



Noakhali Science and Technology University

Institute of Information Technology

Course title: Software matrices Lab

Course Code:3204

Submitted To

Dipak Chandra Das
Assistant Professor
IIT,NSTU

Submitted By

MD. Raju Biswas
Sanzida Sultana
Abdullah Al Mamun

Submission Date:05-03-2023

Table of Contents

Task-3. Code Structure Measurement of SPL I	3
1. DD Graphs of Search	4
2. Here DD Graph of GrepContent.....	5
3. Here DD Graph of MethodFind	6
4. Here DD Graph of processSearchfile.....	7
5. Here DD Graph of processSearchfile.....	8
6. Here DD Graph of CompressFilewriteHelper.....	9
7. Here DD Graph of Compress.....	10
8. Here DD Graph of decompressfilewritehelper.....	11
9. Here DD Graph of FilecountReader.....	12
10. Here DD Graph of FilewriteHelper.....	13
11. Here DD Graph of FrequencyMapCreate.....	14
12. Here DD Graph of Huffman.....	15
13. Here DD Graph of HuffmanTuple.....	17
14. Here DD Graph of Node.....	18
15. Here DD Graph of Code clone.....	19
16. Here DD Graph of cosine similarity.....	20
17. Here DD Graph of getidfield	21
18. Here DD Graph of preprosessing	22
19. Here DD Graph of tfidfcalculate.....	23

Task-3. Code Structure Measurement of SPL I

The size of a development product reveals a great deal about the effort that went into its creation. A software module or design can be seen from several angles. The structure of the product influences not just the amount of development work required, but also how the product is maintained. As a result, we must analyse product structural features and identify how they impact the outcomes we desire. A flowgraph is a directed graph with two special nodes, the start node and the stop node: the stop node has out-degree zero, and every node falls on some route from the start node to the stop node. We differentiate the start and stop nodes in flowgraphs by surrounding them. Flowgraph nodes with zero degrees Procedure nodes are those with a value of 1; all other nodes (save the stop node) are known as predicate nodes.

Our major focus is now on DD-graphs and their intricacy. We also attempt to demonstrate how the DD graph works. We have SPL -II project and based on this project our work is shown below.

Project SPL-II

Project name: **Coding Helper**

Contributor: Md.Raju Biswas, Abdullah Al Mamun, Sanzida Sultana.

Formulae Based on Cyclomatic Complexity

Based on the cyclomatic complexity, the following formulae are being summarized.

1. $V(G) = e - n + 2p$

where e is number of edges, n is the number of nodes in the graph, and p is number of components in the whole graph.

2. $V(G) = d + p$; where d is the number of decision nodes in the graph.

3. $V(G) = \text{number of regions in the graph}$

Here we follow the **rule-1** to calculate the cyclomatic complexity of DD graph.

