ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики

Подчезерцев Алексей Евгеньевич, группа БИВ172
Солодянкин Андрей Александрович, группа БИВ172

КОМПЬЮТЕРНАЯ ИГРА «ЭРУДИТ» (SCRABBLE)

Курсовая работа
по направлению 09.03.01 Информатика и вычислительная техника
студента образовательной программы бакалавриата
«Информатика и вычислительная техника»

Студент _____ А.Е. Подчезерцев
Студент _____ А.А. Солодянкин

Руководитель
_____ Е.А. Ерохина

Москва 2018г.

Содержание

1 Аннотация

Разрабатываем игрушку

2 Словарь игры

2.1 Условие задачи

Дан исходный массив слов и набор букв, которые уже есть на поле и у игрока. Необходимо выбрать из исходного списка слов такие, которые теоретически можно составить из данных букв

2.2 Постановка задачи

Дано:
$Words[0 : n-1]$ - строки
$Letters[0 : m-1]$ - символьные
Результат:
$NewWords[0 : k-1]$ - строки
При:
$n \geqslant 1, 1 \leqslant m \leqslant 32$
Связь:
$k = n, NewWords[0 : k-1] = Words[0 : n-1]$, если $m = 32$
$i = \overline{0, k-1}$
$j = \overline{0, n-1}$
$NewWords[i] = Words[j]$, если $\forall Words[j][e], \exists C, C \in LettersWords[j][e] = C$

2.3 Внешняя спецификация

Данная функция не предусматривает взаимодействие программы с пользователем

2.4 Описание алгоритмов

Алг 1. Подготовка словаря

$k := 0$
Цикл от $i := 0$ **до** $n - 1$
    $flag := true$
    $j := 0$
    Цикл-пока $j <$ длина$(Words[i])$ **и** $flag$
        Если $Words[i][j] \in Letters$ **то**
            $j := j + 1$
        иначе
            $flag := false$
        Всё
    кц
    Если $flag$ **то**
        $NewWords[k] := Words[i]$
        $k := k + 1$
    Всё
кц

2.5 Тесты

Тестовый словарь:

биосфера блюз дворянство домолачивание заковывание изъян киноведение колеровщик координированность митраизм налавливание неминуемость одухотворенность окраина плавсостав поборник подхват приматывание пролысина сипловатость солододробилка топаз трином трехсотлетие умывание хранилище централизация шейх

Тесты обрабатывают тестовый словарь и проверяют длину итогового массива

Входные данные	Контрольное значение
АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ	Длина словаря
А	0
БЛЮЗ	1
АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮ	Длина словаря - 3

3 Использованные классы

3.1 Matrix

В классе $Matrix$ осуществляется поиск слов и подсчет стоимости этих слов. Класс $Matrix$ содержит следующие методы:

- $\_\_init\_\_$
- $serch$
- $\_prov$
- $\_schit$
- $reject\_temp$
- $pasteletters$
- $\_ChekKoord$
- $\_ValidationCheck$
- $ValidationKoord$

Инициализация переменных происходит в функции $\_\_init\_\_$

С основной задачей класса справляется функция $serch$. Для её работы нужны два массива: один с координатами новых точек, другой с новыми буквами. Но, перед тем, как находить слова, необходимо подготовить и проверить матрицу и исходные массивы, за это отвечают функции: $\_ChekKoord$, $pasteletters$ и $ValidationKoord$.

$\_ChekKoord$ преобразует исходные массивы, делается это из-за того, что в процессе составления массивов в них попадает ненужная информация.

$pasteletters$ вставляет данные из массивов в матрицу.

$ValidationKoord$ с помощью рекурсивной функции $\_ValidationCheck$ определяет правильность заполнения матрицы, другими словами, функция проверяет выполнение следующих условий: конструкция из пересекающихся слов должна содержать в себе точку с координатами (7,7) и не должно быть букв, не принадлежащих этой конструкции.

Функция $reject\_temp$ предназначена для работы с классом извне. Она очищает временные переменные.

Если $ValidationKoord$ возвращает положительный результат, можно начинать поиск слов. Пробегаемся по всей матрице (кроме последней строки и последнего столбца) и для каждой не пустой ячейки запускаем $\_prov$. $\_prov$ распознает начало слова, при положительном результате запускается функция $\_schit$. $\_schit$ проходит до конца слова, параллельно считая стоимость слова.

Полученный результат обрабатывается, и передаются при помощи класса $MatrixRes$

3.2 MatrixResult

Данный класс предназначен для обработки результата метода $serch$ из класса $Matrix$. $MatrixResult$ состоит всего из одной функции $\_\_init\_\_$, в которой происходит преобразование полученных данных в более удобный формат.