**Отчет по лабораторной работе № 7** по курсу “Фундаментальная информатика”

Студент группы М80-101Б-22, Бычков Артур Сергеевич, № по списку 2

Контакты email: bychkovarthur@gmail.com

Работа выполнена: «14» октября 2022г.

Преподаватель: каф. 806 Крылов Сергей Сергеевич

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Отчет сдан « » \_\_\_\_\_\_\_\_\_202 \_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Тема:** Нормальные алгоритмы Маркова
2. **Цель работы:** Научиться работать с нормальными алгоритмами Маркова и составлять их.

1. **Задание** (вариант № 2\*)**:** Входное слово представляет собой два троичных числа без знака, разделенные знаком «-». Составить алгоритм вычисления разности этих чисел.

1. **Оборудование**:

*Оборудование ПЭВМ студента, если использовалось:*

Процессор **AMD Ryzen 5 5500U 2.10 GHz, 6 ядер** с ОП **8192 Мб**, ТТН **512000 Мб**. Мониторы **Lenovo.**

1. **Программное обеспечение:**

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства **Linux**, наименование **Ubuntu** версия **20.04.5,** интерпретатор команд bash версия 5.0.17(1).

Система программирования CLion версия 2021.1.3

Редактор текстов nano версия 6.2

Утилиты операционной системы WinRar, Microsoft Word.

Прикладные системы и программы Ubuntu wsl, Clion, Google Chrome

Местонахождение и имена файлов программ и данных на домашнем компьютере /home/artur

**6. Идея, метод, алгоритм** решения задачи(в формах:словесной,псевдокода,графической[блок-схема,диаграмма,рисунок,таблица] или формальные спецификации с пред- и постусловиями)

***Идея:***

Из каждого числа вычитать по единице, и, если правое число обнулилось, то я убираю минус между числами и оставляю только левое, иначе, я стираю левое и оставляю минус, т.к. левое число было меньше правого.

Сложность такого алгоритма = O(n), где n – наименьшее из двух чисел.

***Алгоритм:***

1. Проверяю, если у меня есть ведущие нули в правом числе, то я их убираю (-0 -> -).
2. Ставлю \*\* в конец правого слова.
3. После, если я встречаю 0\*\*, то я меняю на \*\*2, когда я встречаю не нулевой разряд, то уменьшаю его на единицу и количество звездочек увеличиваю на 1, (например, было 1\*\* станет \*\*\*0).
4. Перехожу к левому числу, увеличивая количество звездочек до 4-х.
5. Повторяю пункт 3), но для первого числа и с количеством звездочек 4 и 5 соответственно.
6. Далее с количеством звездочек 5 убираю незначащие нули у первого числа.
7. Меняю количество звездочек на 6, и иду к знаку «-».
8. Меняю количество на 7 и иду в конец правого числа.
9. Меняю количество звездочек на 2 и повторяю действия, начиная с пункта 2).
10. Если от правого числа ничего не осталось (пустота, т.к. я всегда убираю ведущие нули), то я, убирая знак «-» и остается ответ.
11. Если от левого числа остался лишь один ноль, то я полностью стираю звездочки и левое число, тогда ответом будет является знак «-» и правое число.

**7. Сценарий выполнения работы** [план работы,первоначальный текст программы в черновике(можно на отдельном листе)итесты либо соображения по тестированию].

|  |  |  |
| --- | --- | --- |
| ***Входные данные*** | ***Выходные данные*** | ***Описание тестируемого случая*** |
| 1-0 | 1 | ***Пограничные случаи*** |
| 0-1 | -1 |
| 0-0 | 0 |
| 1-1 | 0 |
| 2-2 | 0 |
| 2-0 | 2 |
| 0-2 | -2 |
| 2210-2210 | 0 |
| 12-12 | 0 |
| 000000000-0 | 0 |
| 0-000000000 | 0 |
| 000000002-0 | 2 |
| 2-000000000 | 2 |
| 20-1 | 12 | ***Займ из старшего разряда*** |
| 10-2 | 1 |
| 1-20 | -12 |
| 2-10 | -1 |
| 1000000000-1 | 222222222 | ***Большое количество займов из старшего разряда*** |
| 1000000000-2 | 222222221 |
| 2000000000-1 | 1222222222 |
| 2000000000-2 | 1222222221 |
| 1-1000000000 | -222222222 |
| 2-1000000000 | -222222221 |
| 1-2000000000 | -1222222222 |
| 2-2000000000 | -1222222221 |
| 102012-0001012 | 101000 | ***Случайные (обычные) проверки*** |
| 102-0010 | 22 |
| 0000021-01020 | -222 |
| 2010-022201 | -20121 |

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

// КОНЕЦ

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*->.