1.DD Graphs of Search

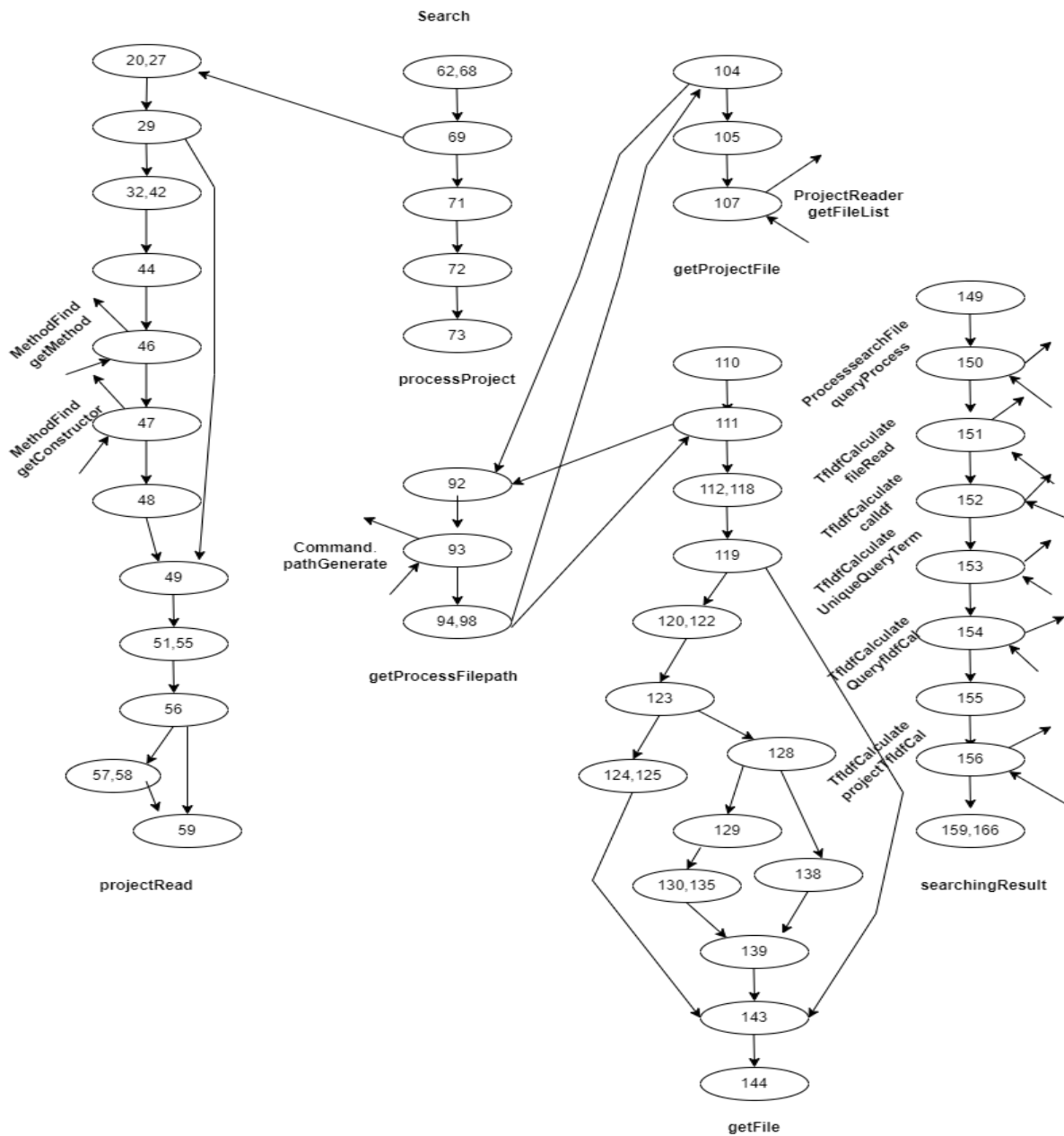


Figure -1: DD Graph of Search

<p>Now for project read</p> $V(G) = e - n + 2p$ $= 13 - 12 + 2 \times 1$ $= 3$	<p>Now for process project</p> $V(G) = e - n + 2p$ $= 4 - 5 + 2 \times 1$ $= 1$	<p>Now for getprojectfile</p> $V(G) = e - n + 2p$ $= 2 - 3 + 2 \times 1$ $= 1$
<p>For getprojectfile path</p> $V(G) = e - n + 2p$ $= 2 - 3 + 2 \times 1$ $= 1$	<p>Now for getfile</p> $V(G) = e - n + 2p$ $= 16 - 14 + 2 \times 1$ $= 0$	<p>Now for searschresult</p> $V(G) = e - n + 2p$ $= 8 - 9 + 2 \times 1$ $= 1$

Now the total complexity

$$\begin{aligned} V(G) &= e - n + 2p \\ &= 45 - 46 + 2 * 6 \\ &= 11 \end{aligned}$$

