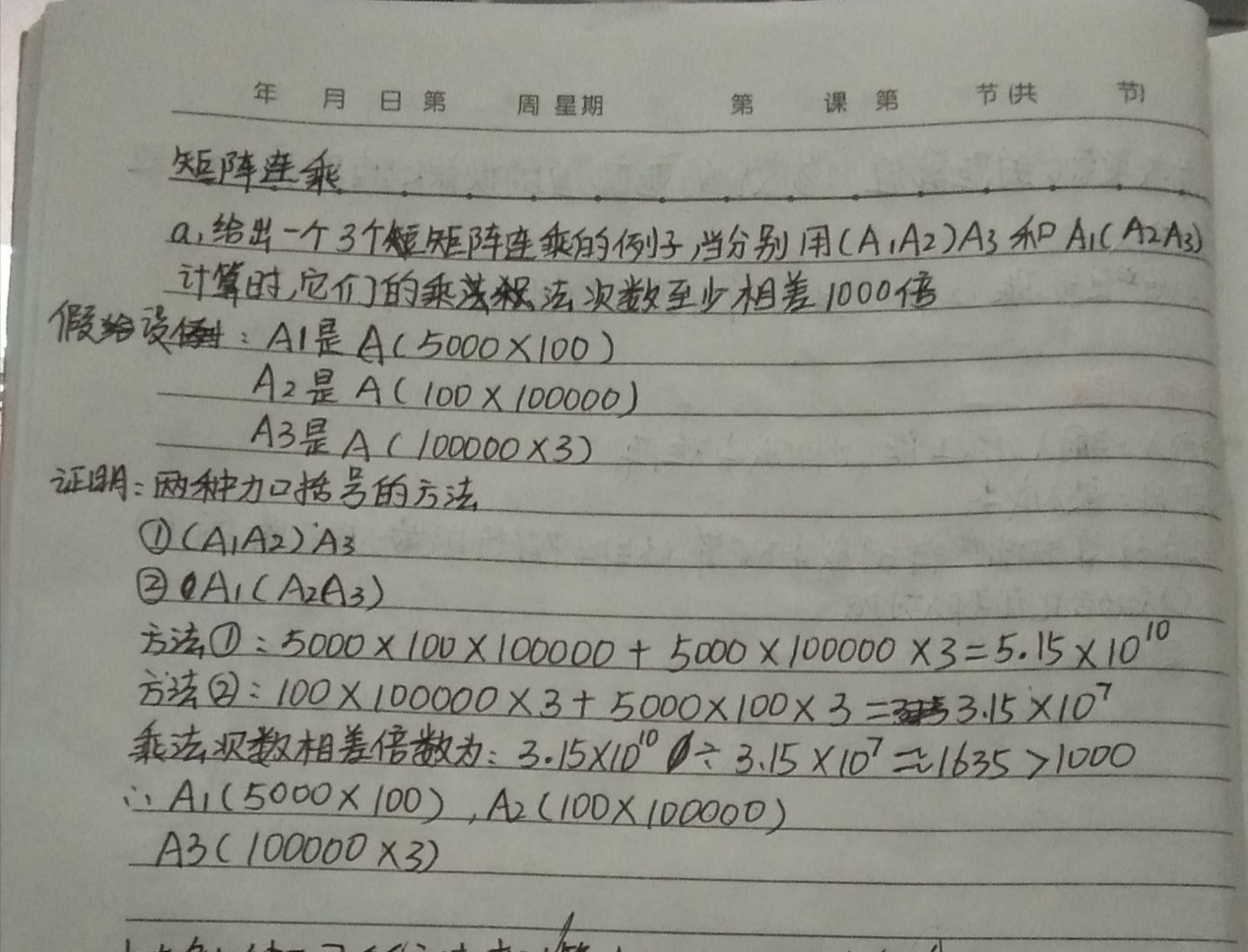
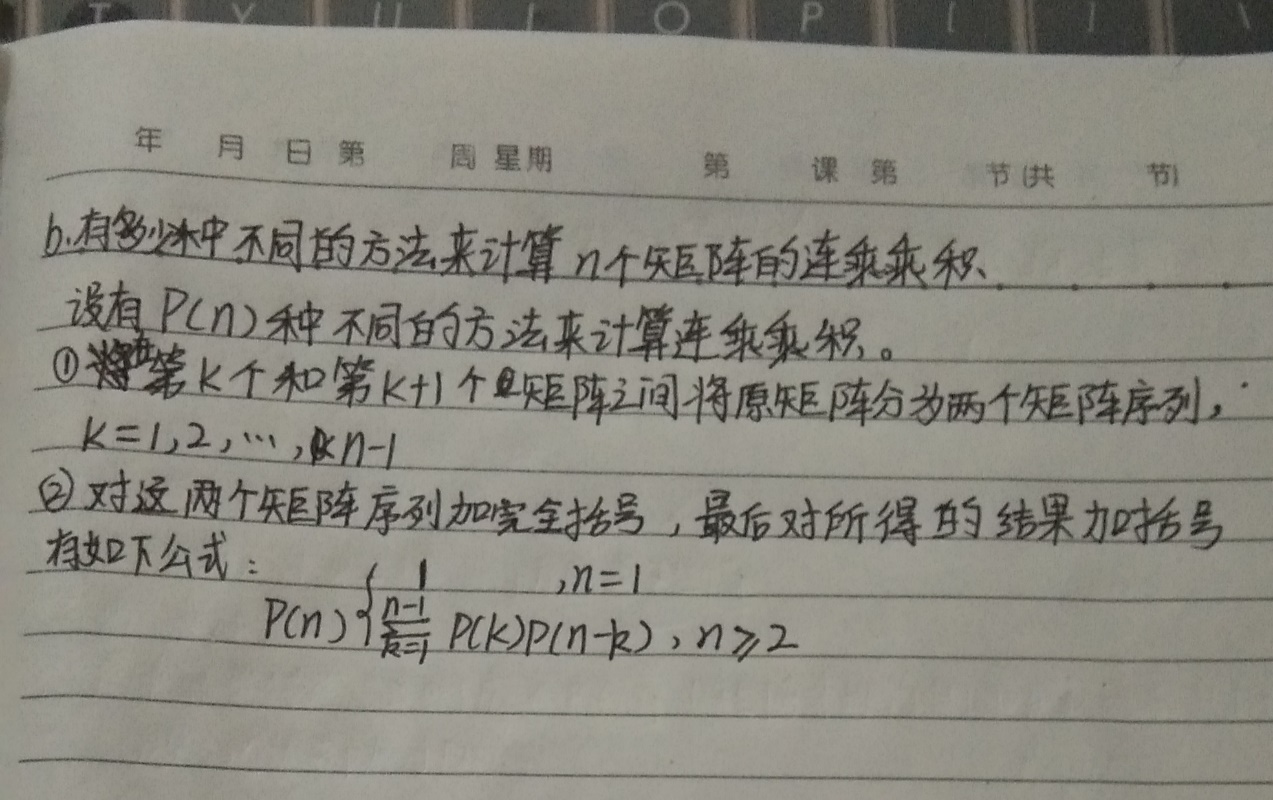
# 矩阵连乘

