Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет систем управления и робототехники

Алгоритмы и структуры данных

Отчёт по лабораторной работе №1 (2025)

Преподаватель: Тропченко А. А.

Выполнил: Марухленко Д. С.

Группа: R3235

Санкт Петербург, 2021г.

# Цель работы

Решить задачу №2025 на платформе Timus Online Judge <https://acm.timus.ru/problem.aspx?space=1&num=2025>

# Задача

## Условие

Бокс, каратэ, самбо… Классические боевые единоборства пресытили аудиторию. Поэтому известный спортивный канал запускает новый формат соревнований, основанный на традиционной русской забаве — боях стенка на стенку. В соревновании могут участвовать от двух до k команд, каждая из которых будет соперничать с остальными. Всего в соревновании примут участие n бойцов. Перед началом боя они должны разделиться на команды, каждый боец должен войти ровно в одну команду. За время боя два бойца сразятся, если они состоят в разных командах. Организаторы считают, что популярность соревнований будет тем выше, чем больше будет количество схваток между бойцами. Помогите распределить бойцов по командам так, чтобы максимизировать количество схваток между бойцами, и выведите это количество.

Ограничение времени: 1.0 секунды

Ограничение памяти: 64 МБ

## Исходные данные

В первой строке дано количество тестов T (1 ≤ T ≤ 10). В следующих T строках перечислены тесты. В каждой из них записаны целые числа n и k через пробел (2 ≤ k ≤ n ≤ 104).

## Результат

Для каждого теста в отдельной строке выведите одно целое число — ответ на задачу.

## Пример

|  |  |
| --- | --- |
| **Исходные данные** | **Результат** |
| 3 6 3 5 5 4 2 | 12 10 4 |

# Материалы работы

# Вариант 1

* + 1. Объяснение алгоритма

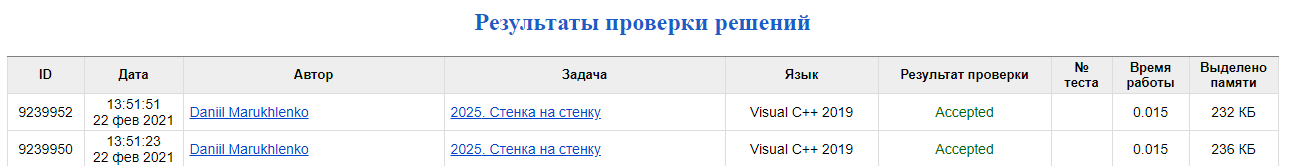
Максимальное число боёв может достигаться тогда, когда распределение участников по командам будет максимально равномерным, чтобы каждый из участников смог сразиться с максимальным количеством противников. В случае, если участников нельзя разделить поровну, число участников в команде должно равняться n//k или n//k+1. Число боёв вчитается циклом, исходя из мысли, что, условно, если каждый участник команды A сразился со всеми участниками всех других команд, то команда A больше не участвует в поединках, значит её можно «снять с состязаний».

* + 1. Код программы.

1. *#include <iostream>*
2. *using namespace std;*
3. *int main(int argc, char const \* argv[]) {*
4. *int t, n, k;*
5. *int team\_members, team\_members\_r, remainder, sum;*
6. *cin >> t;*
7. *for (int i = 0; i < t; i++) {*
8. *cin >> n >> k;*
9. *team\_members = n / k;*
10. *team\_members\_r = team\_members + 1;*
11. *remainder = n % k;*
12. *sum = 0;*
13. *for (int j = 0; j < k - remainder; j++) {*
14. *n -= team\_members;*
15. *sum += n \* team\_members;*
16. *}*
17. *for (int j = 0; j < remainder; j++) {*
18. *n -= team\_members\_r;*
19. *sum += n \* team\_members\_r;*
20. *}*
21. *cout << sum << endl;*
22. *}*
23. *return 0;*
    1. Вариант 2
       1. Объяснение алгоритма

В процессе работы над отчётом появилась идея оптимизации алгоритма и избавления от циклов. Если рассматривать всех участников, как вершины графа, то бои можно представить в виде рёбер графа. Максимальное число боёв – все вершины графа соединены. В таком случае граф будет являться полным, и число рёбер вычисляется по формуле n\*(n-1)/2, где n – число участников. Чтобы исключить бои внутри команды, рассмотрим каждую команду как отдельный граф и вычтем из общего числа боёв число возможных боёв внутри команды.

* + 1. Код программы.

1. *#include <iostream>*
2. *using namespace std;*
3. *int main(int argc, char const \*argv[]) {*
4. *int t,n,k;*
5. *int team\_members,team\_members\_r,remainder,sum;*
6. *cin >> t;*
7. *for (int i=0; i<t; i++){*
8. *cin >> n >> k;*
9. *team\_members = n/k;*
10. *remainder = n%k;*
11. *sum = n\*(n-1)/2 - (k-remainder)\*team\_members\*(team\_members-1)/2 - (remainder)\*team\_members\*(team\_members+1)/2;*
12. *cout << sum << endl;*
14. *}*
15. *return 0;*
16. *}*
17. Результат выполнения и ссылка на репозиторий Github

<https://github.com/japersik/algorithms_and_data_structures/tree/master/task_1_2025>

# Вывод

Работа выполнена, задача была решена «в лоб», сложением всех возможных боёв, и более оптимально, с использованием ранее полученных знаний из других дисциплин.