2. Here DD Graph of GrepContent.

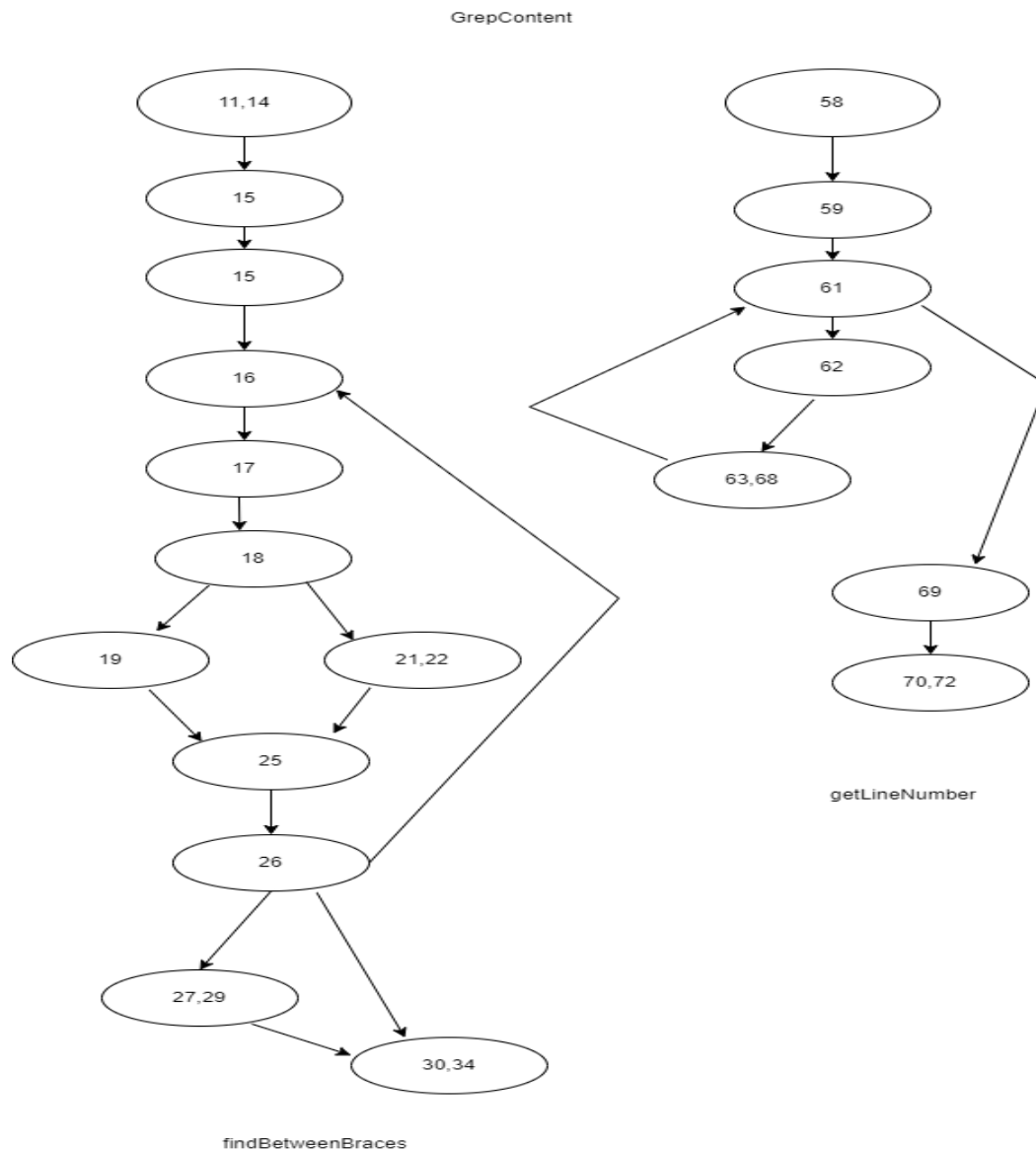


Figure -2: DD Graph of GrepContent

<p>Now for Find between braces</p> $V(G) = e - n + 2p$ $= 14 - 12 + 2 * 1$ $= 0$	<p>Now for getlinenumber</p> $V(G) = e - n + 2p$ $= 7 - 7 + 2 * 1$ $= 2$
--	--

Now the total complexity will different because above two graphs are not connected.

