

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

Студент группы М80-109Б-22 Фомин Иван Дмитриевич, № 22

Контакты email: grenka388@gmail.com,

Telegram: @Haliaven

Работа выполнена: «10» октября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «17» ____10____2022 г., итоговая оценка ____

Подпись преподавателя _____

1. **Тема:** Нормальные алгоритмы Маркова

2. **Цель работы:** Научиться работать с нормальными алгоритмами Маркова и составлять их.

3. **Задание:** Вариант 25. Составить алгоритм увеличения на единицу целого неотрицательного числа в шестнадцатеричной позиционной системе счисления

4. **Оборудование** (студента):

Процессор AMD Ryzen 7 4800HS with Radeon Graphics с ОП 8 Гб, SSD 512 Гб. Монитор 1920x1080

5. **Программное обеспечение** (студента):

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04 *jammy*

интерпретатор команд: *bash* версия 5.1.16(1)-release

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

6. **Идея, метод, алгоритм** решения задачи

Для обозначения конца числа используем **.

Для всех цифр, кроме F, увеличиваем число на 1 и убираем меняем ** на ***, чтобы завершить программу.

Если встречаем F, то меняем F на 0 и передвигаем * влево. Так делаем, пока не встретим цифру, отличную от F.

В худшем случае алгос работает за $O(n)$, где n-длина исходного числа (FFFFFF)

7. **Сценарий выполнения работы**

Входные данные	Выходные данные	Описание тестируемого случая
0	1	Пограничный случай, одна цифра - ноль
F	10	Пограничный случай, одна цифра – F (переполнение разряда)
FFFFFFFFFF	1000000000	Множественное переполнение разряда подряд
123456789ABCDEF	123456789ABCDF0	Обычный случай

8. Распечатка протокола

***_
>
F**->**0
0**->***1
1**->***2
2**->***3
3**->***4
4**->***5
5**->***6
6**->***7
7**->***8
8**->***9
9**->***A
A**->***B
B**->***C
C**->***D
D**->***E
E**->***F
0->*10
0->0
1->1
2->2
3->3
4->4
5->5
6->6
7->7
8->8
9->9
A->A
B->B
C->C
D->D
E->E
F->F
0->0*
1->1*
2->2*
3->3*
4->4*
5->5*
6->6*
7->7*
8->8*
9->9*
A->A*
B->B*
C->C*
D->D*
E->E*

F->F*

9. Дневник отладки

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	лаб	10.10.2022	11:30	Изначальный алгоритм содержал помимо *, я решил подумать, можно ли его оптимизировать в плане памяти	Переделал алгоритм чуток	Удивляюсь простоте НАМ

10. Замечания автора

Если честно, то замечаний нет, потому что НАМ можно использовать, например, в L-системах (визуализацию которых я, кстати, реализовал на js 😊).

11. Выводы

Мне понравилось работать с НАМ, потому что это просто (в этом, конечно, играет большую роль то, что мне попалась простая задача) и потому, что это реально может пригодиться в дальнейшем. НАМ вообще очень напоминают собой словари, этот факт делает работу с НАМ ещё приятнее.

Претензий нет, потому что они не успели сформироваться, ведь лабу за написал довольно быстро. (это не оправдание)

Как, обычно, анекдот:

Неинтересные факты:

Когда французы проходят экзамен по вождению, у них с лёгкостью получается сдать город.

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

Подпись студента _____