

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.

(дата)

(подпись)

Котельникова А.В.

(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 – C++, 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](#)

Для реализации задач использовалась среда программирования Microsoft 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 |
|-------|------|-------------------------|-----------------------|-----------|
| 1. | 1207 | Median on the Plane | Задача для начинающих | 124 |
| 2. | 1277 | Cops and Thieves | ## | 795 |
| 3. | 2012 | About Grisha N. | ## | 16 |
| 4. | 2117 | Polyphemus' triples | ## | 532 |
| 5. | 1872 | Spacious Office | ## | 783 |
| 6. | 1627 | Join | ## | 1099 |
| 7. | 1767 | The House of Doctor Dee | ## | 722 |
| 8. | 1242 | Werewolf | ## | 178 |
| 9. | 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 C++, C# и Python. На всех трех языках реализованы задачи: ... (3 задачи), на языках C++ и C# реализованы задачи: ... (3 задач), на языках C++ 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.

| | | | | | | | | |
|--------------------------|-------------------------|----------------------------------|---|--------------------|----------|--|-------|----------|
| 10562634 | 01:46:11 22 Feb 2024 | Zua Angelo Abreu | 1207. Median on the Plane | Visual C++ 2022 | Accepted | | 0.031 | 452 KB |
| 10562641 | 02:00:56 22 Feb 2024 | Zua Angelo Abreu | 1277. Cops and Thieves | Visual C++ 2022 | Accepted | | 0.031 | 280 KB |
| 10562688 | 03:34:03 22 Feb 2024 | Zua Angelo Abreu | 2012. About Grisha N. | Visual C++ 2022 | Accepted | | 0.015 | 136 KB |
| 10909522 | 00:59:12 17 Mar 2025 | Zua Angelo Abreu | 2012. About Grisha N. | Python 3.12 x64 | Accepted | | 0.093 | 328 KB |
| 10562697 | 03:50:21 22 Feb 2024 | Zua Angelo Abreu | 2117. Polyphemos' triples | Visual C++ 2022 | Accepted | | 0.046 | 184 KB |
| 10562686 | 03:29:26 22 Feb 2024 | Zua Angelo Abreu | 1872. Spacious Office | Visual C++ 2022 | Accepted | | 0.031 | 324 KB |
| 10562671 | 03:10:00 22 Feb 2024 | Zua Angelo Abreu | 1627. Join | Visual C++ 2022 | Accepted | | 0.015 | 160 KB |
| 10562674 | 03:17:25 22 Feb 2024 | Zua Angelo Abreu | 1767. The House of Doctor Dee | Visual C++ 2022 | Accepted | | 0.031 | 236 KB |
| 10562639 | 01:54:39 22 Feb 2024 | Zua Angelo Abreu | 1242. Werewolf | Visual C++ 2022 | Accepted | | 0.015 | 324 KB |
| 10562655 | 02:35:39 22 Feb 2024 | Zua Angelo Abreu | 1382. Game with Cards | Visual C++ 2022 | Accepted | | 0.046 | 4 184 KB |

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:

- $A = y_2 - y_1$
- $B = x_1 - x_2$
- $C = x_2y_1 - x_1y_2$

3. **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.
4. **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 $N \leq 10000$.

2. 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.

3. 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: $(12 - f) \times 45$ 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: $(12 - f) \times 45$.

4. Проверить, если это время меньше или равно 240 минутам (время, которое есть во втором и последующих часах).
5. Вывести **YES**, если задача решается, и **NO**, если нет.

4. 2117 Polyphemus' triples

Для решения этой задачи рассмотрим уравнение:

$$\sqrt{A} + \sqrt{B} = \sqrt{C}$$

Преобразуем его, возведя в квадрат с обеих сторон:

$$A + B + 2\sqrt{AB} = C$$

Откуда следует:

$$2\sqrt{AB} = C - A - B$$

Для того чтобы \sqrt{AB} был целым числом, произведение AB должно быть полным квадратом, то есть $AB = k^2$ для некоторого целого k .

Алгоритм решения:

1. Перебрать возможные значения A , начиная с 0.
2. Вычислить B как $C - A - 2\sqrt{AB}$
3. Проверить, является ли AB полным квадратом.
4. Подсчитать количество таких пар (A, B) .

5. 1872 Spacious Office

Алгоритм:

1. Чтение входных данных:
 - n - количество проектов (и комнат).
 - Список площадей комнат.
 - Диапазоны допустимых площадей для каждого проекта.
2. Сортировка комнат по площади:
 - Чтобы назначать комнаты командам, сначала сортируем комнаты по площади.
3. Назначение комнат командам:
 - Для каждой команды (в порядке индекса), ищем свободную комнату, которая удовлетворяет её условиям (площадь от \min до \max).
 - Если нашли единственную подходящую комнату, назначаем её.
 - Se houver várias opções, armazenamos todas as possibilidades.
4. Проверка результата:
 - Se cada equipe recebeu exatamente um quarto, imprimimos **"Perfect!"** e a atribuição.
 - Se houver mais de uma maneira de atribuir os quartos, imprimimos **"Ask Shiftman for help."**
 - Se não conseguirmos atribuir os quartos, imprimimos **"Let's search for another office."**

6. 1627 Join

Бизнесмен Петя купил новый дом, который состоит из $n \times m$ квадратных комнат, расположенных в виде прямоугольной сетки. Некоторые из комнат являются кладовками (**), а остальные – спальнями ((.)).

Петя хочет соединить все спальни дверями так, чтобы между любой парой спален существовал ровно один путь (то есть, чтобы они образовывали остовное дерево). Двери можно строить только между соседними спальнями, то есть теми, которые имеют общую стену.

Необходимо вычислить, сколькими способами можно выполнить это соединение, взяв результат по модулю 10^9 .