3. Here DD Graph of MethodFind .

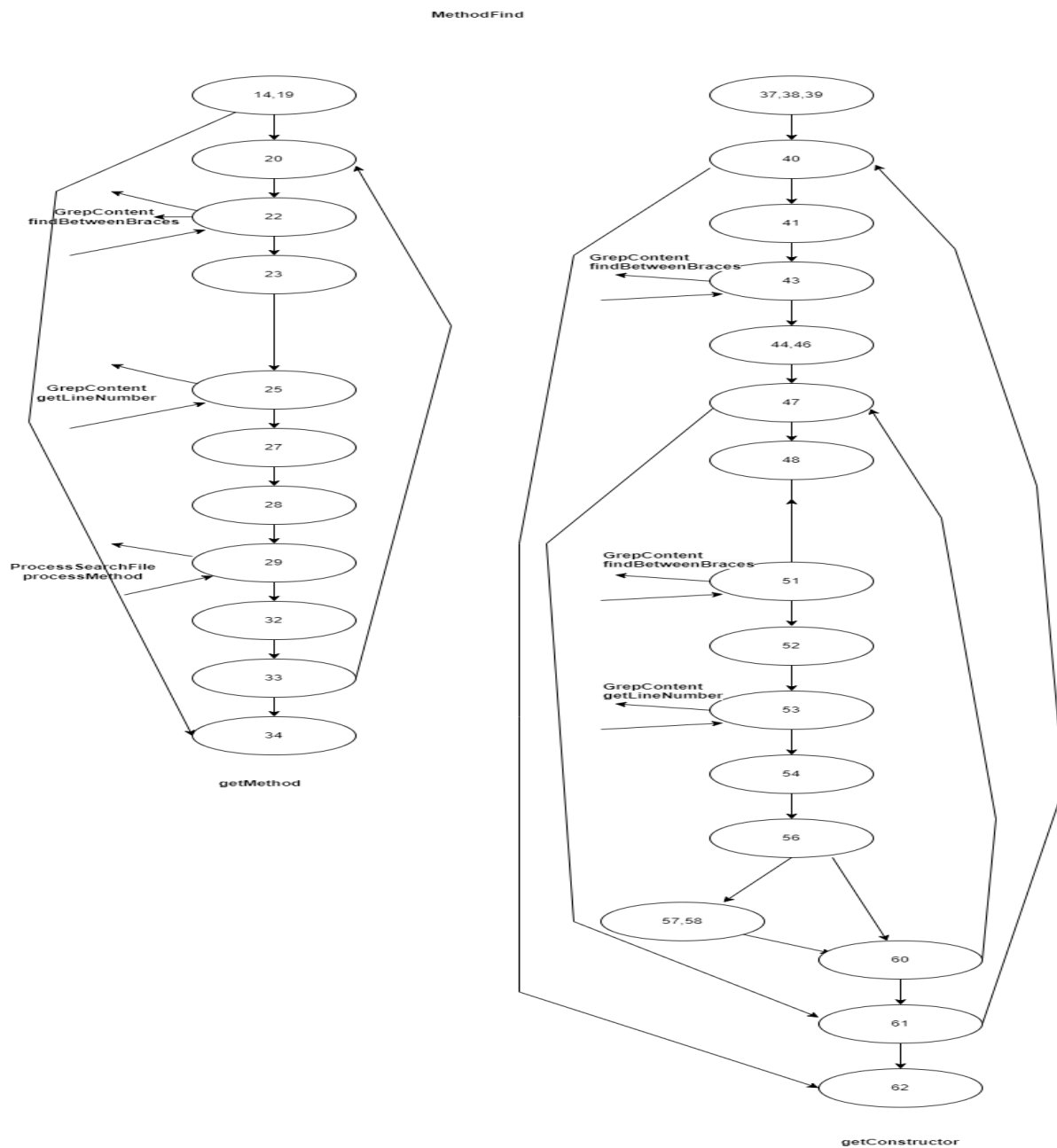


Figure-3:DD graph of method find.

<p>Now for getmethod</p> $V(G) = e - n + 2p$ $= 12 - 11 + 2 * 1$ $= 3$	<p>Now for getconstructor</p> $V(G) = e - n + 2p$ $= 20 - 16 + 2 * 1$ $= 6$
--	---

Now the total complexity will different because above two graphs are not connected.

