**最小代价子母树**

#include "iostream"

#include "cstdio"

#include "queue"

#define qwq 21000000

int n,m,q;

using namespace std;

int f[1000][1000];

int g[1000][1000];

int gra[1000],minn=-2100000;

int main()

{

int n;

cin>>n;

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

cin>>gra[i];

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

g[i][i]=gra[i];

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

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

g[i][j]=g[i][j-1]+gra[j];

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

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

f[i][j]=(i!=j)?qwq:0; //i -> j(i)的代价为 +∞(0)

for (int p=2;p<=n;p++)//穷举堆数//因为是递推(Recursion)所以从小堆数向大堆数

for (int i=1;i<=(n-p+1);i++)//一段的确定性(Certainty) 列举起点

{

int j=i+p-1; //终点(Destination)

for (int k=i;k<=j-1;k++)//穷举隔开位置 //注意超界所致的溢出(Overflow)

f[i][j]=(f[i][j]>f[i][k]+f[k+1][j]+g[i][j])?f[i][k]+f[k+1][j]+g[i][j]:f[i][j];//三目运算符不清楚的可以百度，下面也有简介

} //正确写出状态转移方程//无后效性 (Unfollow-up Effect)

cout<<f[1][n]<<endl;

}

**1686: 取数**

时间限制: 0 Sec  内存限制: 128 MB  
提交: 35  解决: 7  
[[提交](http://10.5.54.252/oj/submitpage.php?id=1686)][[状态](http://10.5.54.252/oj/problemstatus.php?id=1686)][[讨论版](http://10.5.54.252/oj/bbs.php?pid=1686)]

**题目描述**

给出长度为N的数列{A\_i}，每次可以从最左边或者最右边取走一个数，第i次取数得到的价值是i  \*  A\_j。求价值之和最大的取数方案。 N  < =  2000  ,  A\_i  < =  1000

**输入**

        第一行，一个整数，表示数列长度N。         接下来N行，每行一个整数，表示数列A\_i。

**输出**

一个整数，表示最大的价值之和。

**样例输入**

5

1

3

1

5

2

**样例输出**

43

**提示**

考虑第i次取数时最右边（或者最左边）取了j个数的最大值

**来源**

#include<stdio.h>

#include<string.h>

int a[2002];

int max(int a,int b)

{

    if(a>b)

    return a;

    else

    return b;

}

int main()

{

    int n,i,m=-100000000,j;

    while(scanf("%d",&n)!=EOF)

    {

        for(i=1;i<=n;i++)

        scanf("%d",&a[i]);

        int b[2002]={0};

        b[0]=a[1];b[1]=a[n];

        for(i=2;i<=n;i++)

        {

            for(j=i;j>=0;j--)

            {

                if(j>0)

                b[j]=max(b[j-1]+i\*a[n-j+1],b[j]+i\*a[i-j]);

                else

                b[j]+=i\*a[i];

              //  printf("%d ",b[j]);

            }

        }

        for(i=0;i<=n;i++)

        if(b[i]>m)

        m=b[i];

        printf("%d\n",m);

    }

}

**能量项链：**

#include <iostream>

using namespace std;

int n, e=0;

int a[111\*2];

int dp[111\*2][111\*2];

int main()

{

cin >> n;

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

{

cin >> a[i];

a[i+n] = a[i];//用数组模拟环，减少一层循环

}

for(int j=2; j<=n+n; j++)

{

for(int i=j-1; i>=1&&j-i<n; i--)

{

int maxV = 0;

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

{

if(maxV < dp[i][k]+dp[k+1][j]+a[i]\*a[k+1]\*a[j+1])

maxV = dp[i][k]+dp[k+1][j]+a[i]\*a[k+1]\*a[j+1];

}

dp[i][j] = maxV;

if(e < dp[i][j]) e = dp[i][j];

}

}

cout << e;

return 0;

}

**石子归并：**

#include <iostream>

using namespace std;

int sum[111] = {0};

int n,minV;

int w[111];

int dp[111][111] = {0};

int main()

{

cin >> n;

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

{

cin >> w[i];

sum[i] = sum[i-1] + w[i];

}

for(int len=2; len<=n; len++)

{

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

{

int j=i+len-1;

minV = 0x7fffffff;

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

{

if(minV > dp[i][k]+dp[k+1][j]+sum[j]-sum[i-1])

minV = dp[i][k]+dp[k+1][j]+sum[j]-sum[i-1];

}

dp[i][j] = minV;

}

}

cout << dp[1][n];

return 0;

}