**Лабораторная работа 2**

**Тема:** Структурное тестирование программного обеспечения.

**Задание:** Лабораторная состоит из двух частей:

1.1 Необходимо протестировать требования, предъявляемые заказчиком. Требования берём из документа lab1V2.pdf. Для каждого требования необходимо написать уточняющие вопросы если требование сформулировано не корректно или не однозначно. Номер задания рассчитываем по формуле. Остаток от деления (порядковый номер в списке / количество заданий) + 1

1.2 Провести структурное тестирование алгоритма по методу базового пути. Берём алгоритм из приведённого списка, пишем алгоритм для решения задачи. Для каждой функции строим потоковый граф. Ищем все возможные пути в графе и пишем для каждого пути свой тест.

**Варианты заданий:**

Вариант 1. Компилятор простых арифметических выражений, например 2+(-5)\*(7-8). Вход и выход в виде строк

Вариант 2. Функция поиска пути в неориентированном графе методом поиска в ширину. На вход подается граф и две вершины. На выходе – путь между этими вершинами.

Вариант 3. Функция поиска пути в неориентированном графе методом поиска в глубину. На вход подается граф и две вершины. На выходе – путь между этими вершинами.

Вариант 4. Функция поиска пути в неориентированном графе методом А\*. На вход подается карта (граф с географическими координатами вершин) и два угла. На выходе – путь между этими узлами.

Вариант 5. Функция балансировки двоичного дерева

Вариант 6. Функция, рассчитывающая контур пересечения двух треугольников

Вариант 7. Хеш-таблица, не перетирающая элементы при вводе значений с совпадающим ключом, а хранящая список таких элементов и, соответственно, возвращающая их методом get. Метод – двойное хеширование

Вариант 8. Функция, рассчитывающая следующий ход в игре крестики-нолики на доске заданного размера и для заданной длины выигрышной последовательности путем построения полного дерева решений(например, доска 5 на 5 и длина выигрышной последовательности 4)

Вариант 9 Функция, производящая поиск заданного набора строк в текстовом файле. Поиск должен уметь находить любую строку из набора, при этом должен правильно обрабатывать переносы текста. Использовать алгоритм Ахо-Корасик.

Вариант 10Парсер, использующий простые регулярные выражения, вводимые с клавиатуры, содержащие управляющие конструкции . – любой символ, \* - 0 и более символов, + - 1 и более символов (вводится регулярное выражение и строка, результатом является позиция, с которой это выражение встречается в тексте)

Вариант 11 Молекула ДНК состоит из последовательностей нуклеотидов А, Г, Ц и У. Несколько одинаковых молекул известной длины были нарезаны на фрагменты произвольной длины.

Функция восстанавливает исходную молекулу в том случае, если это возможно сделать единственным образом Пример: АГЦЦГГУААЦЦ нарезана на фрагменты АГЦЦ, ЦГГУ , ГГУАА и УААЦЦ. Пример невосстанавливаемой последовательности: АГЦЦГГУААЦЦ нарезана на фрагменты АГЦЦ, ГГУАА и УААЦЦ. Вариант алгоритма решения. В памяти строится ориентированный граф, в вершинах которого находятся фрагменты, а связи соединяют два фрагмента, если фрагмент-источник может быть слева от фрагмента-приемника. Далее в

графе ищутся все пути и для каждого проверяется, что он содержит в себе все фрагменты. Если такой путь один, то задача считается решенной

Вариант 12В матричной форме задается система линейных уравнений, необходимо ее решить (например, методом Гаусса).

Вариант 13. Реализовать структуру «Список с пропусками”, см. https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA\_%D1%81\_%D0%BF%D1%80%D0%BE%D0%BF%D1%83%D1%81%D0%BA%D0%B0%D0%BC%D0%B8

Реализовать функции добавления, удаления и поиска.

Вариант 14. В заданном произвольном тексте найти все повторяющиеся фрагменты текста длиной не менее трех слов (без использования стемминга, т.е. слова в различных склонениях и падежах считаются разными, знаки препинания не учитываются). При этом для каждого повторяющегося фрагмента должна указываться максимальная длина, например, для данного текста · «поиска пути в неориентированном графе методом поиска в” встречается дважды · «поиска пути в неориентированном графе методом” встречается трижды· Более короткие части отдельно не встречаются, поэтому не рассматриваются

Вариант 15. Написать программу для решения судоку произвольного размера.

Вариант 16. Некое заведение общепита очень любит акции. Акции заключаются либо в назначении сниженной цены за комбинацию продуктов, либо в предоставлении бесплатного продукта при покупке определенной комбинации продуктов. Напишите алгоритмы для определения набора акций такого, чтобы:

a. купить желаемый набор продуктов максимально выгодно.

b. максимально сытно поесть на фиксированную сумму.

Вход — меню (с ценами и калорийностью), акции, набор продуктов или сумма. Подсказка: Задача a может быть решена с помощью алгоритма для решения задачи b.

Вариант 17. Представьте, что вы являетесь разработчиком стратегии реального времени. Допустим, игрок может создавать на карте здания-приемники (амбары, лесопилки, кузницы и т. д.). В них поступают ресурсы с прилегающих территорий (зерно, лес, руда и т. д.). При имеющемся расположении приемников необходимо поделить всю карту на непересекающиеся области таким образом, чтобы в каждой области был ровно один приемник и он был ближайшим для всех ресурсов в этой области. Можно считать, что препятствий на карте нет.

Вариант 18. У преподавателя есть список заданий, каждое из который имеет тип (теория/практика/блиц/прочие), тематику и уровень сложности. Постройте алгоритм для генерации списка билетов, такой, чтобы одновременно выполнялись условия: 1) в зависимости от внешних условий состав билетов меняется; 2) билет содержит одинаковое количество вопросов из разных тем, и одинаковое соотношение заданий разных типов (например, 2 теории и одна практика); 3) уровень сложности всех билетов приблизительно одинаковый.

Вариант 19. Постройте программу для решения головоломки «Пятнашки» произвольного размера.

Вариант 20. Напишите программу для получения минимальной ДНФ булевой функции.

Вариант 21. Предположим, что вы устроились погромистом в некую успешную компанию. Вам поручили автоматизировать составление сменного графика работы сотрудников. В зависимости от должности, каждый сотрудник должен отработать определенное число дней в месяц (например, 22 из 30) по сменному графику. Каждый из сменных работников отправляет желаемое расписание на месяц, а ваша задача — составить график так, чтобы максимально удовлетворить все заинтересованные стороны (сделать меньше всего изменений в графиках

работников и избежать случая, когда в один день на работе 10 сотрудников, а в другой — ни одного). Помните, что число дней в месяце может быть разным.

Вариант 22. Напишите алгоритм, который принимает на вход набор целых неотрицательных чисел и переупорядочивает их так, чтобы сумма любых двух соседних чисел была полным квадратом (или выводит ответ, что это невозможно).