4. Here DD Graph of processSearchfile.

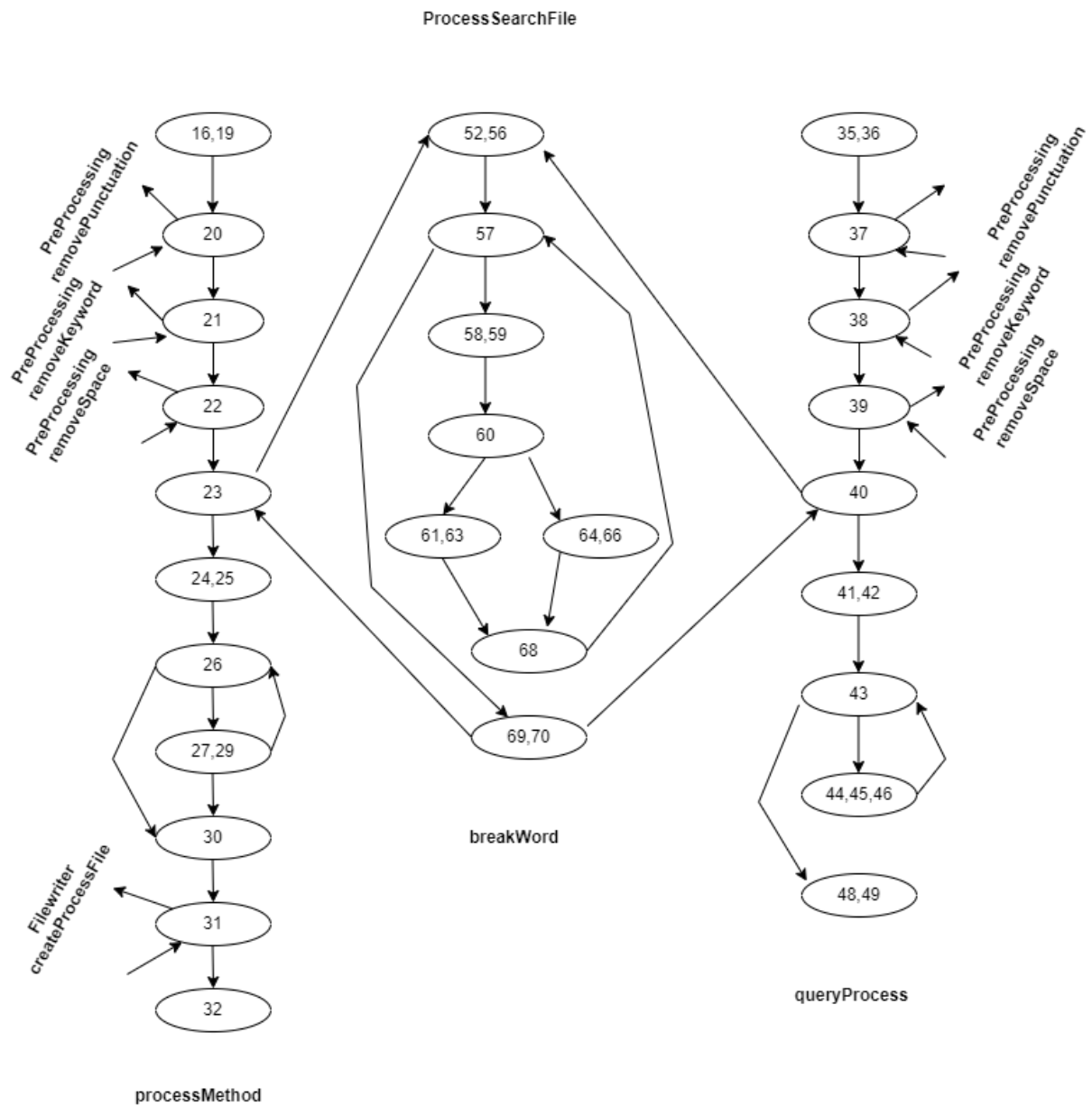


Figure-4:DD graphs of processSearchfile.

