

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ**  
**ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г.**  
**ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных  
систем

**ЛАБОРАТОРНАЯ РАБОТА №6**  
**Дисциплина: Теория надежности**  
**Тема: Анализ надежности сложной резервируемой системы**

Выполнил: ст. группы ВТ-31  
Подкопаев Антон Валерьевич  
Проверил: доц. каф. ПО и ВТАС  
Кабалянц Петр Степанович

**Белгород 2020**

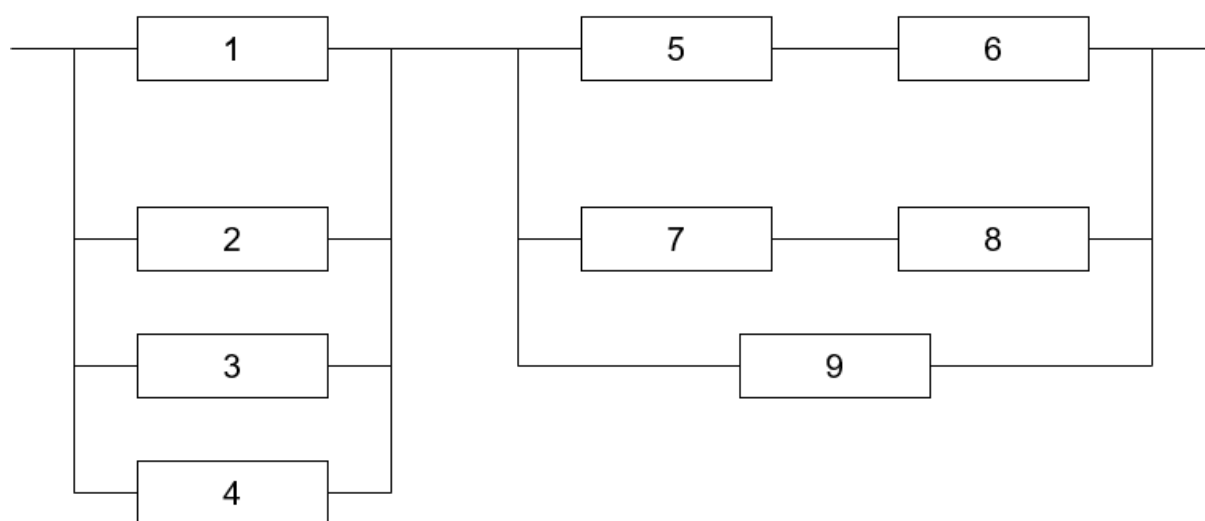
## Задания для выполнения к работе

1. Построить структурную схему системы.
2. Выполнить анализ системы методом, основанным на теоремах теории вероятности.
3. Выполнить анализ системы методом минимальных путей и минимальных сечений.
4. Составить для системы функцию алгебры логики.
5. Применить алгоритм разрезания для вычисления вероятности безотказной работы системы.
6. Составить программу, моделирующую работу системы при разном режиме обслуживания и случайном времени отказов и восстановлений.

## Ход выполнения работы

### Вариант 9

Система состоит из 3 узлов: элемент 1 резервирован с кратностью  $m=3$ , а элементы 2 и 3 с кратностью  $m=3/2$ .



Вероятность отказа узла равна произведению вероятности отказа основного и дублирующего элемента, тогда вероятность безотказной работы узла

$$P_1 = 1 - q_1 * q_2 q_3 q_4 \quad P_2 = 1 - ((1 - p_5 p_6) * (1 - p_7 p_8) q_9)$$

Вероятность безотказной работы системы равна произведению этих вероятностей:

$$P = (1 - q_1 * q_2 q_3 q_4)(1 - ((1 - p_5 p_6) * (1 - p_7 p_8) q_9)),$$

а вероятность отказа:

$$Q = 1 - P.$$

Минимальные пути:

1-5-6, 1-7-8, 1-9, 2-5-6, 2-7-8, 2-9, 3-5-6, 3-7-8, 3-9, 4-5-6, 4-7-8, 4-9;

Минимальные сечения:

1-2-3-4, 5-7-9, 5-8-9, 6-7-9, 6-8-9.

