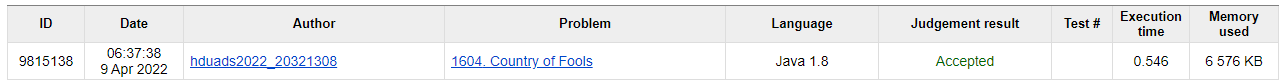
Laboratory work #3

Student: CAO Xinyang  
Student ID: 20321308  
Timus Name: hduads2022\_20321308

Mail: c.x\_yang@foxmail.com

Problem #1604

Screenshot from Timus:



Explanation of algorithm:

First, we divide the large board into four smaller ones. Then we determine if there is a special square in each small board in turn. If so, we just recursively find the solution. If not, cover the joint with L-shaped dominoes, and treat it as a special square for recursion.

The recursion terminates once the board is divided into 1\*1 sub boards.

Computational complexity of algorithm:

F(N) = 4 ^ N + N ^ 2 + (2 ^ N) ^ 2

T(N) = O(4 ^ N)

Source code:

import java.util.Scanner;

public class App {

public static int number = 1;

public static int[][] arr = new int[512][512];

public static void paveSquare(int square\_x, int square\_y, int x, int y, int size) {

if (size == 1) {

return;

}

int t = number++;

int newSize = size / 2;

if (x < square\_x + newSize && y < square\_y + newSize) {

paveSquare(square\_x, square\_y, x, y, newSize);

} else {

arr[square\_x + newSize - 1][square\_y + newSize - 1] = t;

paveSquare(square\_x, square\_y, square\_x + newSize - 1, square\_y + newSize - 1, newSize);

}

if (x < square\_x + newSize && y >= square\_y + newSize) {

paveSquare(square\_x, square\_y + newSize, x, y, newSize);

} else {

arr[square\_x + newSize - 1][square\_y + newSize] = t;

paveSquare(square\_x, square\_y + newSize, square\_x + newSize - 1, square\_y + newSize, newSize);

}

if (x >= square\_x + newSize && y < square\_y + newSize) {

paveSquare(square\_x + newSize, square\_y, x, y, newSize);

} else {

arr[square\_x + newSize][square\_y + newSize - 1] = t;

paveSquare(square\_x + newSize, square\_y, square\_x + newSize, square\_y + newSize - 1, newSize);

}

if (x >= square\_x + newSize && y >= square\_y + newSize) {

paveSquare(square\_x + newSize, square\_y + newSize, x, y, newSize);

} else {

arr[square\_x + newSize][square\_y + newSize] = t;

paveSquare(square\_x + newSize, square\_y + newSize, square\_x + newSize, square\_y + newSize, newSize);

}

}

public static void main(String[] args) throws Exception {

Scanner scan = new Scanner(System.in);

int n = scan.nextInt();

int x = scan.nextInt();

int y = scan.nextInt();

int size = 1;

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

size \*= 2;

paveSquare(0, 0, x - 1, y - 1, size);

for (int i = 0; i < size; i++) {

String str = "";

for (int j = 0; j < size; j++) {

str = str + arr[i][j] + " ";

}

System.out.println(str);

}

scan.close();

}

}