

$$P(c/d) = \frac{P(d/c)}{P(d)}, \quad (1)$$

где  $P(c/d)$  — вероятность, что сообщение  $d$  принадлежит классу  $c$ , именно её нам надо рассчитать;

$P(d/c)$  — вероятность встретить сообщение  $d$  среди всех сообщений класса  $c$ ;

$P(c)$  — безусловная вероятность встретить сообщение класса  $c$  в базе сообщений;

$P(d)$  — безусловная вероятность сообщения  $d$  в базе сообщений.

Приложение реализовано на платформе .Net с помощью языка C#.

Для верной классификации писем необходимо собрать статистическую информацию по предыдущим данным, которые заносятся в подключаемую к модулю базу данных. Любое новое письмо будет проверено на наличие спама, используя указанную теорему.

#### ЛИТЕРАТУРА

1. Спам [Электронный ресурс] / Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B0%D0%>. – Дата доступа: 01.04.2017.

2. Борьба со спамом: история и методы [Электронный ресурс] – Режим доступа: [https://mipt.ru/dmcp/student/diff\\_articles/no\\_spam.php](https://mipt.ru/dmcp/student/diff_articles/no_spam.php). – Дата доступа: 03.04.2017.

3. Методы борьбы со спамом [Электронный ресурс] – Режим доступа: <http://www.securelist.com/ru/threats/spam?chapter=157> – Дата доступа: 03.04.2017.

4. Герман, О. В. Экспертные системы: лабораторный практикум / О. В. Герман, Н. В. Батин. – Минск : БГУИР, 2003. – 75 с.

УДК 004.4

Студ. А.С. Бубель

Науч. рук. доц. Н.Н. Буснюк

(кафедра информационных систем и технологий, БГТУ)

#### **АЛГОРИТМ ДЛЯ НАХОЖДЕНИЯ ВСЕХ ВОЗМОЖНЫХ РЕШЕНИЙ В СЕТЕВОМ ГРАФИКЕ, ЗАДАННОМ В ДЕКАРТОВОЙ СИСТЕМЕ КООРДИНАТ**

**Цель и задачи.** Разработать алгоритм, который будет находить максимальное и минимальное расстояния между начальным узлом

сетевого графика и конечным узлом  $n$ , если возможно пройти это расстояние различными маршрутами [2]. Количество узлов  $n$  вводится вручную. Каждый узел задаётся в декартовой системе координат. Дуги между узлами создаются вручную.

**Практическое применение.** Алгоритм можно использовать в навигации и прокладке маршрутов, т.к. каждый узел сетевого графика имеет свои координаты на осях  $x$ ,  $y$ . [1]. Маршрут может разветвляться, а сам алгоритм выводит все возможные маршруты для заданного условия, максимальный и минимальный маршрут, а также их длины.

**Полученные результаты и их описание.** Вводится количество узлов сетевого графика, а также координаты  $x$ ,  $y$  для каждого из них. Инициализируется двумерный динамический массив  $M$ , в котором будут храниться решения. Создаются дуги между узлами. Производится запись решения в массив  $M$ , преждевременно для каждого маршрута проверяется: есть ли соответствующий переход из заданного узла функцией `isNum`. Если для узла создаётся больше 1 дуги - это ведёт к разветвлению маршрута: функция `new_str` производит копирование уже построенного решения до разветвления в новую строку массива  $M$ . Решения разветвляются. После произведённых вычислений в каждой строке двумерного массива  $M$  хранится полное и уникальное решение для данного сетевого графика. Каждый элемент строки массива представляет собой дугу, где номер столбца есть узел, от которого строится дуга к узлу, номер которого записан в этой ячейке массива. Производятся вывод маршрутов из массива. Вычисляются наибольший и наименьший маршруты путём складывания длин дуг, хранящихся в строке массива. Вычисление длины дуги производится в функции `length_s`. Производится вывод результатов выполнения операции. Освобождается память, выделенная под массив  $M$ .

**Пример 1.** Имеется сетевой график (рис. 1), состоящий из четырёх узлов и шести дуг в декартовой системе координат. В данном сетевом графике прослеживается всего три возможных маршрута:  $1 - 2 - 4$ ;  $1 - 2 - 3 - 4$ ;  $1 - 3 - 4$ . Введём известные нам данные в программу и получим результат (рис. 2).

