

СКИТ – Лабораториска вежба 2

За функцијата **closestLowerPrime** потребно е да се направи покривање со помош на графови и да се дефинираат соодветните тест случаи (според Data Flow Criteria, на ист начин како примерот од вежби).

```
public static int closestLowerPrime(int n) {
    if(n <= 1)
        throw new IllegalArgumentException("The number has to be greater than 1.");

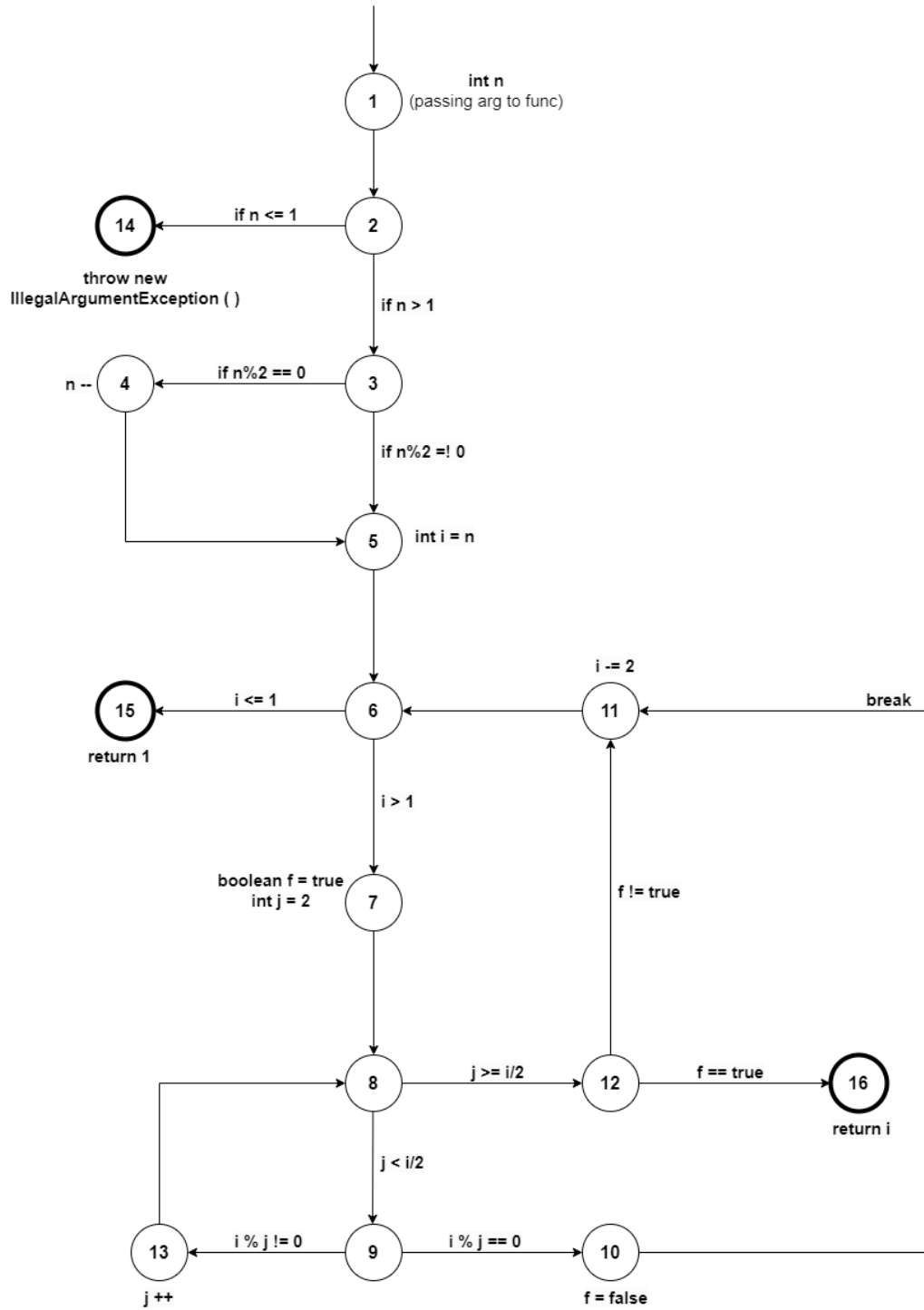
    if(n%2 == 0)
        n--;

    for(int i=n;i>1;i-=2) {
        boolean f=true;
        for(int j=2;j<=i/2;j++) {
            if(i%j == 0) {
                f = false;
                break;
            }
        }

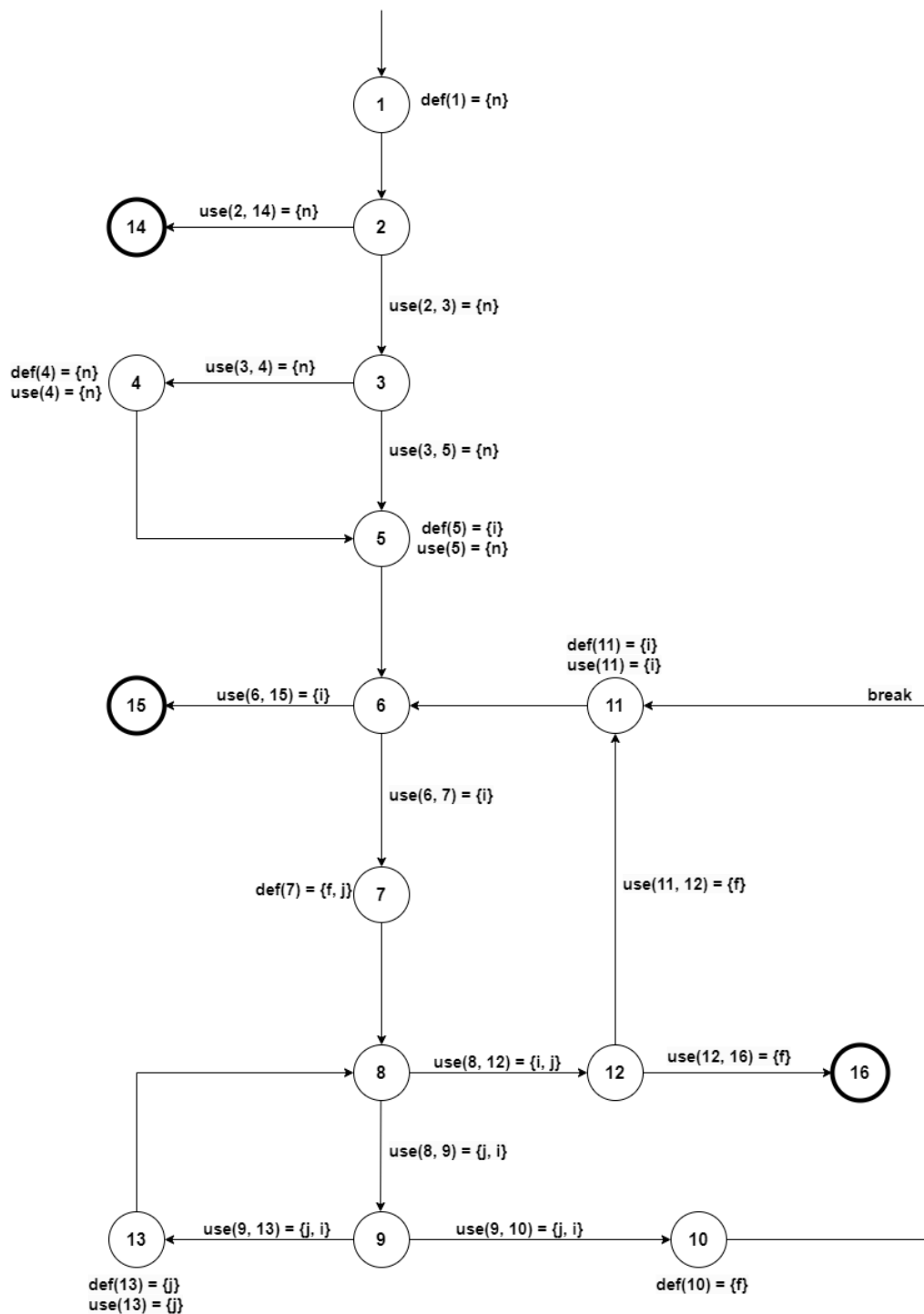
        if(f)
            return i;
    }

    return 1;
}
```

