Laboratory work #2

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

Mail: c.x\_yang@foxmail.com

Problem #2025

Screenshot from Timus:



Explanation of algorithm:

We know from the mathematics that when the number of people in each group is as close as possible, the number of fighting is the most. Therefore, we first give each group (n / k) people, and then divide the rest of the people into different groups, so that the number of people in each group is the closest, so as to maximize the results.

Computational complexity of algorithm:

T(N) = O(N)

Source code:

import java.util.Scanner;

public class App {

public static int check(int n, int k) {

int min = n / k;

int remain = n - min \* k;

int number = 0;

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

number += (min + 1) \* (n - (min + 1));

}

for (int i = 0; i < k - remain; i++) {

number += min \* (n - min);

}

number /= 2;

return number;

}

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

Scanner scan = new Scanner(System.in);

int T = scan.nextInt();

int[] N = new int[T];

int[] K = new int[T];

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

int n = scan.nextInt();

int k = scan.nextInt();

N[i] = n;

K[i] = k;

}

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

System.out.println(check(N[i], K[i]));

}

scan.close();

}

}