最小值。则其状态转移方程为：

f[i][j]＝max(f[i][j]，f[k][j－1]×(((sum[i]－sum[k])%10＋10)%10))

g[i][j]＝min(g[i][j]，g[k][j－1]×(((sum[i]－sum[k])%10＋10)%10))

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 | //模拟人生  #include <bits/stdc++.h>  using namespace std;  int Line[200],n,m,sum[200],Max,Min;  int f[200][20];  int g[200][20];  void dp(int a[])  {  for(int i=1; i<=n; i++) //前序和  sum[i]=sum[i-1]+a[i];  for(int i=0; i<=n; i++) //初始化  for(int j=0; j<=m; j++)  {  f[i][j]=0;  g[i][j]=-1u>>1;  }  for(int i=1; i<=n; i++)  f[i][1]=g[i][1]=(sum[i]%10+10)%10;  f[0][0]=1;  g[0][0]=1;  for(int j=2; j<=m; j++)  for(int i=j; i<=n; i++)  for(int k=j-1; k<i; k++)  {  f[i][j]=max(f[i][j],f[k][j-1]\*(((sum[i]-sum[k])%10+10)%10));  g[i][j]=min(g[i][j],g[k][j-1]\*(((sum[i]-sum[k])%10+10)%10));  }  Max=max(Max,f[n][m]);  Min=min(Min,g[n][m]);  }  int main()  {  Max=0;  Min=-1u>>1; //即2147483647  scanf("%d%d",&n,&m);  for(int i=1; i<=n; i++)  {  scanf("%d",&Line[i]);  Line[i+n]=Line[i]; //数组扩展一倍  }  for(int i=0; i<n; i++)  dp(Line+i);  printf("%d\n%d\n",Min,Max);  return 0;  } |