零一编程[29059](https://oj.01class.cn/problem/29059)

解法1：按题目说的做，画出螺旋矩阵，然后取出数字。

#include<bits/stdc++.h>

using namespace std;

int main(){

int N,I,J;

cin>>N>>I>>J;

int spiral[N][N];

memset(spiral,0,sizeof(spiral));

int i=0,j=0;

int dir=0;

int cnt=1;

bool done=0;

while(!done){

spiral[i][j]=cnt;

switch(dir){

case 0:

if(j+1<N&&spiral[i][j+1]==0){

j++;

cnt++;

}

else if((i+1>=N||spiral[i+1][j])&&(j+1>=N||spiral[i][j+1])&&(i<1||spiral[i-1][j])&&(j<1||spiral[i][j-1])){

done=1;

}else{

dir++;

dir%=4;

}

break;

case 1:

if(i+1<N&&spiral[i+1][j]==0){

i++;

cnt++;

}

else if((i+1>=N||spiral[i+1][j])&&(j+1>=N||spiral[i][j+1])&&(i<1||spiral[i-1][j])&&(j<1||spiral[i][j-1])){

done=1;

}else{

dir++;

dir%=4;

}

break;

case 2:

if(i-1<N&&spiral[i-1][j]==0){

i--;

cnt++;

}

else if((i+1>=N||spiral[i+1][j])&&(j+1>=N||spiral[i][j+1])&&(i<1||spiral[i-1][j])&&(j<1||spiral[i][j-1])){

done=1;

}else{

dir++;

dir%=4;

}

break;

case 3:

if(j-1<N&&spiral[i][j-1]==0){

j--;

cnt++;

}

else if((i+1>=N||spiral[i+1][j])&&(j+1>=N||spiral[i][j+1])&&(i<1||spiral[i-1][j])&&(j<1||spiral[i][j-1])){

done=1;

}else{

dir++;

dir%=4;

}

break;

default:

break;

}

}

cout<<spiral[I-1][J-1]<<endl;

}

这个解法在数据量不大时没有问题，可要是数据两大一点就会超出256MB的内存限制。

解法2：将i、j放到离I，J最近的位置，然后再判断并计算。

（老师的代码）

#include<bits/stdc++.h>

using namespace std;

int main (){

int n,a,b;

cin>>n;

cin>>a;

cin>>b;

int ans=1;

while(a!=n && b!=n && a>1 && b>1){

a--;b--;

ans+=(4\*(n-1));

n-=2;

}

if(a==1)

ans+=b-1;

else if(a>1 && a!=n){

if(b==1)

ans+=(4\*n-5-a+2);

else

ans+=(n+a-2);

}

else

ans+=(3\*n-2-b);

cout<<ans<<endl;

return 0;

}

这个解法的if和else部分比较难写，但是更加简洁，可以直接计算，不用循环。

解法3：递归。

#include<bits/stdc++.h>

using namespace std;

int shift=0;

int getfromspiral(int N, int I, int J, int i=0, int j=0){

if(I-1==0){

return shift+J;

}

if(J-1==0){

if(I-1!=i){

return shift+3\*N+N-I-2;

}

return shift+1;

}

if(I==N)

return shift+3\*N-2-J+1;

if(J==N)

return shift+N+I-1;

shift+=N\*4-4;

return getfromspiral(N-2,--I,--J,++i,++j);

}

int main(){

int N,I,J;

cin>>N>>I>>J;

cout<<getfromspiral(N,I,J);

}

这种解法代码最短，但因是递归，比较难理解。