20

19

18

17

16

27

26

25

24

23

22

21

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

1. **Определение цикломатической сложности.**

32 дуги – 20 узлов +2=14

6 предикатных узлов +1=7

Потоковый граф имеет 5 региона.

1. **Определение базового множества независимых линейных путей.**

**Путь 1:** 1-2-3-5-14-21-22-23-27-16-17-20;

**Путь 2:** 1-2-3-5-14-21-22-23-27-16-18-19-20;

**Путь 3:** 1-2-3-5-14-21-22-23-24-25-26-23-27-16-18-19-20;

**Путь 4:** 1-2-3-5-14-15-14-21-22-23-24-25-26-23-27-16-18-19-20;

**Путь 5:** 1-2-3-5-6-7-8-9-11-13-5-14-15-14-21-22-23-24-25-26-23-27-26-18-19-20;

**Путь 6:** 1-2-3-5-6-7-8-9-10-6-7-8-9-11-12-6-7-8-9-11-13-5-14-15-14-21-22-23-24-25-26-23-27-16-18-19-20;

**Путь 7:** 1-2-3-4-3-5-6-7-8-9-10-6-7-8-9-11-12-6-7-8-9-11-13-5-14-15-14-21-22-23-24-25-26-23-27-16-18-19-20.

1. **Подготовка тестовых вариантов.**

**Тестовый вариант№1**

public void test\_search\_win\_default()

{

Dictionary<int, int> Castle = new Dictionary<int, int> {{ 1, 5 }, { 2, 3 }, { 3, 4}, { 4, 1 }, { 5, 3 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 4 }, { 10, 5 }};

Assert.AreEqual(true, Program.SeachWin(Castle));

}

**Тестовый вариант№2**

public void test\_search\_win\_failure()

{

Dictionary<int, int> Castle = new Dictionary<int, int> { { 1, 5 }, { 2, 3 }, { 3, 4}, { 4, 1 }, { 5, 3 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 4 }, { 10, 5 }};

Assert.AreEqual(true, Program.SeachWin(Castle));

}

**Тестовый вариант№3**

public void test\_search\_win\_feilure1()

{

Dictionary<int, int> Castle = new Dictionary<int, int> { { 1, 5 }, { 2, 5 }, { 3, 5}, { 4, 5 }, { 5, 5 }, { 6, 5 }, { 7, 5 }, { 8, 5 }, { 9, 5}, { 10, 5 }};

Assert.AreEqual(false, Program.SeachWin(Castle));

}

**Тестовый вариант№4**

public void test\_search\_win\_default1()

{

Dictionary<int, int> Castle = new Dictionary<int, int> { { 1, 5 }, { 2, 1 }, { 3, 4}, { 4, 1 }, { 5, 3 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 5 }, { 10, 5 }};

Assert.AreEqual(true, Program.SeachWin(Castle));

}

**Тестовый вариант №5**

public void test\_search\_win\_default2()

{

Dictionary<int, int> Castle = new Dictionary<int, int> { { 1, 1 }, { 2, 2 }, { 3, 4}, { 4, 4 }, { 5, 3 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 1 }, { 10, 5 }};

Assert.AreEqual(true, Program.SeachWin(Castle));

}

**Тестовый вариант №6**

public void test\_search\_win\_default3()

{

Dictionary<int, int> Castle = new Dictionary<int, int> { { 1, 1 }, { 2, 1 }, { 3, 4}, { 4, 4 }, { 5, 2 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 1 }, { 10, 5 }};

Assert.AreEqual(true, Program.SeachWin(Castle));

}

**Тестовый вариант №7**

public void test\_search\_win\_default4()

{

Dictionary<int, int> Castle = new Dictionary<int, int> { { 1, 1 }, { 2, 1 }, { 3, 3}, { 4, 4 }, { 5, 2 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 1 }, { 10, 5 }};

Assert.AreEqual(true, Program.SeachWin(Castle));

}

**Таблица для тестовых вариантов**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ теста** | **Тестовая строка** | **Эталонный вариант max** | **Эталонный вариант**  **min** | **Полученный результат max** | **Полученный результат**  **min** |
| **1** | { 1, 5 }, { 2, 3 }, { 3, 4}, { 4, 1 }, { 5, 3 },{6, 4 }, { 7, 2 }, { 8, 1}, { 9, 4 }, { 10, 5 } | true | false | true | false |
| **2** | { 1, 5 }, { 2, 3 }, { 3, 4}, { 4, 1 },{ 5, 3 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 4 }, { 10, 5 } | true | false | true | false |
| **3** | { 1, 5 }, { 2, 5 }, { 3, 5}, { 4, 5 }, { 5, 5 }, { 6, 5 }, { 7, 5 }, { 8, 5 }, { 9, 5}, { 10, 5 } | false | true | false | true |
| **4** | { 1, 5 }, { 2, 1 }, { 3, 4}, { 4, 1 }, { 5, 3 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 5 }, { 10, 5 } | true | false | true | false |
| **5** | { 1, 1 }, { 2, 2 }, { 3, 4}, { 4, 4 }, { 5, 3 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 1 }, { 10, 5} | true | false | true | false |
| **6** | { 1, 1 }, { 2, 1 }, { 3, 4}, { 4, 4 }, { 5, 2 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 1 }, { 10, 5 } | true | false | true | false |
| **7** | { 1, 1 }, { 2, 1 }, { 3, 3}, { 4, 4 }, { 5, 2 }, { 6, 4 }, { 7, 2 }, { 8, 1 }, { 9, 1 }, { 10, 5 } | true | false | true | false |

**Вывод:** Тестирование прошло успешно.