-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*->.-1

// Возвращаем единицу, которую отняли на первом ходу

0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1

1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2

2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0

// Обработка случая, если первое число = 0, а второе от него отлично

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2->2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1->1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0->0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*2\*\*->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*0->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-1->-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-2->-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->.-1

// Я в правом конце и нужно прибавить единицу

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Конец

\*\*\*\*\*\*\*\*\*\*\*\*->.

// Стираю нули

\*\*\*\*\*\*\*\*\*\*\*01->\*\*\*\*\*\*\*\*\*\*\*1

\*\*\*\*\*\*\*\*\*\*\*02->\*\*\*\*\*\*\*\*\*\*\*2

\*\*\*\*\*\*\*\*\*\*\*00->\*\*\*\*\*\*\*\*\*\*\*0

\*\*\*\*\*\*\*\*\*\*\*1->\*\*\*\*\*\*\*\*\*\*\*\*1

\*\*\*\*\*\*\*\*\*\*\*2->\*\*\*\*\*\*\*\*\*\*\*\*2

// Встретил \*\*\*\*\*\*\*\*\*\* - нужно стереть нули

2\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*2

1\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*1

0\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*0

// Если одинокие \*\*\*\*\*\*\*\*\*\*, то я в левом крае числа, значит нужно убирать нули справа

\*\*\*\*\*\*\*\*\*\*->\*\*\*\*\*\*\*\*\*\*\*

// Возвращение направо

\*\*\*\*\*\*-->-\*\*\*\*\*\*\*

\*\*\*\*\*\*\*2->2\*\*\*\*\*\*\*

\*\*\*\*\*\*\*1->1\*\*\*\*\*\*\*

\*\*\*\*\*\*\*0->0\*\*\*\*\*\*\*

// Если я справа, то меняю \*\*\*\*\*\*\* на \*\*

\*\*\*\*\*\*\*->\*\*

// Если наибольший разряд второго числа закончился, убираем его

-0->-

// Возвращение направо

\*\*\*\*\*\*0->0\*\*\*\*\*\*

\*\*\*\*\*\*1->1\*\*\*\*\*\*

\*\*\*\*\*\*2->2\*\*\*\*\*\*

// Перемещение в левый край

2\*\*\*\*\*->\*\*\*\*\*2

1\*\*\*\*\*->\*\*\*\*\*1

0\*\*\*\*\*->\*\*\*\*\*0

// Завершение

\*\*\*\*\*0-1->.-1

\*\*\*\*\*0-2->.-2

\*\*\*\*\*0-->.0

//Стираю незначащие нули у левого числа

\*\*\*\*\*0->\*\*\*\*\*

// Если я слева, то меняю \*\*\*\*\* на \*\*\*\*\*\*

\*\*\*\*\*->\*\*\*\*\*\*

// Вычитание 1 из левого числа

2\*\*\*\*->\*\*\*\*\*1

1\*\*\*\*->\*\*\*\*\*0

0\*\*\*\*->\*\*\*\*2

// Возвращение к началу

\*\*\*\*\*->\*\*\*\*\*\*

// Переход к левому числу

-\*\*\*->\*\*\*\*-

// Встретил -\*\* значит число закончилось.

-\*\*->\*\*\*\*\*\*\*\*\*\*

// Перемещение к минусу

2\*\*\*->\*\*\*2

1\*\*\*->\*\*\*1

0\*\*\*->\*\*\*0

// Вычитание 1 из правого числа.

2\*\*->\*\*\*1

1\*\*->\*\*\*0

0\*\*->\*\*2

// Ставлю звездочки, чтобы образовались 2 штуки и работать дальше с 2-мя

\*0->0\*

\*1->1\*

\*2->2\*

-->-\*

**9. Дневник отладки** должен содержать дату и время сеансов отладки и основные события(ошибки в сценарии и программе,нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы
2. **Выводы**

Благодаря этой работе я научился составлять и использовать нормальные алгоритмы Маркова. Изначально мое задание мне показалось очень сложным, и я подумал, что засяду с ним на несколько дней, как-то было с МТ, однако, заранее посидев и обдумав мой алгоритм я понял, что это будет довольно просто. После обдумывания задачи, я написал приблизительно за час алгоритм, который идеально считает, когда первое число больше второго, однако, я подумал, что можно усовершенствовать алгоритм, чтобы он считал вообще любые числа, что в итоге и сделал, правда на это и ушло больше времени. Мне понравилась эта работа, т.к. после обоих лабораторных работ с МТ голова уже начала мыслить по-другому и получается быстрее находить и реализовывать решение различных задач.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_