Граф за тек на контрола на програмата (Control Flow Graph)



Графот со обележани дефиниции (def) и употреби (use)



Structural coverage

Node coverage: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 и 16.

Edge coverage: (1,2), (2, 3), (2, 14), (3, 4), (3, 5), (4, 5), (5, 6), (6, 7), (6, 15), (7, 8), (8, 9), (8, 12), (9, 10), (9, 13), (10, 11), (11, 6), (12, 11), (12, 16), (13, 8).

Табела со сите дефиниции и употреби на секој јазел посебно

Node	Definition	Usage
1	{n}	
2		
3		
4	{n}	{n}
5	{i}	{n}
6		
7	{f, j}	
8		
9		
10	{f}	
11	{i}	{i}
12		
13	{j}	
14		
15		
16		

Табела со сите дефиниции и употреби на секој раб посебно

Node	Usage
(1, 2)	
(2, 3)	{n}
(2, 14)	{n}
(3, 4)	{n}
(3, 5)	{n}
(4, 5)	
(5, 6)	
(6, 7)	{i}
(6, 15)	{i}
(7, 8)	
(8, 9)	{i, j}
(8, 12)	{i, j}
(9, 10)	{i, j}
(9, 13)	{i, j}
(10, 11)	
(11, 6)	
(12, 11)	{f}
(12, 16)	{f}
(13, 8)	

