СОФТВЕРСКИ КВАЛИТЕТ И ТЕСТИРАЊЕ

ДОМАШНА ЗАДАЧА 4

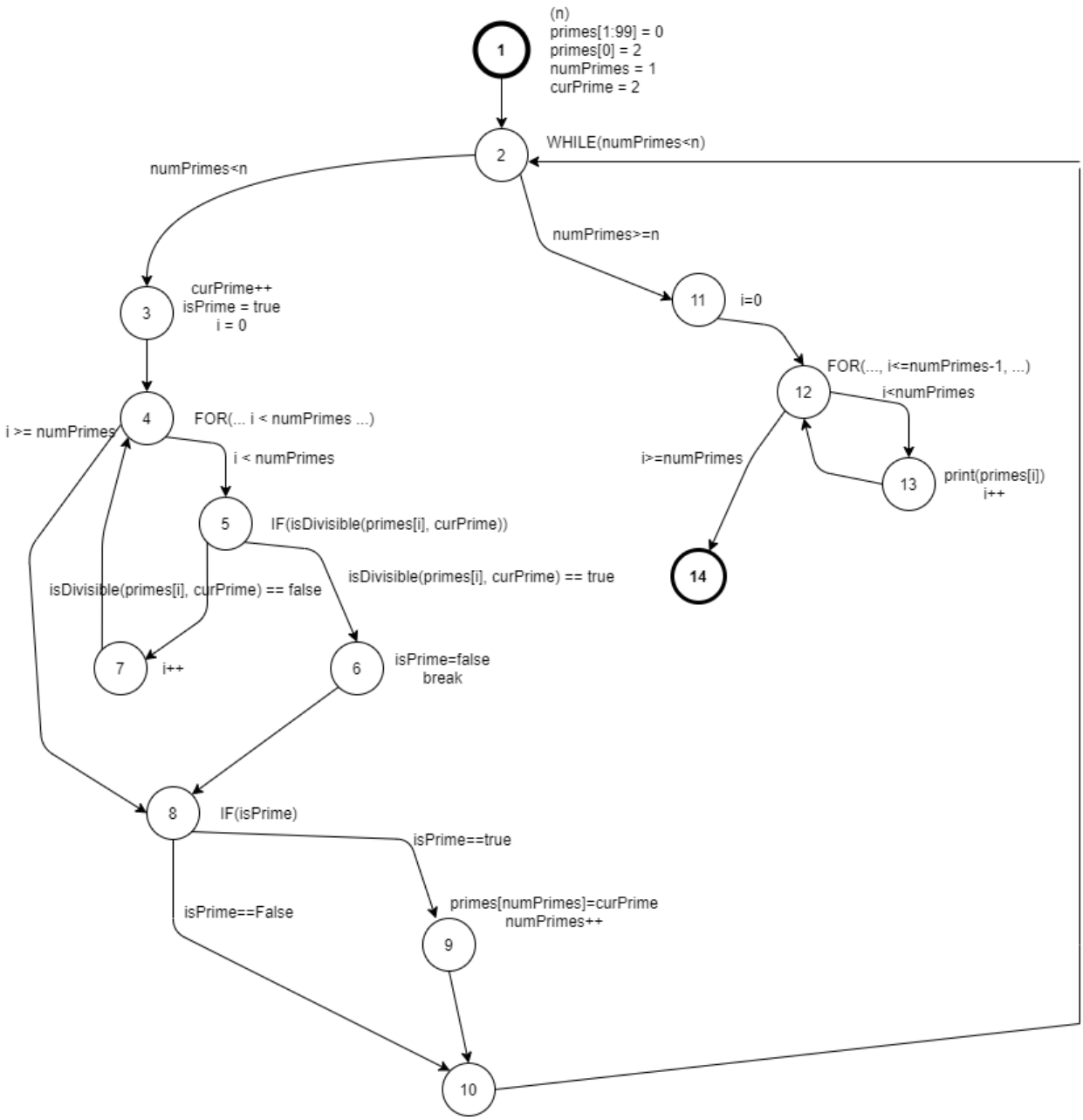
Фисник Лимани, 151027

Го имаме функцијата:

