

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата

09.03.04 Программная инженерия

Профиль направления подготовки бакалавриата

“Системное и прикладное программное обеспечение”

Отчет о прохождении производственной практики

Участие в интенсивах Rucode. Решение олимпиадных задач
Дивизиона А/В на чемпионате Rucode

Выполнил:

студент группы 22207

К.С. Кришталь _____
подпись

Место прохождения практики:

Кафедра информатики и математического обеспечения

Период прохождения практики:

29.05.23-11.06.23

Руководитель:

О.Ю. Богоявленская, к.т.н., доцент

подпись

Итоговая оценка

оценка

Содержание

Введение	3
1 Интенсивы	4
1.1 Выбор дивизиона для прохождения интенсивов	4
1.2 Изученные темы	4
1.3 Практика	4
2 Чемпионат	5
3 Задача с чемпионата	6
4 Дневник практики	12
Заключение	12
Приложение А. Диплом призера третьей степени	13

Введение

Цель практики:

Участие в фестивале RuCode для получения и применения теоретических знаний в области алгоритмического программирования. Отработка навыков командой работы на чемпионате.

Задачи практики:

- на лекциях изучить новые алгоритмы и идеи, а так же закрепить и углубить знания по уже имеющимся темам;
- изучив лекции, на практике получить или улучшить опыт применения знаний;
- достойное выступление на всероссийском чемпионате в конце фестиваля;
- улучшение командного взаимодействия в рамках чемпионата и взаимопомощи по решению задач на интенсивах.

План прохождения практики:

В рамках всероссийского учебного фестиваля по искусственному интеллекту и программированию RuCode проводились интенсивы и чемпионат по двум направлениям: искусственный интеллект и алгоритмическое программирование (далее АП). Мной было принято участие во втором направлении, АП.

В течение 5 дней, 15-19 мая, проводились интенсивы по АП.

Каждый день интенсивов имел следующий план:

- лекция
- тематический контекст по лекции
- разбор задач тематического контекста

21 мая был проведен чемпионат по АП в разных точках страны. Одной из точек был Петрозаводский государственный университет, в стенах которого мы с командой из 3-х человек и приняли участие. Чемпионат проходил в течение 5 часов и делился на 2

различных по уровню сложности дивизиона: $C - F$ и A/B , где в A/B приняли участие продвинутые команды.

Наша команда «Радикальные псевдокодеры» организовала работу по достижению целей и выполнению задач, приняв участие в чемпионате A/B .

1 Интенсивы

1.1 Выбор дивизиона для прохождения интенсивов

Занятия проходили сразу в 4 дивизионах, каждый из которых отличался сложностью изучаемых тем. Здесь перечислены все дивизионы по возрастанию уровня сложности: F, E, D, C . близким к моему уровню знаний оказался див. C , изучением тем которого я и занимался.

1.2 Изученные темы

Каждому дню занятий соответствовала своя тема. Ниже приведены темы всех 5 дней:

- SQRT-декомпозиция, алгоритм Мо, Мо на деревьях, 3D-Мо.
- Метод сканирующей прямой.
- Паросочетания, минимальное вершинное покрытие/максимальное независимое множество.
- Алгоритм Ахо-Корасик.
- Потоки и Ко: Определение, Форд-Фалкерсон, Диниц.

С помощью лекций, проходящих очно в аудиториях ПетрГУ, я изучил все эти темы.

1.3 Практика

После лекций сразу шла практика. На каждый день было предложено порядка 10 задач различной сложности, начиная от «напишите то, что мы прошли» и заканчивая серьезными задачами уровня очень сложных соревнований.

Посмотрев разбор некоторых задач, мне удалось подчерпнуть новых идей, которые можно применять в дальнейшем.

2 Чемпионат

21 мая состоялся чемпионат, являющийся завершением фестиваля. Я с командой писал его в организованной на базе ПетрГУ очной точке.

Нам удалось решить 4 из 12 задач, при том, что некоторые из них имели крайне высокий уровень сложности, в связи с участием в чемпионате ведущих команд мира.

После конкурса прошел разбор задач, на котором мы узнали как решать те задачи, что не поддались нам во время соревнования, однако 3 из них были почти решены нами во время конкурса, и для успешной их сдачи нам не хватало маленьких нюансов.

По итогу чемпионата мы заняли 9 место из 49 и получили диплом 3 степени.