<p>Now for processmethod</p> $V(G) = e - n + 2p$ $= 12 - 11 + 2 * 1$ $= 3$	<p>Now for breakword</p> $V(G) = e - n + 2p$ $= 9 - 8 + 2 * 1$ $= 3$	<p>Now for queryprocess</p> $V(G) = e - n + 2p$ $= 9 - 9 + 2 * 1$ $= 2$
--	--	---

Now the total complexity

$$V(G) = e - n + 2p$$

$$= 30 - 28 + 2 * 3$$

$$= 8$$

5. Here DD Graph of processSearchfile.

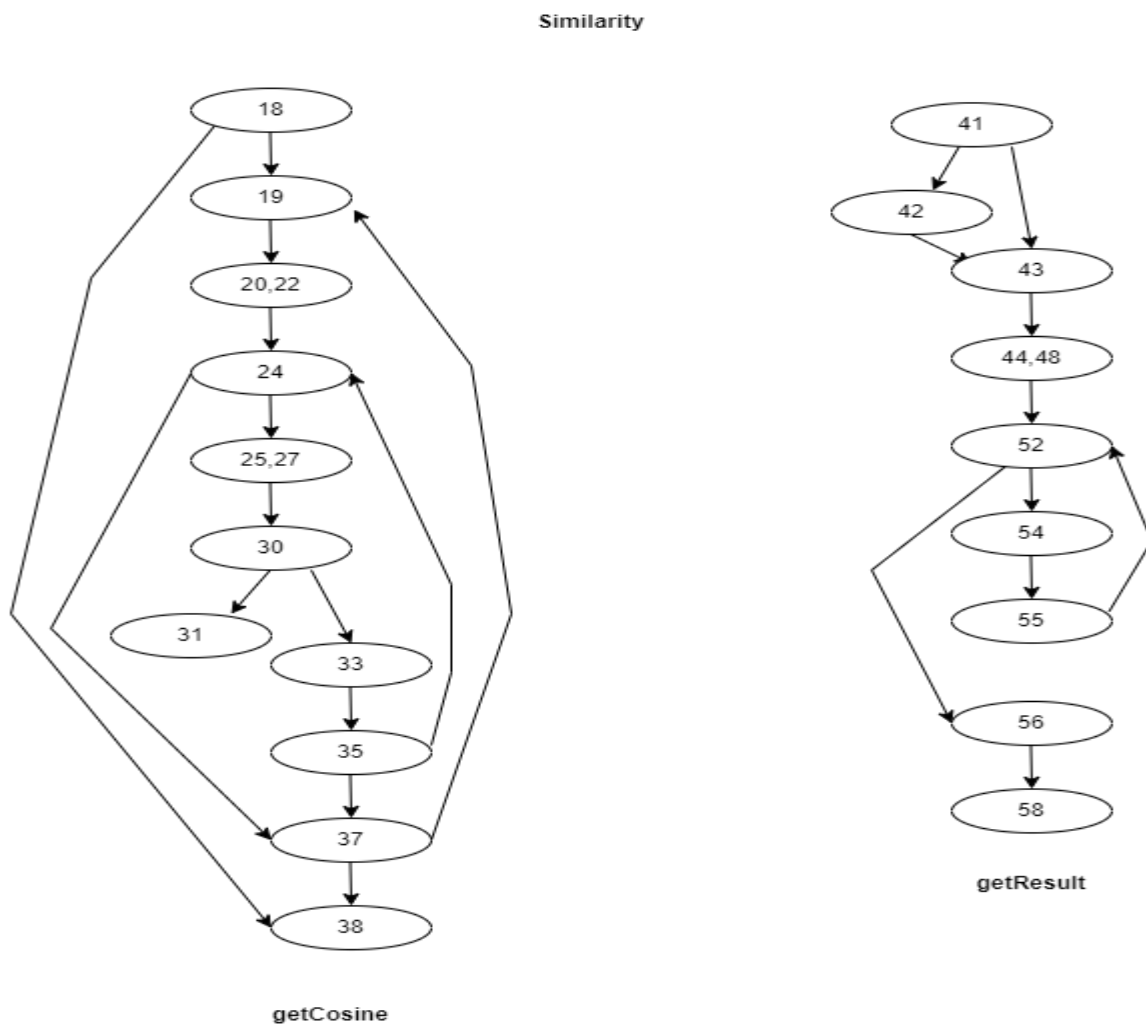


Figure-5:DD graphs of similarity.

<p>Now for getcosine</p> $V(G) = e - n + 2p$ $= 14 - 11 + 2 * 1$ $= 5$	<p>Now for getresults</p> $V(G) = e - n + 2p$ $= 10 - 9 + 2 * 1$ $= 3$
--	--

Now the total complexity will different because above two graphs are not connected.

