Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

02.03.02

Информационная безопасность

Лабораторная работа № 1

Проектирование алгоритмов поддержки информационной безопасности

Цель работы: научиться проектировать алгоритмы поддержки информационной безопасности.

Задание: необходимо разработать алгоритмы и программы решения задач 0-5 на языке C++.

Указания к работе. На занятии студенты решают задачу № 0 Далее каждому студенту выдается своя задача. Номер задачи, которую решает студент, вычисляется по формуле $N = (X \ mod \ 5) + 1$, где X – номер студента в списке. Студент разрабатывает алгоритм и программу решения задачи на языке C++ на лабораторном занятии. Если студент не успел в течение занятия разработать программу, то ему необходимо закончить разработку к следующему лабораторному занятию, используя систему управления версиями и разместив её на личный Git репозиторий. За работу на занятии студент может получить оценку «зачтено», «удовлетворительно», «хорошо», «отлично». Если студент не получил оценку, то ему будет необходимо подготовить отчёт по задаче и реализовать программу, используя систему управления версиями Git и размещая её репозиторий на GitHub.

Задание 0. Шифровка. Штирлиц хочет передать очень важное сообщение s в штаб. Для этого он использует беспрефиксный код, зафиксированный по ГОСТ. К сожалению, противнику известен код, а канал связи Штирлица прослушивается. Чтобы отвести подозрения, Штирлиц хочет

разбить шифровку на куски, каждый из которых нельзя расшифровать тем же кодом. Для максимальной безопасности Штирлиц хочет, чтобы количество кусков было максимально. Найдите максимальное количество кусков или определите, что разбить шифровку требуемым образом невозможно.

Исходные данные

В первой строке записано одно целое число k ($1 \le k \le 52$) – количество символов в алфавите. Символы пронумерованы от 1 до 52 в порядке A-Za-z, в тексте сообщения будут использованы только символы с номерами от 1 до k.

Во второй строке записана строка s длины до 10^6, состоящая из символов с номерами от 1 до k (нумерация символов определена выше).

В следующих k строках записаны двоичные коды символов алфавита согласно нумерации. Каждый символ алфавита кодируется последовательностью 0 и 1 длины не более k. Гарантируется, что никакой код не является префиксом другого кода.

Результат

Выведите одно число – максимальное количество кусков, либо -1, если разбить на куски указанным способом невозможно.

Таблица 1 – Примеры

исходные данные	результат
3 CACB 011 1 001	2

Продолжение таблицы 1

исходные данные	результат
3 ACBABCAACABCAACC 0 10 110	-1

Замечания

В первом семпле зашифрованный текст выглядит как 0010110011. Единственный способ разбить его на куски, которые нельзя расшифровать – 0 010110011. Следовательно, ответ равен 2.

Во втором семпле можно показать по индукции, что любая строчка, которая заканчивается на 0 и не содержит трёх 1 подряд, расшифровывается данным кодом. Любой суффикс зашифрованного текста имеет такой вид, поэтому ответ – -1.

Задание 1. Шифрограмма. Слово зашифровано «циклическим сдвигом на 3 символа назад». Например, слово СЕЛО, шифр – ОСЕЛ, слово КООРДИНАТА, шифр – РДИНАТАКОО. Написать функцию decode, которая по заданному шифру восстанавливает зашифрованное слово. Задан зашифрованный как указано выше текст, например, в виде текстового файла. Слова в тексте разделяются одним или несколькими пробелами. Написать программу, использующую функцию decode, расшифровки этого текста, например, в результирующий текстовый файл.

Задание 2. Система оповещения. Руководитель организации решил создать систему оповещения своих сотрудников на случай опасности. Для этого он разбил всех сотрудников на несколько групп следующим образом. В первую группу вошли сотрудники, номера телефонов которых он знал сам. Во вторую группу были включены сотрудники, номера которых в целом знали сотрудники первой группы, и т.д. В результате этого получилась схема,

изображенная на рисунке, где представлены три группы сотрудников. Цифра 1 соответствует руководителю. Стрелки определяют знание i-ым сотрудником номера телефона j-го сотрудника.

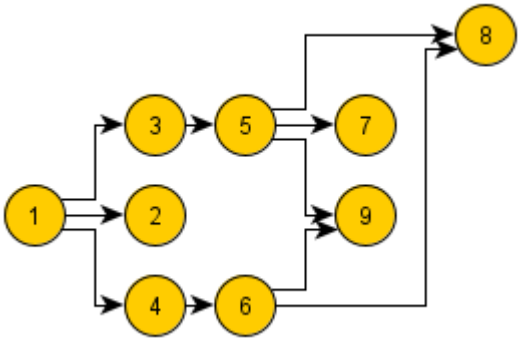

Рисунок 1 – Схема

Система оповещения должна функционировать следующим образом. В случае экстраординарной ситуации руководитель должен последовательно позвонить всем сотрудникам первой группы. Сотрудник, получивший сообщение, должен позвонить по известным ему телефонам сотрудникам другой группы и т.д. В результате этого каждый сотрудник должен быть проинформирован по телефону об экстраординарной ситуации, причем ему могли звонить только один раз. Во время передачи сообщения длительность телефонного разговора у всех сотрудников одинакова и равна 1 минуте.

В зависимости от того, кто кому звонит, общее время оповещения всех сотрудников будет разным. Например, если для изображенной на рисунке 1 схеме будет использоваться последовательность звонков, представленная в таблице 1, то время будет равно 4 минутам, а если в таблице 2, то – 5 минутам. Руководителю нужно знать такую последовательность звонков, которая обеспечит полное оповещение за минимальное время.