3 Задача с чемпионата

На рис. 1 представлено условие одной из задач чемпионата.

Задача Н. «Железные» секреты

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 512 мегабайт

Компания Thinkbox планирует построить новую фабрику в джунглях, которая будет выпускать новый продукт компании — беспроводные клавиатуры с миниджойстиком, подсветкой клавиатуры и полугодовым временем работы между зарядками. Успех этого продукта гарантирует компании Thinkbox длительное доминирование на рынке.

Чтобы избежать утечек информации к конкурентам, фабрика будет расположена *под лесом* и будет иметь форму квадрата. Лес на данной территории может быть представлен как матрица $R \times C$, для каждой клетки которой определён вид деревьев, доминирующих в данной клетке. Виды деревьев заданы строчными латинскими буквами. Территория фабрики должна быть квадратом $F \times F$, где $1 \leq F \leq \min(R, C)$, стороны которого проходят точно по границам между клетками леса.

Подлесное расположение фабрики обозначает, что фабрика должна быть максимально безопасно расположена. Сейчас конкуренты могут легко сделать фотографию участка леса, под которым расположена компания, используя дроны. Поэтому CEO компании хочет выбрать конфигурацию типов деревьев над фабрикой таким образом, что в лесу будет как минимум K квадратов с точно такой же стороной и одинаковым расположением доминирующих деревьев (иначе говоря, для каждого из K квадратов существует параллельный перенос, который при совмещении этих квадратов даёт полное совпадение всех обозначающих типы букв). Квадраты при этом могут перекрываться.

Подлесное расположение фабрики обозначает также, что фабрика должна быть построена «на халяву», но если такое невозможно — то с минимумом дополнительных затрат. Налог на постройку фабрики не зависит от длины стороны квадрата, то есть большая длина F обозначает меньшие относительные затраты на квадратный метр.

Вам дана карта леса и значение коэффициента секретности K . Найдите максимальное значение стороны квадрата F для постройки фабрики, или -1 , если требование секретности выполнить не удастся.

Формат входных данных

Первая строка входных данных содержит три целых числа R , C и K ($1 \leq R, C \leq 600$, $2 \leq K \leq R \cdot C$) — количество строк и столбцов в матрице, задающей лес, и коэффициент секретности (число одинаково выглядящих сверху квадратов).

Далее следуют R строк, каждая из которых содержит C строчных латинских букв — карту леса. Клетки с одним и тем же доминирующим типом растительности обозначены одинаковыми буквами, клетки с разными доминирующими типами — разными.

Формат выходных данных

Выведите одно целое число — максимальную длину квадрата F . Если требование секретности выполнить не удастся и K одинаково выглядящих квадратов не существует, выведите -1 .

Примеры

стандартный ввод	стандартный вывод
4 5 2 baacd bbaab bbbba bbbbb	3
3 3 5 cbb dxb xxx	-1

Рис. 1 – Условие задачи Н с чемпионата Rucode 6.5 A/B

Нашей команде почти удалось решить эту задачу (была сдана чуть позже после окон-

чания соревнования), и ниже описана идея с программной реализацией:

Введём функцию $f(x)$. $f(x) = 1$, если x является потенциальным ответом (есть хотя бы k одинаковых квадратов со стороной x) и $f(x) = 0$ иначе.

W_x - Множество из k одинаковых квадратов, со стороной x . W_x Существует если $f(x) = 1$. Элементами множества для удобства будут левые верхние углы квадратов.

Лемма 1:

Если $f(x) = 1$, то $f(y) = 1, y \leq x$.

Доказательство:

W_y может быть равно W_x . Действительно, так как элементами множеств являются углы квадратов, а $y \leq x$. Говоря простым языком, мы возьмем в W_y левые верхние подквадраты уже одинаковых квадратов из W_x

Введем хеш функцию от квадрата - $h(square)$. Вычислять ее можно как полиномиальный хеш матрицы, по аналогии с полиномиальным хешом строки.

Основываясь на лемме 1, можно реализовать бинарный поиск по ответу. Пусть мы в бинарном поиске зафиксировали размер стороны квадрата X . Теперь можно перебрать все квадраты pos со стороной X , и запомнить в хеш-таблице, сколько квадратов с $h(pos)$ встретилось. Если для какого-то $h(pos)$ встретилось k или более квадратов, то $f(x) = 1$, иначе 0.