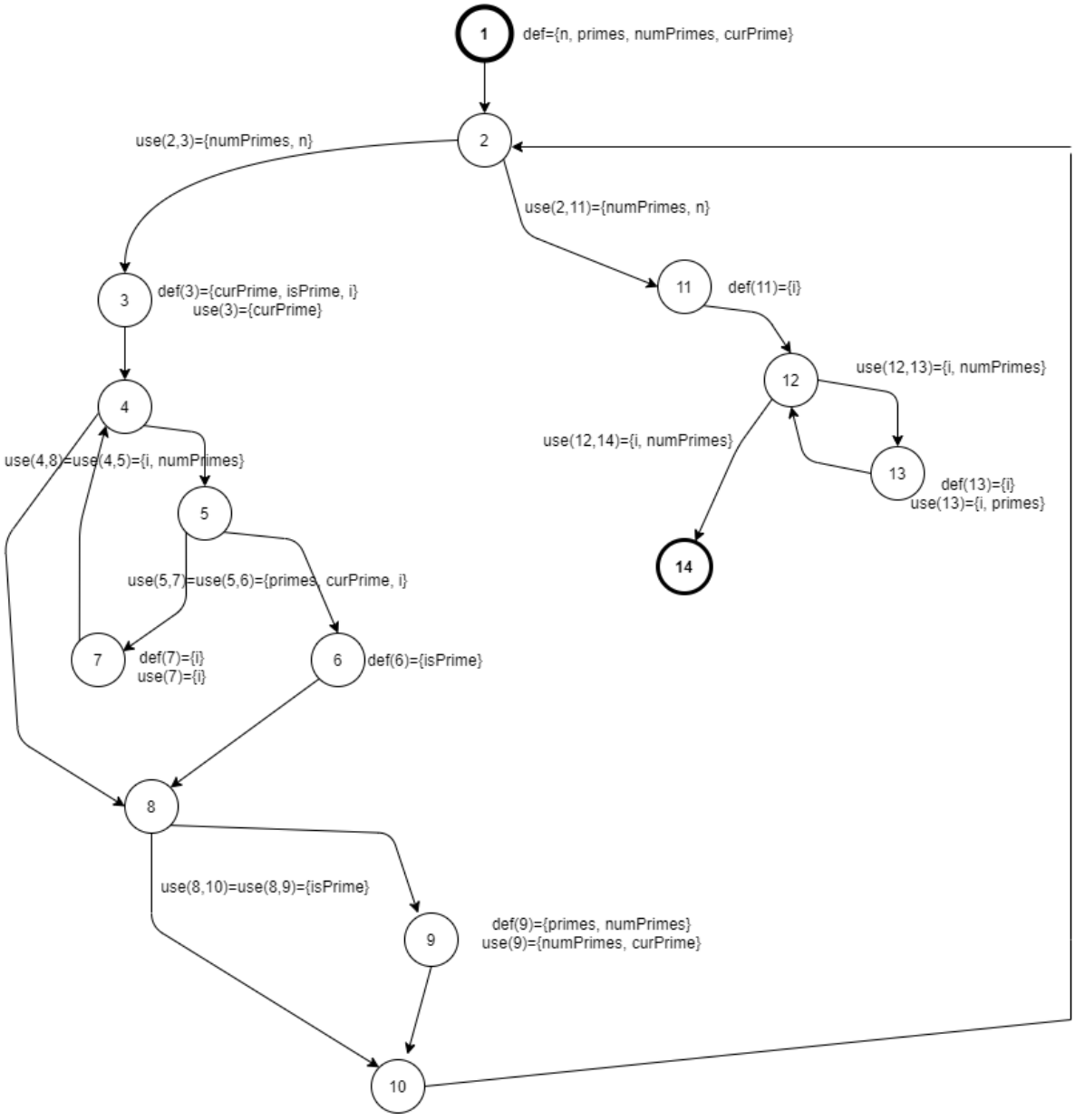
```
private static void printPrimes (int n)
{
    int curPrime;           // Value currently considered for primeness
    int numPrimes;          // Number of primes found so far.
    boolean isPrime;        // Is curPrime prime?
    int [] primes = new int [100]; // The list of prime numbers.

    // Initialize 2 into the list of primes.
    primes [0] = 2;
    numPrimes = 1;
    curPrime  = 2;
    while (numPrimes < n)
    {
        curPrime++;  // next number to consider ...
        isPrime = true;
        for (int i = 0; i <= numPrimes-1; i++)
        {   // for each previous prime.
            if (isDivisible (primes[i], curPrime))
            {  // Found a divisor, curPrime is not prime.
                isPrime = false;
                break; // out of loop through primes.
            }
        }
        if (isPrime)
        {   // save it!
            primes[numPrimes] = curPrime;
            numPrimes++;
        }
    }  // End while

    // Print all the primes out.
    for (int i = 0; i <= numPrimes-1; i++)
    {
        System.out.println ("Prime: " + primes[i]);
    }
}  // end printPrimes
```