1. 给出一个三个矩阵连乘的例子。分别用（A1A2）A3、A1(A2A3)计算时，他们的乘法次数至少相差1000倍。



1. 有多少种不同的方法来计算n个矩阵的连乘乘积？



1. 设计一个求n个矩阵乘法最优次数的动态规划算法。

#include <stdio.h>

#define MAX 10000

int p[MAX+1];

int m[MAX][MAX];//矩阵连乘的最优解解

int s[MAX][MAX];//对应m[i][j]的断开位置k记为s[i][j]

int n;//矩阵的个数

void juzhen(){

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

m[i][i] = 0; //当i=j时，证明只有一个矩阵，不涉及相乘问题，所以乘积为0

for(int z=2; z<=n; z++) //当计算矩阵的长度大于等于2时,可以进行矩阵连乘

{

for(int i=1; i<=n-z+1; i++)//n-z+1为最后一个z的前边界

{

int j = i+z-1;//计算前边界为z，长为z的的后边界

m[i][j] = m[i+1][j] + p[i-1]\*p[i]\*p[j];

//将ij划分为A(i) \* ( A[i+1:j] )

s[i][j] = i;

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

{

//将ij划分为( A[i:k] )\* (A[k+1:j])

int t = m[i][k] + m[k+1][j] + p[i-1]\*p[k]\*p[j];//i<j

if (t < m[i][j])

{

m[i][j] = t;

s[i][j] = k;

}

}

}

}

}

void cixu(int i, int j)

{//矩阵的计算次序为

if(i==j) //i=j证明只有一个矩阵

printf("A%d",i);

else{

printf("(");

cixu(i,s[i][j]);

cixu(s[i][j]+1,j);

printf(")");

}

}

int main(){

int i;

printf("请输入矩阵相乘的矩阵个数；\n");

scanf("%d",&n);

printf("请依次输入相邻矩阵的行数和列数：\n") ;

for(i=0;i<n+1;i++)

{

scanf("%d",&p[i]);

}

juzhen();

printf("矩阵连乘最小次数为:\n%d\n",m[1][n]);

printf("矩阵的计算次序为：\n");

cixu(1,n);

printf("\n");

return 0;

}