Таким образом, итоговая асимптотическая оценка времени работы алгоритма - $\mathcal{O}(n \cdot m \cdot \log(\min(n, m)))$.

Заметим, что при реализации полиномиального хеширования по простому модулю, мы будем иметь достаточно высокую коллизию, поэтому в данной задаче высчитывались 2 хеша, а в рамках реализации для этого использовалась встроенная в C++ `pair`.

Код решения на языке C++ представлен в листинге 1:

Листинг 1 – Решение задачи Н.

```
#include <bits/stdc++.h>
using namespace std;
```

```

#define int long long
#define htype pair<int, int>

htype operator + (htype a, htype b){
    return {a.first + b.first, a.second + b.second};
}

htype operator - (htype a, htype b){
    return {a.first - b.first, a.second - b.second};
}

htype operator * (htype a, htype b){
    return {a.first * b.first, a.second * b.second};
}

htype operator % (htype a, htype b){
    return {a.first % b.first, a.second % b.second};
}

const int SZ = 1 << 10;

htype mod = {1e9 + 7, 1e9 + 9};
htype base = {29, 31};
htype h[SZ][SZ];
htype pw[SZ * SZ];
htype rw[SZ * SZ];

char a[SZ][SZ];
int n, m, k;

int f_pow(int a, int p, int mod){
    if (p == 0){
        return 1;
    }
    if (p & 1){

```

```

        return a * f_pow(a, p - 1, mod) % mod;
    }
    return f_pow(a * a % mod, p >> 1, mod);
}

hType get(int i, int l, int r){
    return (h[i][r] - h[i][l - 1] + mod) * pw[SZ - r] % mod;
}

bool f(int x){
    unordered_map<int, int> mp;
    for (int j = 1; j <= m - x + 1; j++){
        hType cur = {0, 0};
        for (int i = 1; i <= x; i++){
            cur = (cur + get(i, j, j + x - 1) * pw[(i - 1) * x]) % mod;
        }
        mp[cur.first + (cur.second << 32)]++;
        for (int i = x + 1; i <= n; i++){
            cur = (cur - get(i - x, j, j + x - 1) + mod) % mod;
            cur = (cur * rw[x]) % mod;
            cur = (cur + get(i, j, j + x - 1) * pw[x * (x - 1)]) % mod;
            mp[cur.first + (cur.second << 32)]++;
        }
    }

    for (auto e : mp){
        if (e.second >= k){
            return true;
        }
    }

    return false;
}

void solve() {

```

```

pw[0] = {1, 1};
rw[0] = {1, 1};
for (int i = 1; i < SZ * SZ; i++){
    pw[i] = pw[i - 1] * base % mod;
    rw[i].first = f_pow(pw[i].first, mod.first - 2, mod.first);
    rw[i].second = f_pow(pw[i].second, mod.second - 2, mod.second);
}

cin >> n >> m >> k;
for (int i = 1; i <= n; i++){
    for (int j = 1; j <= m; j++){
        cin >> a[i][j];
        h[i][j] = (h[i][j - 1] + make_pair(a[i][j] - 'a' + 1, a[i][j] - 'a' + 1
    }
}

if (!f(1)){
    cout << -1;
    return;
}

int l = 1;
int r = min(n, m) + 1;
while (l < r){
    int mid = (l + r) >> 1;
    if (f(mid)){
        l = mid + 1;
    }else{
        r = mid;
    }
}

cout << l - 1;
}

```

```
signed main(){  
    ios_base::sync_with_stdio(0);  
    cin.tie(0);  
    cout.tie(0);  
  
    solve();  
}
```

4 Дневник практики

Период	Выполняемая работа
15.05 - 19.05	Интенсивы Rucode.
21.05 - 21.05	Участие в чемпионате Rucode A/B.
30.05 - 08.06	Подготовка отчета о результатах прохождения практики

Заключение

По итогу участия в фестивале я улучшил свои знания в области продвинутых алгоритмов, получил хорошую практику применения своих знаний, подумал над решением некоторых сложных тематических задач.

На чемпионате подкрепил свои навыки работы в команде, а так же нам удалось достичь хорошего результата, и получить диплом 3 степени.

Список литературы

1. сборник алгоритмов // <https://e-maxx.ru/algo/>

Приложение А. Диплом призера третьей степени