Source: https://cs.gmu.edu/~offutt/softwaretest/java/PrintPrimes.java

ГРАФОТ АНОТИРАН СО ИНСТРУКЦИИТЕ ОД ИЗВОРНИОТ КОД



КОНВЕРТИРАЊЕ НА АНОТАЦИИТЕ ВО *DEF* И *USE* МНОЖЕСТВА



ТАБЕЛИТЕ ЗА *DEFS* И *USES*

NODE	DEF	USE
1	{n, primes, numPrimes, curPrime}	
2		
3	{curPrime, isPrime, i}	{curPrime}
4		
5		
6	{isPrime}	
7	{i}	{i}
8		
9	{primes, numPrimes}	{numPrimes, curPrime}
10		
11	{i}	
12		
13	{i}	{i, primes}
14		

	EDGE	**USE**
1.	(1,2)	
2.	(2,3)	{numPrimes, n}
3.	(2,11)	{numPrimes, n}
4.	(3,4)	
5.	(4,5)	{i, numPrimes}
6.	(4,8)	{i, numPrimes}
7.	(5,6)	{primes, curPrime, i}
8.	(5,7)	{primes, curPrime, i}
9.	(6,8)	

	EDGE	**USE**
10.	(7,4)	
11.	(8,9)	{isPrime}
12.	(8,10)	{isPrime}
13	(9, 2)	
14	(10, 2)	
15	(11, 12)	
16	(12, 13)	{i, numPrimes}
17	(12, 14)	{i, numPrimes}
18	(13, 12)	

DU ПАТИШТА

VARIABLE	DU Pairs	DU Paths
n	(1, (2,3)) (1, (2,11))	[1,2,3] [1,2,11]
primes	(1, (5, 6)) (1, (5, 7)) (1, 13) (9, 13) (9, (5, 6)) (9, (5,7))	[1,2,3,4,5,6] [1,2,3,4,5,7] [1, 2, 11, 12, 13] [9, 2, 11, 12, 13] [9, 10, 2, 3, 4, 5, 6] [9, 10, 2, 3, 4, 5, 7]
numPrimes	(1, (2,3)) (1, (2,11)) (1, (4,5)) (1, (4,8)) (1, 9) (1, (12,13)) (1, (12,14)) (9, 9) (9, (2,3)) (9, (2,11)) (9, (4,5)) (9, (4,8)) (9, (12,13)) (9, (12,14))	[1,2,3] [1,2,11] [1,2,3,4,5] [1,2,3,4,8] [1,2,3,4,5,6,8,9] [1,2,3,4,8,9] [1, 2, 11, 12, 13] [1, 2, 11, 12, 14] [9, 10, 2, 3, 4, 5, 6, 8 ,9] [9, 10, 2, 3, 4, 8, 9] [9, 10, 2, 3] [9, 10, 2, 11] [9, 10, 2, 3, 4, 5] [9, 10, 2, 3, 4, 8] [9, 10, 2, 11, 12, 13] [9, 10, 2, 11, 12, 14]
curPrime	(1,3) (3,3) (3, (5,6)) (3, (5,7)) (3, 9)	[1, 2, 3] [3, 4, 5, 6, 8, 9, 10, 2, 3] [3, 4, 8, 9, 10, 2, 3] [3, 4, 5, 6] [3, 4, 5, 7] [3, 4, 5, 6, 8, 9] [3, 4, 8, 9]

isPrime	(3, (8,9)) (3, (8,10)) (6, (8,9) (6, (8, 10))	[3, 4, 8, 9] [3, 4, 8, 10] [6, 8, 9] [6, 8, 10]
i	(3, (4,5)) (3, (4,8)) (3, (5,6)) (3, (5,7)) (3, 7) (7,7) (7, (4,5)) (7, (4,8)) (7, (5,6)) (7, (5,7)) (11, (12,13)) (11, (12,14)) (11, 13) (13,13) (13, (12,13)) (13, (12,14))	[3, 4, 5] [3, 4, 8] [3, 4, 5, 6] [3, 4, 5, 7] [3, 4, 5, 7] [7, 4, 5, 7] [7,4,5] [7,4,8] [7,4,5,6] [7,4,5,7] [11, 12, 13] [11, 12, 14] [11, 12, 13] [13, 12, 13] [13,12,13] [13,12,14]