6. Here DD Graph of CompressFilewriteHelper.

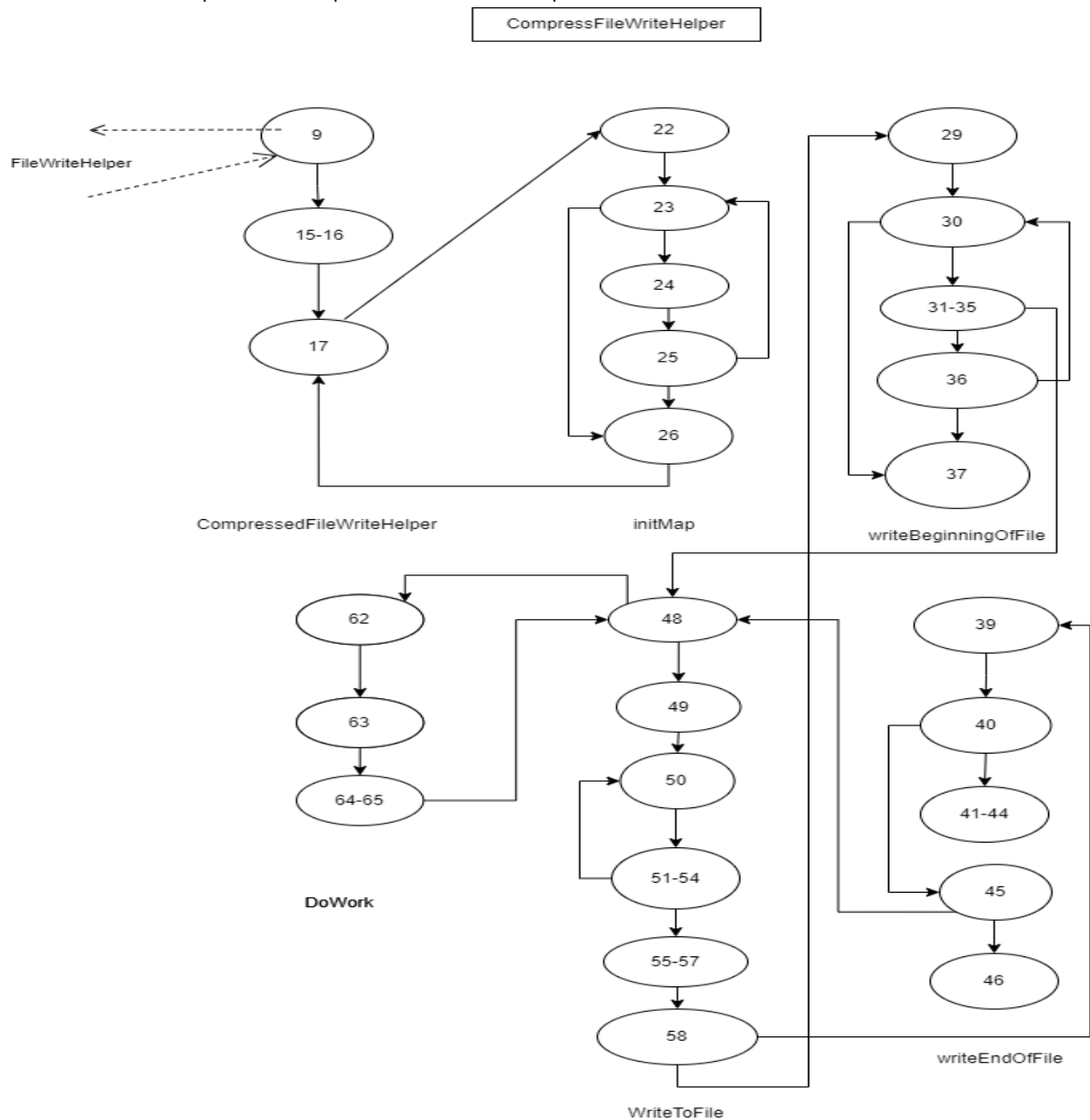


Figure-6:DD graphs of CompressFilewriteHelper

<p>Now for CompressFilewriteHelper</p> $V(G) = e - n + 2p$ $= 2 - 3 + 2 * 1$ $= 1$	<p>Now for intmap</p> $V(G) = e - n + 2p$ $= 6 - 5 + 2 * 1$ $= 3$	<p>Now for writebeginingfofile</p> $V(G) = e - n + 2p$ $= 6 - 5 + 2 * 1$ $= 3$
<p>For dowork</p> $V(G) = e - n + 2p$ $= 2 - 3 + 2 * 1$ $= 1$	<p>Now for getfile</p> $V(G) = e - n + 2p$ $= 6 - 6 + 2 * 1$ $= 2$	<p>Now for searschresult</p> $V(G) = e - n + 2p$ $= 4 - 5 + 2 * 1$ $= 1$

