Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

(ВятГУ)

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ № 1,   
ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ) ПРАКТИКЕ**

**Зуа Анжело Абреу**

*(Ф.И.О. обучающегося)*

02.03.02.51 Фундаментальная информатика и информационные технологии.   
Разработка программного обеспечения

*(направление подготовки (специальность), направленность (профиль))*

|  |  |  |
| --- | --- | --- |
| Место прохождения практики: | | ФГБОУ ВО «Вятский государственный университет», |
|  | *(наименование организации, структурного подразделения организации)* | |

|  |
| --- |
| кафедра прикладной математики и информатики |
| *(наименование организации, структурного подразделения организации)* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Итоговая оценка: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | |
| Руководитель  практики от университета | 04.05.2024 г. |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Котельникова А.В. |
|  | *(дата)* |  | *(подпись)* |  | *(Ф.И.О.)* |

Киров, 2025 г.

**Введение**

Учебная практика проходила в ФГБОУ ВО «Вятский государственный университет», на кафедре прикладной математики и информатики факультета компьютерных и физико-математических наук с 12.02.2024 г. по 05.05.2024 г., количество недель: 12. Установочная конференция состоялась 12.02.2024 г.

Целью учебной практики является закрепление и углубление теоретической подготовки, приобретение практических навыков и компетенций в сфере профессиональной деятельности, в основном путём самостоятельного решения предусмотренных программой задач.

Форма проведения учебной практики: компьютерная (практикум по решению задач на ЭВМ).

В ходе прохождения практики необходимо реализовать компьютерные программы, каждую на двух или трех языках программирования – С++, C# и Python. Каждая задача сначала должна быть принята системой (http://acm.timus.ru) на обоих языках, а потом нужно защитить предложенный алгоритм решения задачи и его программную реализацию в часы консультаций.

Для получения доступа к архиву задач была пройдена регистрация на сайте http://acm.timus.ru, регистрационное имя: [Zua Angelo Abreu](https://acm.timus.ru/author.aspx?id=376439)

Для реализации задач использовалась среда программирования Microsof Visual Studio или vs Code.

**Список задач**

Индивидуальный список задач из архива задач с сайта http://acm.timus.ru, полученный на установочной конференции:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **ID** | **Название** | **Тема** | **Сложность** |
|  | 1207 | Median on the Plane | Задача для начинающих | 124 |
|  | 1277 | Cops and Thieves | ## | 795 |
|  | 2012 | About Grisha N. | ## | 16 |
|  | 2117 | Polyphemus' triples | ## | 532 |
|  | 1872 | Spacious Office | ## | 783 |
|  | 1627 | Join | ## | 1099 |
|  | 1767 | [The House of Doctor Dee](https://acm.timus.ru/problem.aspx?space=1&num=1767) | ## | 722 |
|  | 1242 | [Werewolf](https://acm.timus.ru/problem.aspx?space=1&num=1242) | ## | 178 |
|  | 1382 | Game with Cards | ## | 1742 |

**Описание решенных задач**

В ходе прохождения практики мною были решены следующие задачи: 1207, 1277, 2012, 2117, 1872, 1627, 1767, 1242, 1382 всего 9 задач. Каждая задача была реализована на языках программирования С++, C# и Python. На всех трех языках реализованы задачи: … (3 задачи), на языках С++ и C# реализованы задачи: … (3 задач), на языках С++ и Python реализованы задачи: … (3 задачи).

Детально будут описаны наиболее интересные с точки зрения решения и тестирования решения задачи.

На следующем скриншоте показаны принятые системой задачи.





















**Описание задачи**

* + - 1. **1207. Median on the Plane**

На плоскости расположены N точек (N - чётное число). Ни три точки не лежат на одной прямой. Ваша задача - выбрать две такие точки, что прямая, проходящая через них, делит множество точек на две равные части.

**Подход к решению**

1. **Перебираем все пары точек**, рассматривая их как возможные разделители.
2. **Определяем, с какой стороны от прямой находится каждая точка**:
   * Используем уравнение прямой:

**Ax + By + C = 0**

где:

* + - A = y2 - y1
    - B = x1 - x2
    - C = x2y1 - x1y2

1. **Подсчитываем количество точек по разные стороны**:
   * Если ровно N/2 точек находится по одну сторону и N/2 по другую, значит, эта пара точек подходит.
2. **Выводим индексы выбранных точек** (нумерация с 1).

**Анализ сложности**

* Перебираем все пары точек O(N^2).
* Для каждой пары классифицируем оставшиеся точки, что занимает O(N).
* Общая сложность алгоритма O(N^2), что приемлемо для N ≤ 10000.
  + - 1. **1277. Cops and Thieves**

Необходимо определить, можно ли заблокировать все пути между станциями F и S с помощью заданного количества полицейских. Мы также должны гарантировать, что полиция не будет размещена на самих станциях музея или убежища, а также на их соседних станциях.

**Шаги решения**

1. **Представление графа:** Система транспортировки галактики может быть представлена как неориентированный граф, где:
   * Узлы (вершины) — это станции.
   * Рёбра (каналы) — это телепортационные каналы между станциями.
2. **Ограничения:**
   * Мы не можем разместить полицейских на станциях музея (S) или убежища (F).
   * Мы не можем разместить полицейских на станциях, непосредственно соседних с S или F.
3. **Блокировка маршрутов:** Для того, чтобы заблокировать маршруты между F и S, необходимо разорвать граф, разместив полицейских на определённых станциях. Это эквивалентно задаче на нахождение **минимального разреза** между двумя станциями, где разрез — это набор рёбер или узлов, удалив которые, можно разделить две станции.
4. **Алгоритм:**
   * Мы можем использовать подход **максимальный поток / минимальный разрез**, чтобы решить эту задачу. Идея заключается в том, чтобы представить станции как узлы в сети потока и вычислить минимальный разрез между станциями F и S. Этот разрез даст нам минимальное количество станций (с учётом полицейских), которые нужно контролировать для разрыва связи между S и F.
   * Для вычисления максимального потока можно использовать алгоритм **Форда-Фалкерсона** или **Эдмондса-Карпа**.
5. **Ограничения на размещение полицейских:** После вычисления минимального разреза нужно проверить, если сумма полицейских, требуемых для станций в разрезе, не превышает доступное количество полицейских K.
   * + 1. **2012. About Grisha N.**

**Подход**

1. В первый час Гриша решает **f** задач. Оставшееся время (если оно есть) в первом часе он тратит на прогулки.
2. С каждой последующей задачей, начиная со второго часа, Гриша тратит по 45 минут. У нас есть 4 оставшихся часа (240 минут), и мы должны проверить, хватит ли этого времени для решения оставшихся задач.

**План**

1. В первый час Гриша решает **f** задач. Оставшиеся задачи составляют 12 - f задач.
2. После первого часа у нас остаётся 240 минут на следующие 4 часа.
3. Гриша решает по одной задаче за 45 минут, следовательно, на решение оставшихся задач потребуется: (12 - f) × 45 минут.
4. Если эта сумма меньше или равна 240 минутам, то ответ — **YES**. Если больше — **NO**.

**Алгоритм**

1. Считать число **f**.
2. Вычислить оставшееся количество задач: 12 - f
3. Вычислить, сколько времени нужно для решения оставшихся задач: (12 - f) × 45.
4. Проверить, если это время меньше или равно 240 минутам (время, которое есть во втором и последующих часах).
5. Вывести **YES**, если задача решаема, и **NO**, если нет.