Имаме вкупно 51 DU-патишта:

[1,2,3]	[9, 10, 2, 3]	[3, 4, 5]
[1,2,11]	[9, 10, 2, 11]	[3, 4, 8]
	[9, 10, 2, 3, 4, 5]	[3, 4, 5, 6]
[1,2,3,4,5,6]	[9, 10, 2, 3, 4, 8]	[3, 4, 5, 7]
[1,2,3,4,5,7]	[9, 10, 2, 11, 12, 13]	[3, 4, 5, 7]
[1, 2, 11, 12, 13]	[9, 10, 2, 11, 12, 14]	[7, 4, 5, 7]
[9, 2, 11, 12, 13]		[7,4,5]
[9, 10, 2, 3, 4, 5, 6]	[1, 2, 3]	[7,4,8]
[9, 10, 2, 3, 4, 5, 7]	[3, 4, 5, 6, 8, 9, 10, 2, 3]	[7,4,5,6]
	[3, 4, 8, 9, 10, 2, 3]	[7,4,5,7]
[1,2,3]	[3, 4, 5, 6]	[11, 12, 13]
[1,2,11]	[3, 4, 5, 7]	[11, 12, 14]
[1,2,3,4,5]	[3, 4, 5, 6, 8, 9]	[11, 12, 13]
[1,2,3,4,8]	[3, 4, 8, 9]	[13, 12, 13]
[1,2,3,4,5,6,8,9]		[13,12,13]
[1,2,3,4,8,9]	[3, 4, 8, 9]	[13,12,14]
[1, 2, 11, 12, 13]	[3, 4, 8, 10]	
[1, 2, 11, 12, 14]	[6, 8, 9]	
[9, 10, 2, 3, 4, 5, 6, 8 ,9]	[6, 8, 10]	
[9, 10, 2, 3, 4, 8, 9]		

Ги подредуваме DU-патиштата (цртата на средината на патот значи дека е дупликат):

[1,2,3]	[3,4,5,7]	[9,10,2,3]
~~[1,2,3]~~	~~[3,4,5,7]~~	[9,10,2,11]
~~[1,2,3]~~	~~[3,4,5,7]~~	[9,10,2,3,4,5]
[1,2,11]	[3,4,8,9]	[9,10,2,3,4,8]
~~[1,2,11]~~	~~[3,4,8,9]~~	[9,10,2,3,4,5,6]
[1,2,3,4,5]	[3,4,8,10]	[9,10,2,3,4,5,7]
[1,2,3,4,8]	[3,4,5,6,8,9]	[9,10,2,3,4,8,9]
[1,2,3,4,5,6]	[3,4,8,9,10,2,3]	~~[9,10,2,11,12,13]~~
[1,2,3,4,5,7]	[3,4,5,6,8,9,10,2,3]	[9,10,2,11,12,13]
[1,2,3,4,8,9]		[9,10,2,11,12,14]
[1,2,11,12,13]	[6,8,9]	[9,10,2,3,4,5,6,8,9]
~~[1,2,11,12,13]~~	[6,8,10]	
[1,2,11,12,14]		[11,12,13]
[1,2,3,4,5,6,8,9]	[7,4,5]	~~[11,12,13]~~
	[7,4,8]	[11,12,14]
[3,4,5]	[7,4,5,6]	
[3,4,8]	[7,4,5,7]	[13,12,13]
[3,4,5,6]	~~[7,4,5,7]~~	~~[13,12,13]~~
~~[3,4,5,6]~~		[13,12,14]

Имаме 39 уникатни патишта:
(Р – значи дека патот е веќе потпат на друг (поширок) пат од истото множество на променливата која се испитува)