Функция работоспособности:

$$y = x_1 x_5 x_6 \vee x_1 x_7 x_8 \vee x_1 x_9 \vee x_2 x_5 x_6 \vee x_2 x_7 x_8 \vee x_2 x_9 \vee x_3 x_5 x_6 \vee x_3 x_7 x_8 \vee x_3 x_9 \\ \vee x_4 x_5 x_6 \vee x_4 x_7 x_8 \vee x_4 x_9$$

Функция неработоспособности:

$$y = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_5 \bar{x}_7 \bar{x}_9 \vee \bar{x}_5 \bar{x}_8 \bar{x}_9 \vee \bar{x}_6 \bar{x}_7 \bar{x}_9 \vee \bar{x}_6 \bar{x}_8 \bar{x}_9$$

```
C:\Users\D4rkn\AppData\Local\Programs\Python\Python38-32\python.exe

Анализ системы на основе теорем теории вероятности: P = (1-q0q1)(1-q2q3)(1-q4q5) = 0.5019074560000001; Q = 1 - P = 0.4980925439999999

Анализ системы методом минимальных путей и сечений:
Минимальные пути: [{1, 5, 6}, {8, 1, 7}, {1, 9}, {2, 5, 6}, {8, 2, 7}, {9, 2}, {3, 5, 6}, {8, 3, 7}, {9, 3}, {4, 5, 6}, {8, 4, 7}, {9, 4}]
Минимальные сечения: [{1, 2, 3, 4}, {9, 5, 7}, {8, 9, 5}, {9, 6, 7}, {8, 9, 6}]

Вероятность отказа методом минимальных путей и сечений: Q = q1q2q3q4 + q5q7q9 + q5q8q9 + q6q7q9 + q6q8q9 - q1q2q3q4q5q7q9 - q1q2q3q4q5q8q9 - q1q2q3q4q6q7q9 - q1q2q3q4q6q8q9 - q5q7q8q9 - q5q6q7q9 - q5q6q7q8q9 - q5q6q8q9 - q6q7q8q9 + q1q2q3q4q5q7q8q9 + q1q2q3q4q5q6q7q9 + q1q2q3q4q5q6q7q8q9 + q1q2q3q4q5q6q8q9 + q1q2q3q4q6q7q8q9 + q5q6q7q8q9 + q5q6q7q8q9 + q5q6q7q8q9 + q5q6q7q8q9 - q1q2q3q4q5q6q7q8q9 - q1q2q3q4q5q6q7q8q9 - q1q2q3q4q5q6q7q8q9 - q5q6q7q8q9 + q1q2q3q4q5q6q7q8q9
Q = 0.49809254400000014

Теоретически: P = 0.987980360274256; Q = 0.012019639725744047
Вероятность безотказной работы в результате эксперимента: 0.989
Вероятность отказа в результате эксперимента: 0.011

Press any key to continue . . .
```

```

import numpy
import random
from itertools import combinations

n = 9 # количество элементов
P = []
Q = []
for i in range(0, 9):
    tmp = random.random() * 0.1
    P.append(1 - tmp)
    Q.append(tmp)
    P.append(1 - tmp)
    Q.append(tmp)

P = [0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4]
Q = [0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6]

print("Вероятности безотказной работы элементов P = {}\nВероятности отказов Q = {}\n".format(P, Q))

min_ways = [{1, 5, 6}, {1, 7, 8}, {1,9}, {2, 5, 6}, {2, 7, 8}, {2,9},
            {3, 5, 6}, {3, 7, 8}, {3,9}, {4, 5, 6}, {4, 7, 8}, {4,9},]
min_cuts = [{1, 2, 3, 4}, {5, 7, 9}, {5, 8, 9}, {6, 7, 9}, {6, 8, 9}]

P_tv = (1 - Q[0]*Q[1]*Q[2]*Q[3])*(1 - ((1 - P[4]*P[5])*(1 - P[6]*P[7]))*Q[8])
Q_tv = 1 - P_tv
print("Анализ системы на основе теорем теории вероятности: P = (1-q0q1)(1-q2q3)(1-q4q5) = {}; Q = 1 - P = {}\n".format(P_tv, Q_tv))

print("Анализ системы методом минимальных путей и сечений:")
print("Минимальные пути: {}\nМинимальные сечения: {}\n".format(min_ways, min_cuts))

def getStrExpression(combinations, ch, sign):
    res_str = ""
    for comb in combinations:
        res_str += " " + sign + " "
        for j in range(n):
            if j in comb:
                res_str += "{}{}".format(ch, j)

    return res_str

def uniCombinations(combinations, k, set_):
    res = []
    for comb in combinations:
        t = set()
        for j in range(len(set_)):
            if j in comb:
                t = t | set_[j]
        res.append(t)
    return res

def computeCombinations(combinations, p_q):
    res = 0
    for comb in combinations:
        t = 1
        for j in range(n):
            if j in comb:
                t *= p_q[j]
        res += t
    return res

ways_num = [i for i in range(0, len(min_ways))]
exp_str = ""
exp_res = 0
for k in range(1, len(min_ways) + 1):