Data flow coverage

Променлива	Du-pairs
n	[1, 4] [1, 5] [4, 4] [4, 5] [1, (2, 3)] [1, (2, 14)] [1, (3, 4)] [1, (3, 5)] [4, (2, 3)] [4, (2, 14)] [4, (3, 4)] [4, (3, 5)]
i	[5, 11] [5, (6, 7)] [5, (6, 15)] [5, (8, 9)] [5, (8, 12)] [5, (9, 10)] [5, (9, 13)] [11, (6, 7)] [11, (6, 15)] [11, (8, 9)] [11, (8, 12)] [11, (9, 10)] [11, (9, 13)]
j	[7, (8, 9)] [7, (8, 12)] [7, (9, 10)] [7, (9, 13)] [13, (8, 9)] [13, (8, 12)] [13, (9, 10)] [13, (9, 13)]
f	[7, (12, 11)] [7, (12, 16)] [10, (12, 11)] [10, (12, 16)]

Променлива	Du-paths
n	[1, 2, 3, 4] [1, 2, 3, 5] [4, 5] [1, 2, 3] [1, 2, 14] [1, 2, 3, 4] [1, 2, 3, 5] [1, 2, 3, 4]
i	[5, 6, 7, 8, 12, 11] [5, 6, 7] [5, 6, 7, 8, 12, 11, 6, 15] [5, 6, 7, 8, 9] [5, 6, 7, 8, 12] [5, 6, 7, 8, 9, 10] [5, 6, 7, 8, 9, 13] [11, 6, 7] [11, 6, 7, 8, 12, 11, 6, 15) [11, 6, 7, 8, 9] [11, 6, 7, 8, 12] [11, 6, 7, 8, 9, 10] [11, 6, 7, 8, 9, 13]
j	[7, 8, 9] [7, 8, 12] [7, 8, 9, 10] [7, 8, 9, 13] [13, 8, 9] [13, 8, 12] [13, 8, 9, 10] [13, 8, 9, 13]
f	[7, 8, 12, 11] [7, 8, 12, 16] [10, 11, 6, 7, 8, 12, 11] [10, 11, 6, 7, 8, 12, 16]

За да ги добиеме финалните Du-paths, потребно е да ги филтрираме сите Du-paths од горната табела.

Па така, најпрво ги бришеме патеките кои се содржат во други патеки или оние кои се повторуваат.

Следно бараме дали некои од патеките не се definition clear, односно променливата повторно се дефинира на крајот. Такви се: за n , $[1, 2, 3, 4]$ и за j , $[7, 8, 9, 13]$ и $[13, 8, 9, 13]$.

После ова филтрирање ги добиваме следните Du-paths:

1. $[1, 2, 3, 5]$
2. $[4, 5]$
3. $[1, 2, 14]$
4. $[5, 6, 7, 8, 12, 11, 6, 15]$
5. $[5, 6, 7, 8, 9, 10]$
6. $[5, 6, 7, 8, 9, 13]$
7. $[11, 6, 7, 8, 12, 11, 6, 15]$
8. $[11, 6, 7, 8, 9, 10]$
9. $[11, 6, 7, 8, 9, 13]$
10. $[13, 8, 12]$
11. $[13, 8, 9, 10]$
12. $[10, 11, 6, 7, 8, 12, 11]$
13. $[10, 11, 6, 7, 8, 12, 16]$

На крај ги добиваме следните Test paths:

1. [1, 2, 14]
2. [1, 2, 3, 4, 5, 6, 15]
3. [1, 2, 3, 5, 6, 15]
4. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 6, 15]
5. [1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 6, 15]
6. [1, 2, 3, 4, 5, 6, 7, 8, 12, 16]
7. [1, 2, 3, 5, 6, 7, 8, 12, 16]
8. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 6, 7, 8, 12, 16]
9. [1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 6, 7, 8, 12, 16]
10. [1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 8, 12, 16]
11. [1, 2, 3, 5, 6, 7, 8, 9, 13, 8, 12, 16]
12. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 6, 7, 8, 9, 13, 8, 12, 16]
13. [1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 6, 7, 8, 9, 13, 8, 12, 16]

Тест 1: [1, 2, 14]

n = 1

Очекуван излез: throw new IllegalArgumentException ()

Тест 2: [1, 2, 3, 4, 5, 6, 15]

n = 2

Очекуван излез: return 1

Тест 3: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 6, 15]

n =

Очекуван излез:

Тест 4: [1, 2, 3, 4, 5, 6, 7, 8, 12, 16]

n = 4

Очекуван излез: i = 1

Тест 5: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 6, 7, 8, 12, 16]

n =

Очекуван излез:

Тест 6: [1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 8, 12, 16]

n =

Очекуван излез:

Тест 7: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 6, 7, 8, 9, 13, 8, 12, 16]

n =

Очекуван излез:

Изработил:
Бојан Ристов (211151)