[1,2,3]	Р	[3,4,5,7]		[9,10,2,3]	Р
[1,2,11]	Р	[3,4,8,9]	Р	[9,10,2,11]	Р
[1,2,3,4,5]	Р	[3,4,8,10]		[9,10,2,3,4,5]	Р
[1,2,3,4,8]	Р	[3,4,5,6,8,9]	Р	[9,10,2,3,4,8]	Р
[1,2,3,4,5,6]	Р	[3,4,8,9,10,2,3]		[9,10,2,3,4,5,6]	Р
[1,2,3,4,5,7]		[3,4,5,6,8,9,10,2,3]		[9,10,2,3,4,5,7]	
[1,2,3,4,8,9]				[9,10,2,3,4,8,9]	
[1,2,11,12,13]		[6,8,9]		[9,10,2,11,12,13]	
[1,2,11,12,14]		[6,8,10]		[9,10,2,11,12,14]	
[1,2,3,4,5,6,8,9]				[9,10,2,3,4,5,6,8,9]	
		[7,4,5]	Р		
[3,4,5]	Р	[7,4,8]		[11,12,13]	
[3,4,8]	Р	[7,4,5,6]		[11,12,14]	
[3,4,5,6]	Р	[7,4,5,7]			
				[13,12,13]	
				[13,12,14]	

Ако ги тргаме и патиштата со “Р” тогаш ни остануваат само 23 патишта:

~~[1,2,3,4,5,7]~~	1	~~[3,4,5,7]~~	6	[9,10,2,3,4,5,7]*	15
[1,2,3,4,8,9]*	2	[3,4,8,10]*	7	[9,10,2,3,4,8,9]*	16
~~[1,2,11,12,13]~~	3	[3,4,8,9,10,2,3]*	8	~~[9,10,2,11,12,13]~~	17
[1,2,11,12,14]*	4	[3,4,5,6,8,9,10,2,3]*	9	[9,10,2,11,12,14]*	18
[1,2,3,4,5,6,8,9]*	5			[9,10,2,3,4,5,6,8,9]*	19
		[6,8,9]*	10		
		~~[6,8,10]~~	11	~~[11,12,13]~~	20
				[11,12,14]*	21
		~~[7,4,8]~~	12		
		~~[7,4,5,6]~~	13	~~[13,12,13]~~	22
		~~[7,4,5,7]~~	14	~~[13,12,14]~~	23

НЕВОЗМОЖНИ (INFEASIBLE) DU-ПАТИШТА:

1. [6, 8, 9] – не е можно `isPrime` да стане false, а потоа да помине `if(isPrime)`
2. [11, 12, 14] – Почетна вредност на i е 0 а на numPrimes е 1 па затоа не е можно да не помине условот i <= numPrimes – 1 барем еднаш
3. [3, 4, 8] – Почетна вредност на i е 0 а на numPrimes е 1, така да мора барем еднаш да се исполнува условот i < numPrimes
4. [9, 10, 2, 3, 4, 5, 7] – низата на прости броеви се иницијализира со 2, а проверката на броевите дали се прости или не започнува од бројот 3, па натаму не е можно да се движиме на подпатот [9, 10, ..] – што значи дека бројот е прост а потоа да си продолжиме со број кој не е делив со првиот прост број што е 2 (бидејќи за секој прост број поголем од 2 важи дека следен број е парен број, т.е. се дели со 2)
5. [1, 2, 3, 4, 5 ,6] – Овој пат е невозможен бидејќи 3 не се дели со 2.

Сите патишта со (*) во табелата се невозможни.

TEST CASE: n = 1
TEST PATHS: [1, 2, 11, 12, 13, 12, 14]
DU-PATHS COVERED: (3, 20, 23)

TEST CASE: n = 5
TEST PATHS: [1, 2, 3, 4, 5, 7, 4, 8, 9, 10, 2, 3, 4, 5, 6, 8, 10, 2, 3, 4, 5, 7, 4, 5, 7, 4, 8, 9, 10, 2, 3, 4, 5, 6, 8, 10, 2, 3, 4, 5, 7, 4, 5, 7, 4, 5, 7, 4, 8, 9, 10, 2, 3, 4, 5, 6, 8, 10, 2, 3, 4, 5, 7, 4, 5, 6, 8, 10, 3, 4, 5, 6, 8, 10, 2, 3, 4, 5, 7, 4, 5, 7, 4, 5, 7, 4, 5, 7, 4, 8, 9, 10, 2, 11, 12, 13, 12, 13, 12, 13, 12, 13, 12, 13, 12, 14]
DU-PATHS COVERED: (1, 6, 11, 12, 13, 14, 17, 20, 22, 23)