```

```

comb = list(combinations(ways_num, k))
comb = uniCompinations(comb, k, min_ways)
if (k % 2 != 0):
    exp_res += computeCombinations(comb, P)
else:
    exp_res -= computeCombinations(comb, P)

str_ = getStrExpression(comb, "p", "+" if (k % 2 != 0) else "-")
exp_str += str_
exp_str = "P =" + exp_str[2:]
#print("Вероятность безотказной работы методом минимальных путей и сечений: {}".format(exp_str))
#print("P = {}\n".format(exp_res))

cuts_num = [i for i in range(0, len(min_cuts))]
exp_str = ""
exp_res = 0
for k in range(1, len(min_cuts) + 1):
    comb = list(combinations(cuts_num, k))
    comb = uniCompinations(comb, k, min_cuts)
    if (k % 2 != 0):
        exp_res += computeCombinations(comb, Q)
    else:
        exp_res -= computeCombinations(comb, Q)

    str_ = getStrExpression(comb, "q", "+" if (k % 2 != 0) else "-")
    exp_str += str_
exp_str = "Q =" + exp_str[2:]
print("Вероятность отказа методом минимальных путей и сечений: {}".format(exp_str))
print("Q = {}\n".format(exp_res))

#####

def schemFunc(X):
    res = X[0] and X[2] and X[4] or X[0] and X[2] and X[5] or X[0] and X[3] and X[4] or X[0] and
X[3] and X[5]
    res = res or X[1] and X[2] and X[4] or X[1] and X[2] and X[5] or X[1] and X[3] and X[4] or X[1]
and X[3] and X[5]
    return res

def step():
    res = []
    for i in range(n):
        res.append(False if random.random() > P[i] else True)
    return res

for k in range(1):
    if (i != 0):
        P = []
        Q = []
        for i in range(0, int(n/2)):
            tmp = random.random() * 0.1
            P.append(1 - tmp)
            Q.append(tmp)
            P.append(1 - tmp)
            Q.append(tmp)
        P_tv = (1 - Q[0]**2)*(1 - Q[2]**2)*(1-Q[4]**2)
        Q_tv = 1 - P_tv
        print("Теоретически: P = {}; Q = {}".format(P_tv, Q_tv))

N = 1000
ok = 0
for i in range(N):
    X = step()
    if (schemFunc(X)):
        ok += 1
print("Вероятность безотказной работы в результате эксперимента: ", ok/N)
print("Вероятность отказа в результате эксперимента: ", (N-ok)/N, "\n")

```