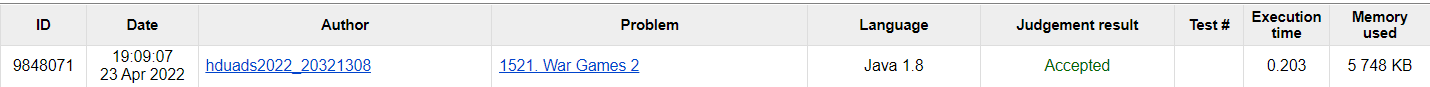
Laboratory work #5

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

Mail: c.x\_yang@foxmail.com

Problem #1521

Screenshot from Timus:



Explanation of algorithm:

Build a tree array, continuously add k module n, and subtract the number before the person is the real number of the current person.

Computational complexity of algorithm:

T(N) = O(N\*(logN)^2)

Source code:

import java.io.\*;

public class App {

static int[] num = new int[100010];

static int N;

static PrintWriter out = new PrintWriter(System.out);

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

StreamTokenizer in = new StreamTokenizer(new BufferedReader(new InputStreamReader(System.in)));

in.nextToken();

int n = (int) in.nval;

in.nextToken();

int k = (int) in.nval;

N = n;

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

plus(i, 1);

}

int count = 1;

int number = n;

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

count = (count + k - 1) % number;

if (count == 0)

count = number;

int temp = search(count, n);

plus(temp, -1);

number--;

}

out.flush();

out.close();

}

private static void plus(int i, int x) {

while (i <= N) {

num[i] += x;

i += i & (-i);

}

}

private static int calculate(int i) {

int ans = 0;

while (i > 0) {

ans += num[i];

i -= i & (-i);

}

return ans;

}

private static int search(int x, int n) {

int l = 1;

int r = n;

while (l <= r) {

int mid = (l + r) / 2;

int temp = calculate(mid);

if (temp < x)

l = mid + 1;

else

r = mid - 1;

}

out.print(l + " ");

return l;

}

}