

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
(ФГБОУ ВО «ВятГУ»)
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчет по учебной практике

Выполнил студент группы ИВТ-11 _____/Рзаев А. Э./
Проверил преподаватель _____/Вожегов Д. В./

Киров 2016

Содержание

1	Кратчайший путь двух коней.....	3
1.1	Технический регламент.....	3
1.2	Пример.....	3
1.3	Решение.....	3
2	Почтальон.....	4
2.1	Технический регламент.....	4
2.2	Пример.....	5
2.3	Решение.....	5
3	Компоненты связности.....	6
3.1	Технический регламент.....	6
3.2	Пример.....	6
3.3	Решение.....	6
4	К-ичные числа.....	7
4.1	Технический регламент.....	7
4.2	Пример.....	7
4.3	Решение.....	7
5	Две строки.....	8
5.1	Технический регламент.....	8
5.2	Пример.....	8
5.3	Решение.....	8
6	Выводы.....	9

1 Кратчайший путь двух коней

Необходимо перевести каждого из двух коней на шахматной доске из одной клетки в другую за наименьшее число ходов. Два коня не могут одновременно находиться в одной клетке.

1.1 Технический регламент

Во входном файле knight2.in записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

Программа должна вывести в выходной файл knight2.out последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений.

1.2 Пример

knight2.in	knight2.out
a1	1 b3
c2	1 d4
c2	2 a1
a1	1 c2

1.3 Решение

- 1) Считать данные из входного файла
- 2) В графе, вершинами которого являются положения двух коней, выполнить обход в ширину, запоминая для каждой вершины ее предка
- 3) Вывести результат, основываясь на информации о предках вершин графа

2 Почтальон

В городе Ω есть n площадей, соединенных улицами. При этом количество улиц не превышает ста тысяч и существует не более трех площадей, на которые выходит нечетное число улиц. Для каждой улицы известна ее длина. По улицам разрешено движение в обе стороны. В городе есть хотя бы одна улица. От любой площади до любой можно дойти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице так, чтобы длина его пути была наименьшей. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

2.1 Технический регламент

Первая строка входного файла `post.in` содержит натуральное число n – количество площадей в городе ($1 \leq n \leq 1000$). Далее следует n строк, задающих улицы. В i -ой из этих строк находится число m_i – количество улиц, выходящих из площади i . Далее следует m_i пар положительных чисел. В j -ой паре первое число – номер площади, в которую идет j -ая улица с i -ой площади, а второе число – длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улиц с площади на нее саму.

Все числа во входном файле не превосходят 10^5 .

Если решение существует, то в первую строку выходного файла `post.out` вывести одно число – количество улиц в исходном маршруте (считая первую и последнюю), а во вторую – номера площадей в порядке их посещения.

Если решений нет – вывести в выходной файл одно число -1.

Если решений несколько, вывести любое.

2.2 Пример

post.in	post.out
4	5
2 2 1 2 2	1 2 3 4 2 1
4 1 4 4 4 3 5 1 1	
2 2 5 4 8	
2 3 8 2 4	

2.3 Решение

- 1) Считать данные из входного файла
- 2) Представить площади и улицы между ними как неориентированный граф.
- 3) Посчитать, сколько вершин, из которых выходит нечетное число ребер:
 - Если их две, добавить между ними фиктивную улицу
 - Если их нечетное количество, решения нет, вывести -1. Алгоритм завершен
- 4) Выполнить рекурсивный обход графа глубину
 - a) Перебрать все ребра из текущей вершины
 - b) Каждое ребро удалить из графа, выполнить обход в глубину из второго конца ребра
 - c) Добавить вершину к результату
 - d) Если есть еще ребра, перейти к пункту a
- 5) Убрать фиктивное ребро
- 6) Вывести результат

3 Компоненты связности

Задан неориентированный граф с N вершинами и M ребрами ($1 \leq N \leq 20000$, $1 \leq M \leq 200000$). В графе отсутствуют петли и кратные ребра. Необходимо определить компоненты связности заданного графа.

3.1 Технический регламент

Граф задан во входном файле `connect.in` следующим образом: первая строка содержит числа N и M . Каждая из следующих M строк содержит описание ребра – два целых числа из диапазона от 1 до N – номера концов ребра.

На первой строке выходного файла `connect.out` вывести число L – количество компонент связности заданного графа. На следующей строке вывести N чисел из диапазона от 1 до L – номера компонент связности, которым принадлежат соответствующие вершины. Компоненты связности следует занумеровать от 1 до L произвольным образом.

3.2 Пример

connect.in	connect.out
4 2	2
1 2	1 1 2 2
3 4	

3.3 Решение

- 1) Считать данные из входного файла
- 2) Выполнить обход графа в глубину из каждой еще не посещенной вершины:
 - Отметить корневую вершину как принадлежащую к компоненте связности с очередным номером K
 - Каждую следующую вершину также отмечать как принадлежащую к компоненте связности K
- 3) Вывести результат

4 К-ичные числа

Даны N-значные числа в системе счисления с основанием K. Число считается правильным, если его K-ичная запись не содержит двух подряд идущих нулей. Например:

- 1010230 — правильное 7-значное число;
- 1000198 не является правильным числом;
- 0001235 — не 7-значное, а 4-значное число.

Даны числа N и K, вычислите количество правильных K-ичных чисел, состоящих из N цифр.

4.1 Технический регламент

Во входном файле input.txt даны числа N и K в десятичной записи, разделенные переводом строки

В единственной строке выходного файла output.txt вывести искомое количество в десятичной записи.

4.2 Пример

Input.txt	output.txt
2 10	90

4.3 Решение

- 1) Считать данные из входного файла
- 2) Вычислить результат по формуле:

$$S(n) = \begin{cases} (k-1)(S(n-1) + S(n-2)), & n > 2 \\ k-1, & n = 1 \\ k(k-1), & n = 2 \end{cases}$$

- 3) Вывести результат

5 Две строки

Дано две строки, α и β . Требуется узнать, где в строке α можно найти строку β как подстроку и выписать все такие позиции.

5.1 Технический регламент

В первой строке входного файла input.txt содержится строка α , во второй – строка β . Строки состоят только из строчных латинских букв, их длины не превосходят 100000.

В первой строке выходного файла output.txt вывести одно число – количество вхождений строки β в строку α . Во второй строке для каждого вхождения вывести номер символа в строке α , где начинается очередная строка β . Вхождения нужно выводить в возрастающем порядке.

5.2 Пример

Input.txt	output.txt
abacaba	2
aba	1 5

5.3 Решение

- 1) Считать данные из входного файла
- 2) Склеить строки α и β в одну: $\beta\#\alpha$
- 3) Для этой строки вычислить z-функцию
- 4) Просмотреть значения z-функции:
 - Если очередной элемент z_i равен по длине строке β , то добавить к результату значение $i - \text{len}_\beta$
- 5) Вывести результат

6 Выводы

В ходе решения задач были применены навыки и умения, полученные в курсе программирования и других предметов. Для выполнения данных работ потребовались знание языка программирования C++, умение минимизировать/оптимизировать алгоритм для уменьшения его времени выполнения, оценивать сложности алгоритмов, умение нестандартно мыслить и находить решение в трудных ситуациях. Закреплены математические навыки, изучены такие алгоритмы, как обход графа в глубину/ширину, поиска Эйлера пути, построения z-функции.