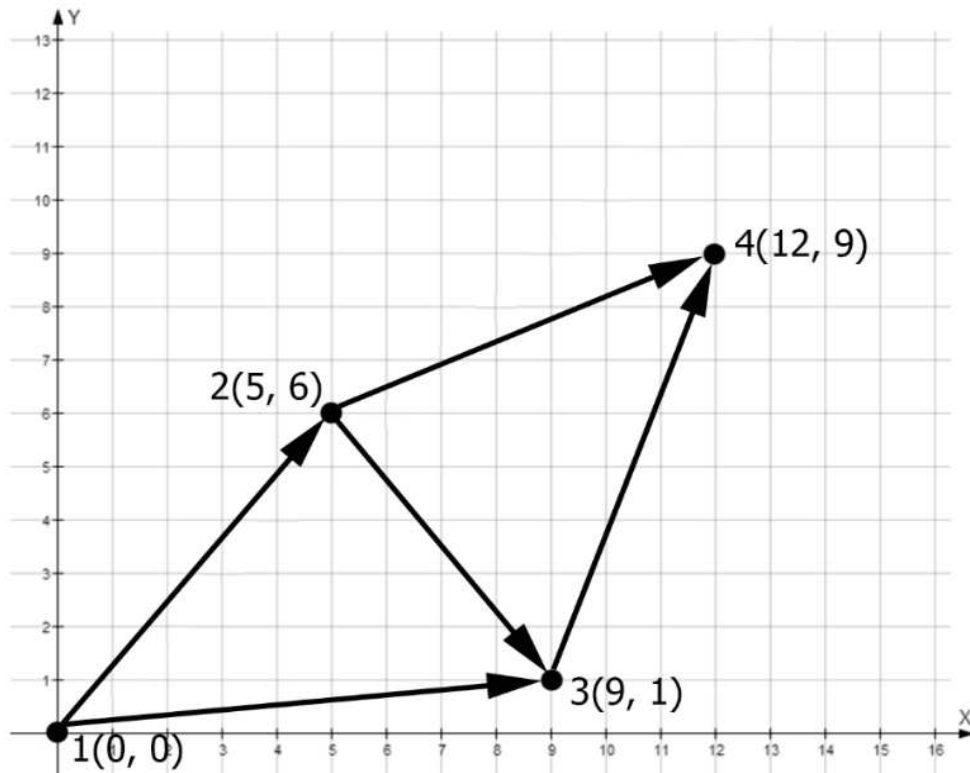


Рисунок 1 - Сетевой график, состоящий из четырёх узлов и пяти дуг.

```

Все возможные пути:
1 > 2 > 3 > 4
1 > 3 > 4
1 > 2 > 4

Решение, имеющее максимально возможную длину пути: 1 > 2 > 3 > 4
Длина пути: 22.7574

Решение, имеющее минимально возможную длину пути: 1 > 2 > 4
Длина пути: 15.426
    
```

Рисунок 2 - Вывод результата в консольном приложении Win32.

**Пример 2.** Имеется сетевой график, состоящий из десяти узлов и шестнадцати дуг в декартовой системе координат (рис. 3). В данной сетевом графике существует 8 возможных маршрутов: 1 – 10; 1 – 2 – 3 – 5 – 8 – 9 – 10; 1 – 2 – 3 – 5 – 7 – 9 – 10; 1 – 2 – 3 – 8 – 9 – 10; 1 – 2 –

10; 1 – 2 – 4 – 7 – 9 – 10; 1 – 2 – 4 – 6 – 9 – 10 ; 1 – 2 – 4 – 6 - 10. Введём известные нам данные в программу и получим результат (рис. 4).

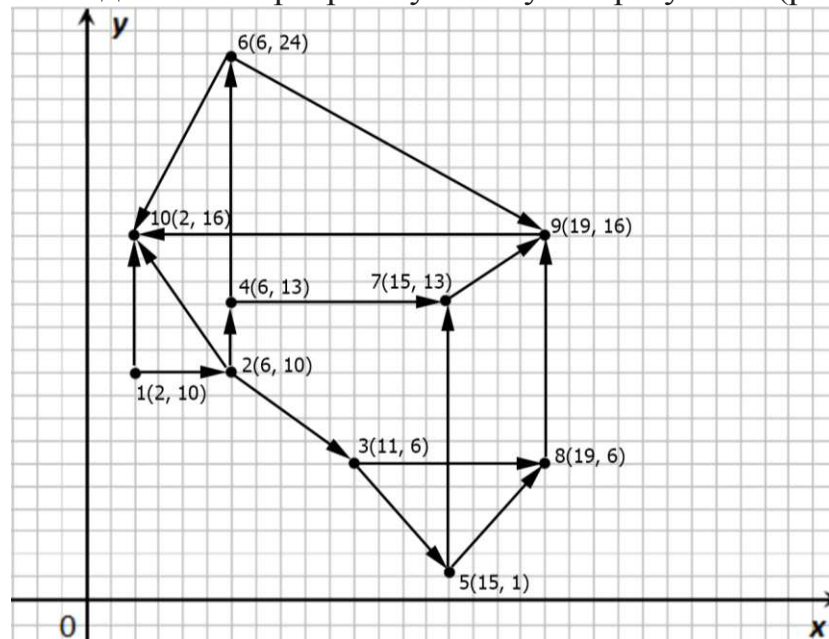


Рисунок 3 - Сетевой график, состоящий из десяти узлов и шестнадцати дуг

```

Все возможные пути:
1 > 2 > 3 > 5 > 7 > 9 > 10
1 > 10
1 > 2 > 4 > 6 > 9 > 10
1 > 2 > 10
1 > 2 > 3 > 8 > 9 > 10
1 > 2 > 4 > 7 > 9 > 10
1 > 2 > 3 > 5 > 8 > 9 > 10
1 > 2 > 4 > 6 > 10

Решение, имеющее максимально возможную длину пути: 1 > 2 > 3 > 5 > 7 > 9 > 10
Длина пути: 50.8062

Решение, имеющее минимально возможную длину пути: 1 > 10
Длина пути: 6
    
```

Рисунок 4 - Вывод результата в консольном приложении Win32

### ЛИТЕРАТУРА

1. Пацей Н.В. Основы алгоритмизации и программирования, Минск, 2010г.
- Буснюк Н.Н. Метод оптимального решения задачи о назначениях в сетевом планировании, Минск, 2016г.