Формат входных данных

- Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 9$) - размеры сетки.
- Следующие n строк содержат ровно m символов каждый, описывающих план дома:
 - . (точка) обозначает спальню.
 - * (звёздочка) обозначает кладовку.
- Гарантируется, что хотя бы одна спальня присутствует.

Формат выходных данных

Выведите одно число — количество способов соединить все спальни в одно остовное дерево по модулю 10^9 .

7. 1767 The House of Doctor Dee

Город построен по прямоугольной схеме, и персонажи всегда используют один из кратчайших маршрутов. Это означает, что они движутся либо горизонтально, либо вертикально.

Нашей задачей является нахождение максимальной длины пересечения двух маршрутов:

- Маршрут Dr. Dee: от (x_1, y_1) до (x_2, y_2) .
- Маршрут Matthew Palmer: от (x_3, y_3) до (x_4, y_4) .

Поскольку можно двигаться только в направлениях вверх-вниз и влево-вправо, длина кратчайшего пути от точки (x_1, y_1) до (x_2, y_2) равна:

$$|x_2 - x_1| + |y_2 - y_1|$$

То же самое применимо ко второму маршруту.

Теперь нам нужно найти максимальную длину их общего участка.

Подход:

- Пути Dr. Dee и Matthew представляют собой отрезки на манхэттенском пути (grid-based movement).
- Общая часть маршрута — это пересечение двух отрезков, когда оба персонажа находятся на одной координате.

Шаги решения:

1. Найти пересекающийся отрезок по оси X (если есть):

$$[\max(\min(x_1, x_2), \min(x_3, x_4)), \min(\max(x_1, x_2), \max(x_3, x_4))]$$

2. Найти пересекающийся отрезок по оси Y (если есть):

$$[\max(\min(y_1, y_2), \min(y_3, y_4)), \min(\max(y_1, y_2), \max(y_3, y_4))]$$

3. Суммировать пересекающиеся отрезки.

8. 1242. Werewolf

В деревне произошла трагедия — завелся оборотень, который беспощадно убивает жителей. Однако известно, что:

1. Оборотень **никогда не убивает своих предков**.
2. Оборотень может убить любого жителя, кроме своих потомков.

Даны родственные связи жителей (кто чей родитель) и список жертв. Нужно определить **подозреваемых** — тех, кто мог быть оборотнем.

Входные данные

1. Число жителей N ($2 \leq N \leq 1000$).
2. Несколько строк с парами `child parent`, указывающими на родственные связи.
3. Строка "BLOOD", после которой идут номера убитых жителей (по одному на строке).

9. 1382. Game with Cards

Нам даны N друзей, каждый из которых делает две взаимно противоречащие утверждения:

1. Он сам имеет карту с номером a_i .
2. Друг под номером b_i имеет карту с номером c_i .

Причем гарантируется:

- Ровно одно из этих утверждений правда.
- Два друга не могут делать одинаковые утверждения о третьем друге.
- Никто не скажет, что b имеет карту c , если b уже утверждает, что он действительно имеет карту c .

Алгоритм решения

1. Создадим структуры данных:
 - Массив `result` длины N для хранения результата (1 или 2).
 - Массив `card_owner` длины $N+1$ (чтобы удобно хранить владельцев карт от 1 до N).
2. Назначим владельцев карт:

- Если утверждение "Я имею карту a_i " возможно (карта a_i не занята), временно считаем его правдой.
- Если оно не подтверждается, тогда принимаем второе утверждение как истинное.

3. Проверяем вторые утверждения:

- Если карта c_i действительно может принадлежать другу b_i , обновляем решение.

Заключение

Учебная практика способствовала закреплению и углублению теоретической подготовки, приобретению практических навыков и компетенций в сфере профессиональной деятельности путём самостоятельного решения задач.

В ходе практики было решено 9 задач, реализованных на языках программирования C++, C# и Python. Листинги всех составленных программ приведены в приложении.

Список литературы

1. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. – М.: Вильямс, 2016. – 400 с.
2. Вирт Н. Алгоритмы и структуры данных. М.: ДМК, 2011. – 272 с.
3. Задачи по программированию: учебное пособие / С. М. Окулов, Т. В. Ашихмина, Н. А. Бушмелева, М. А. Корчёмкин. – 3-е изд. – М.: Лаборатория знаний, 2017. – 826 с.
4. Кнут Д. Искусство программирования. Том 1. Основные алгоритмы. – 3-е изд. – М.: Вильямс, 2017. – 720 с.
5. Кнут Д. Искусство программирования. Том 3. Сортировка и поиск. – 2-е изд. – М.: Вильямс, 2017. – 824 с.
6. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. – М.: Вильямс, 2019. – 1328 с.
7. Лутц М. Изучаем Python. Том 1. – М.: Издательство Диалектика, 2019. – 832 с.
8. Окулов С. М. Основы программирования: учебное пособие / С. М. Окулов. – 10-е изд. – М.: Лаборатория знаний, 2020. – 339 с.
9. Окулов С. М. Программирование в алгоритмах [Текст] / С. М. Окулов. – 6-е изд. – М.: Лаборатория знаний, 2017. – 386 с.
10. Окулов С. М. Алгоритмы обработки строк: учебное пособие / С. М. Окулов. – 4-е изд. – М.: Лаборатория знаний, 2020. – 258 с.
11. Окулов, С. М. Дискретная математика. Теория и практика решения задач по информатике: учеб. пособие / С. М. Окулов. – М.: Лаборатория знаний, 2020. – 422 с.
12. Окулов, С. М. Динамическое программирование: учебное пособие / С. М. Окулов, О. А. Пестов. – 3-е изд. – М.: Лаборатория знаний, 2020. – 299 с.