Таблица 1 – Последовательность звонков

Источник	Приемник	Время
1	4	1
1	3	2
1	2	3
3	5	3
4	6	2
5	7	4
6	9	3
6	8	4

Таблица 2 – Последовательность звонков

Источник	Приемник	Время
1	2	1
1	3	2
1	4	3
3	5	3
4	6	4
5	7	4
5	8	5
6	9	5

Написать программу, которая по заданной схеме взаимодействия определяет ту последовательность звонков, которая обеспечивает оповещение всех сотрудников за минимальное время. Схема взаимодействия задается во входном текстовом файле input.txt.

Формат файла input.txt:

В первой строке задано число N ($N \leq 20$) – количество сотрудников.

Последующие N строк содержат описание схемы взаимодействия сотрудников. При этом каждая i-ая строка содержит номер i сотрудника и

номера сотрудников, телефоны которых известны i-му сотруднику. Все числа в строке разделены пробелом. Например,

9

1 2 3 4

2

3 5

4 6

5 7 8 9

6 8 9

7

8

9

Результаты формируются в текстовом файле output.txt. В первой строке должно быть одно число – общее время оповещения всех сотрудников в минутах. В каждой из последующих строк должны содержаться пары чисел (номер источника и номер получателя), соответствующие передачи сообщения по телефону в один и тот же период времени. Все числа разделены пробелом.

Например,

4

1 4

1 3 4 6

1 2 3 5 6 9

5 7 6 8

Задание 3. Зазеркалье. Написать программу, которая по заданному входному текстовому файлу input.txt формирует результирующий текстовый файл output.txt. Содержимое входного файла – последовательность, разделенных одним или несколькими пробелами зеркально отраженных слов. Требуется в той же последовательности вывести в output.txt результаты зеркального отображения этих слов. Например, файл input.txt: отЭ ремипр

оготсорп атсет. илсЕ ыВ еще ен иляноп, от етишипаз ывкуб огоджак аволс в монтарбо екдяроп.

Файл output.txt: Это пример простого теста. Если Вы еще не поняли, то запишите буквы каждого слова в обратном порядке.

Задание 4. Взлом игры. Мальчик Петя очень хочет поиграть в новую компьютерную игру, но при ее запуске игра задает 10 вопросов, на которые нужно ответить «да» (t) или «нет» (f). Если на все вопросы Петя ответил правильно, то игра запускается, а если хотя бы на один вопрос был дан неправильный ответ, то игра не запускается. После каждой неудачной попытки игра сообщает, на сколько вопросов было отвечено неправильно. Петя уже сделал N $(1 \leq N \leq 100)$ неудачных попыток. Все свои попытки он записывал. Требуется написать программу, которая по уже известным попыткам определяет ответы на вопросы, а если этого однозначно сделать нельзя, то выводит один из возможных вариантов ответа. Во входном текстовом файле input.txt в первой строчке находится число N. Далее идет N строк длиной 11 символов. Из них первые 10 символов строки – это символы «t» или «f» – информация о том, как отвечал Петя на поставленные вопросы: «t» – если Петя отвечал «да», «f» – если Петя отвечал «нет» на соответствующий вопрос, а 11-ый символ – ответ игры, число (вернее цифра) от 0 до 9 – количество правильных ответов в этой попытке. В выходном текстовом файле output.txt в первой строке должна находиться фраза ONE SOLUTION, если можно однозначно определить правильные ответы, и фраза POSSIBLE SOLUTION, если однозначно определить правильные ответы нельзя. Вторая строка должна содержать 10 символов «f» или «t» (подряд, без пробелов). При первом варианте решения вторая строка должна содержать последовательность ответов, при которой игра запускается. При втором варианте, вторая строка должна содержать последовательность ответов, при которой игра может запуститься.

Пример. Файл input.txt:

3

ttttttttt5

fffffffff5

ttttffffff0

output.txt:

ONE SOLUTION

fffffttttt

Задание 5. Шифр. Способ шифрования Цезаря состоит в следующем: буквы латинского алфавита нумеруются по порядку числами от 1 до 26 (A – 1, B – 2, C – 3, D – 4, E – 5, F – 6, G – 7, …, X – 24, Y – 25, Z – 26). Затем в шифруемом сообщении каждая буква с номером n в алфавите заменяется на $(n + k)$-ую букву алфавита (Цезарь использовал $k = 3$). Пробел считается 0-вой буквой алфавита, а сложение выполняется по модулю 27. Пусть k неизвестно. В этом случае для расшифровки такого сообщения необходимо перебрать 26 вариантов для различных значений k. Количество вариантов сокращается при наличии списка слов, которые могут встретиться в расшифрованном сообщении. Составить программу, которая расшифровывает сообщение, используя данный список слов. Программа считывает текстовый файл input.txt следующего формата: в первой строке находится целое число L $(1 \leq L \leq 100)$ – количество слов в списке. В следующих L строках – список слов, по одному слову в каждой строке (в каждом слове не более 20 символов, как известно Цезарь был краток). В следующей строке – зашифрованное сообщение из не более чем 200 символов. Список слов не обязательно содержит все слова из сообщения и наоборот, сообщение не обязательно содержит все слова из списка. Программа должна вывести текстовый файл output.txt следующего формата: расшифрованное сообщение, содержащее максимальное количество слов из входного списка (с учетом их повторений в зашифрованном сообщении). Если решений несколько, то вывести одно из них. Зашифрованное и расшифрованное сообщения состоят только из заглавных латинских букв и пробелов. Пример. Файл input.txt:

8

THIS

DAWN

THAT

THE

ZORRO

OTHER

AT

THING

BUUBDLA PSSPABUAEBXO

Файл output.txt:

ATTACK ZORRO AT DAWN