Бидејќи ја искористевме оптимизацијата дека ако еден пат е потпат од некој поширок пат од истото множество на променливата која се испитува, тогаш можеме да ги држиме само тие патишта кои немаат друг надпад, како што постапивме погоре.
Но, еден надпат може да е infeasible, а патиштата кои ги покрива не значи дека се infeasible.
Погоре видовме дека имаше голем број на невозможни патишта, и за да се осигураме дека сме ги покривале сите DU-патишта кои се возможни ќе се навратиме повторно кај сите основни патишта (без употреба на оптимизацијата) и да провериме дали два тестовите ги покриваат сите тие DU-патишта.

~~[1,2,3]~~	1	~~[3,4,5,7]~~	14	~~[9,10,2,3]~~	26
~~[1,2,11]~~	2	[3,4,8,9]*	15	~~[9,10,2,11]~~	27
~~[1,2,3,4,5]~~	3	[3,4,8,10]*	16	~~[9,10,2,3,4,5]~~	28
[1,2,3,4,8]*	4	[3,4,5,6,8,9]*	17	[9,10,2,3,4,8]*	29
[1,2,3,4,5,6]*	5	[3,4,8,9,10,2,3]*	18	~~[9,10,2,3,4,5,6]~~	30
~~[1,2,3,4,5,7]~~	6	[3,4,5,6,8,9,10,2,3]*	19	[9,10,2,3,4,5,7]*	31
[1,2,3,4,8,9]*	7			[9,10,2,3,4,8,9]*	32
~~[1,2,11,12,13]~~	8	[6,8,9]*	20	~~[9,10,2,11,12,13]~~	33
[1,2,11,12,14]*	9	~~[6,8,10]~~	21	[9,10,2,11,12,14]*	34
[1,2,3,4,5,6,8,9]*	10			[9,10,2,3,4,5,6,8,9]*	35
		~~[7,4,5]~~	22		
~~[3,4,5]~~	11	~~[7,4,8]~~	23	~~[11,12,13]~~	36
[3,4,8]*	12	~~[7,4,5,6]~~	24	[11,12,14]*	37
~~[3,4,5,6]~~	13	~~[7,4,5,7]~~	25		
				~~[13,12,13]~~	38
				~~[13,12,14]~~	39

TEST CASE: n = 1
TEST PATHS: [1, 2, 11, 12, 13, 12, 14]
DU-PATHS COVERED: (2, 8, 36, 39)

TEST CASE: n = 5

TEST PATHS: [1, 2, 3, 4, 5, 7 8 9 10 3 4 5 6 8 10 3 4 5 7 4 5 7 8 9 10 3 4 5 6 8 10 3 4 5 7 4 5 7 4 5 7 8 9 10 3 4 5 6 8 10 3 4 5 7 4 5 6 8 10 3 4 5 6 8 10 3 4 5 7 4 5 7 4 5 7 4 5 7 8 9 10 11 12 13 12 13 12 13 12 13 12 13 14]

DU-PATHS COVERED: (1, 3, 6, 11, 13, 14, 21, 22, 23, 24, 25, 26, 27, 28, 30, 33, 36, 38, 39)

Junit тестови:

```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class PrintPrimesTest {
    @Test
    public void test1(){
        int n = 1;
        String expected = "Prime: " + 2;
        assertEquals(expected, PrintPrimes.printPrimes(n));
    }

    @Test
    public void test2(){
        int n = 5;
        String expected;
        StringBuilder sb = new StringBuilder();
        sb.append("Prime: " + 2 + "\n");
        sb.append("Prime: " + 3 + "\n");
        sb.append("Prime: " + 5 + "\n");
        sb.append("Prime: " + 7 + "\n");
        sb.append("Prime: " + 11);
        expected = sb.toString();

        assertEquals(expected, PrintPrimes.printPrimes(n));
    }
}
```

Резултати: