设M[i][j]表示从第 i 个矩阵到第 j 个矩阵连乘的最少乘法次数，（i 从 0 编号），最终目标是求 M[0][n－1]。

例如要得到Ai×…Ak×Ak+1…Aj这个式子的值（即从矩阵 i 连乘到矩阵 j），假设最后一个矩阵乘法是在矩阵Ａk后断开的（即左右都已连乘运算好了），那么容易得到其递推式：

M[i][j]＝min{﻿M[i][k]＋M[k+1][j]＋di×dk+1×dj+1} （i≤k≤j－1）

其中 di 是矩阵Ai的第一维，dk+1是断开处矩阵Ak的第二维（即Ak+1的第一维，是一样的），dj+1是最后一个矩阵Aj的第二维。

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 | //矩阵连乘  #include <bits/stdc++.h>  using namespace std;  const int MAXN=110;    int F[MAXN][MAXN];  int R[MAXN],C[MAXN],N; //将维度分开存入两个数组，左和右数组  int main()  {  scanf("%d",&N);  for(int i=1; i<=N; ++i) //此处i从1编号  scanf("%d%d",&R[i],&C[i]);  for(int l=2; l<=N; ++l) //l确定几个矩阵合并，为2时，是2个矩阵合并  for(int i=1; i<=N-l+1; ++i) //求某个l值下第i行的格子的值  {  int j =i+l-1; //确定右边的界限  F[i][j]=0x7fffffff;  for(int k=i; k<j; ++k) //从i至j的l个矩阵，在k位置分隔成两部分  F[i][j]=min (F[i][k]+F[k+1][j]+R[i]\*C[k]\*C[j],F[i][j]);  }  printf ("%d\n",F[1][N]);  return 0;  } |