201533661 이승수’s Algorithm homework#18:N-Queens Problem due-date:2016.12.04

<code>

#include <stdio.h>

#include <stdlib.h>

typedef enum boolean\_ {

false = 0,

true = 1

} boolean;

FILE \*inF, \*outF;

int N = 5;

int rev[15] = { 0, };//store the column position of each index(row)

int count = 0;//count number of cases

void solve(int ord);

void main()

{

int n[15];

int num=0;

inF = fopen("input.txt","r");

while (fscanf(inF, "%d", &n[num]))

{

if (n[num] == 0)

break;

num++;

}

fclose(inF);

outF = fopen("output.txt", "w");

for (int i = 0; i < num; i++)

{

N = n[i];

printf("put N(board size,queen number):%d\n",N);

solve(0);

printf("count: %d\n", count);

fprintf(outF,"%d\n",count);

count = 0;

}

fclose(outF);

}

void solve(int ord)

{

int i, j;

boolean field[30] = { false };//store state of each rows whether queen is positioned

if (ord >= N) //if one case is made, print it,count++(we can get faster result if we don't print the route)

{

//printf("solve: ");

for (j = 0; j < N; j++)

{

//printf("%d ", rev[j]);

if (rev[j] >= N)

{

//printf("fail\n");

exit(1);

}

}

//printf("\n");

count++;

return;

}

for (i = 0; i < ord; i++) //promising:field[i] is true

{

if (0 <= rev[i] && rev[i] < N)

field[rev[i]] = true;

if (rev[i] + (ord - i) >= 0 && rev[i] + (ord - i) < N)

field[rev[i] + (ord - i)] = true;

if (rev[i] - (ord - i) >= 0 && rev[i] - (ord - i)< N)

field[rev[i] - (ord - i)] = true;

}

for (j = 0; j < N; j++) //backtrack: for field(row) which is false(queen unpositioned)

{

if (field[j] == false)

{

rev[ord] = j;

solve(ord + 1);

rev[ord] = N;

}

}

return;

}