N皇后算法

#include<fstream>

#include<iostream>

std::ofstream fout("queenoo.txt");

/\*\*//\* 记录当前的放置方案 \*/

int \*x;

/\*\*//\* 皇后的个数N 和 方案数目 \*/

int n,sum=0;

/\*\*//\* 检查参数所指示的这一行皇后放置方案是否满足要求 \*/

int Place(int);

/\*\*//\* 递归方法求取皇后放置方案\*/

void Queen1(void);

/\*\*//\* 用户递归求取皇后放置方案的递归方法 \*/

void TraceBack(int);

/\*\*//\* 打印当前成功的放置方案 \*/

void PrintMethod(void);

int main()

{

using namespace std;

long start,stop;

cout<<"input n 输入皇后个数 :";

cin>>n;

x=(int \*)malloc(sizeof(int)\*n);

time(&start);/\*\*//\*记录开始时间\*/

Queen1();

time(&stop);/\*\*//\*记录结束时间\*/

cout<<"一共的方案数为："<<sum<<"\n";

cout<<"共花时间："<<(int(stop-start))<<"\n";

fout<< "一共的方案数为："<<sum<<"\n";

fout<< cout<<"共花时间："<<((stop-start))<<"\n";

}

int Place(int r)

{

int i;

for(i=0;i<r;i++){

if(x[r]==x[i] || abs(r-i)==abs(x[r]-x[i]))

return 0;

}

return 1;

}

void TraceBack(int r)

{

int i;

if(r>=n){

PrintMethod();

sum++;

/\*\*//\* PrintMethod(); \*/

}else{

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

x[r]=i;

if(Place(r)) TraceBack(r+1);

}

}

}

void PrintMethod(void)

{

int i,j;

std::cout<<"第"<<sum<<"个方案\n";

fout<<"第"<<sum<<"个方案\n";

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

for(j=0;j<n;j++){

if(j==x[i]) std::cout<<"1",fout<<"1";

else std::cout<<"0",fout<<"0";

}

std::cout<<"\n";

fout<<"\n";

}

}

void Queen1(void)

{

TraceBack(0);

}