Now here two type connected graph and there complexity will be difference. And those are

$$V1(G) = e - n + 2p$$

$$= 8 - 8 + 2 * 2$$

$$= 4$$

$$V2(G) = e - n + 2p$$

$$= 18 - 19 + 2 * 4$$

$$= 7$$

7. Here DD Graph of Compress.

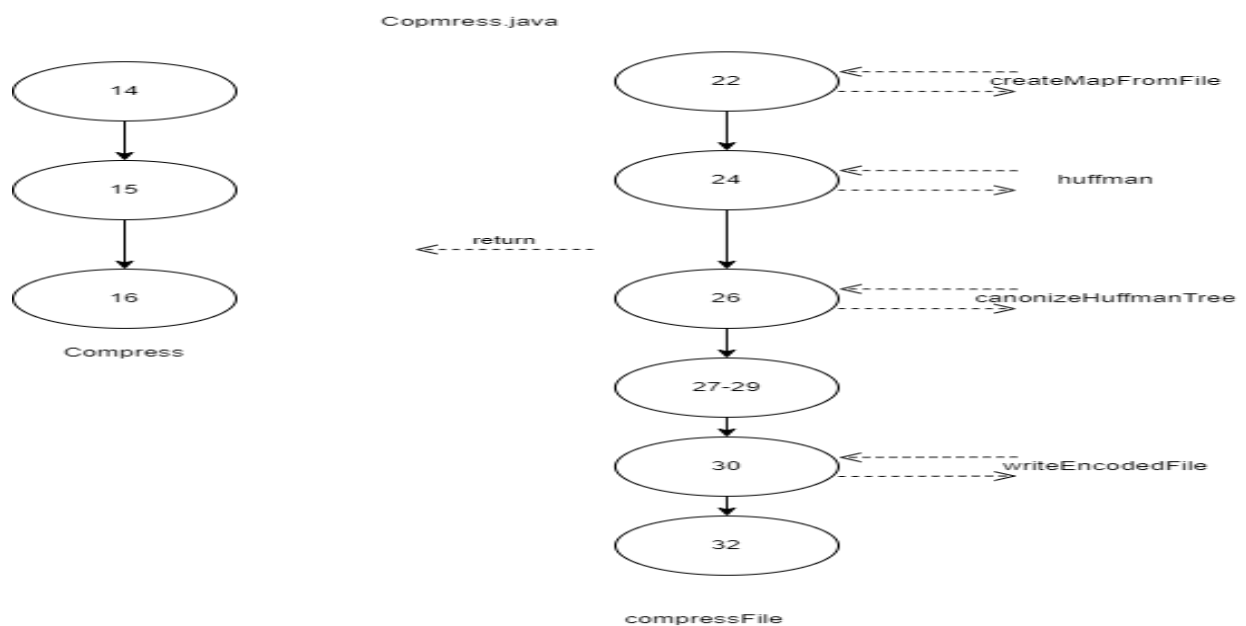


Figure-7:DD graphs of Compress.

<p>Now for compress</p> $V(G) = e - n + 2p$ $= 2 - 3 + 2 * 1$ $= 1$	<p>Now for compressfile</p> $V(G) = e - n + 2p$ $= 5 - 6 + 2 * 1$ $= 1$
---	---

Now the total complexity will different because above two graphs are not connected.

8. Here DD Graph of decompressfilewritehelper.

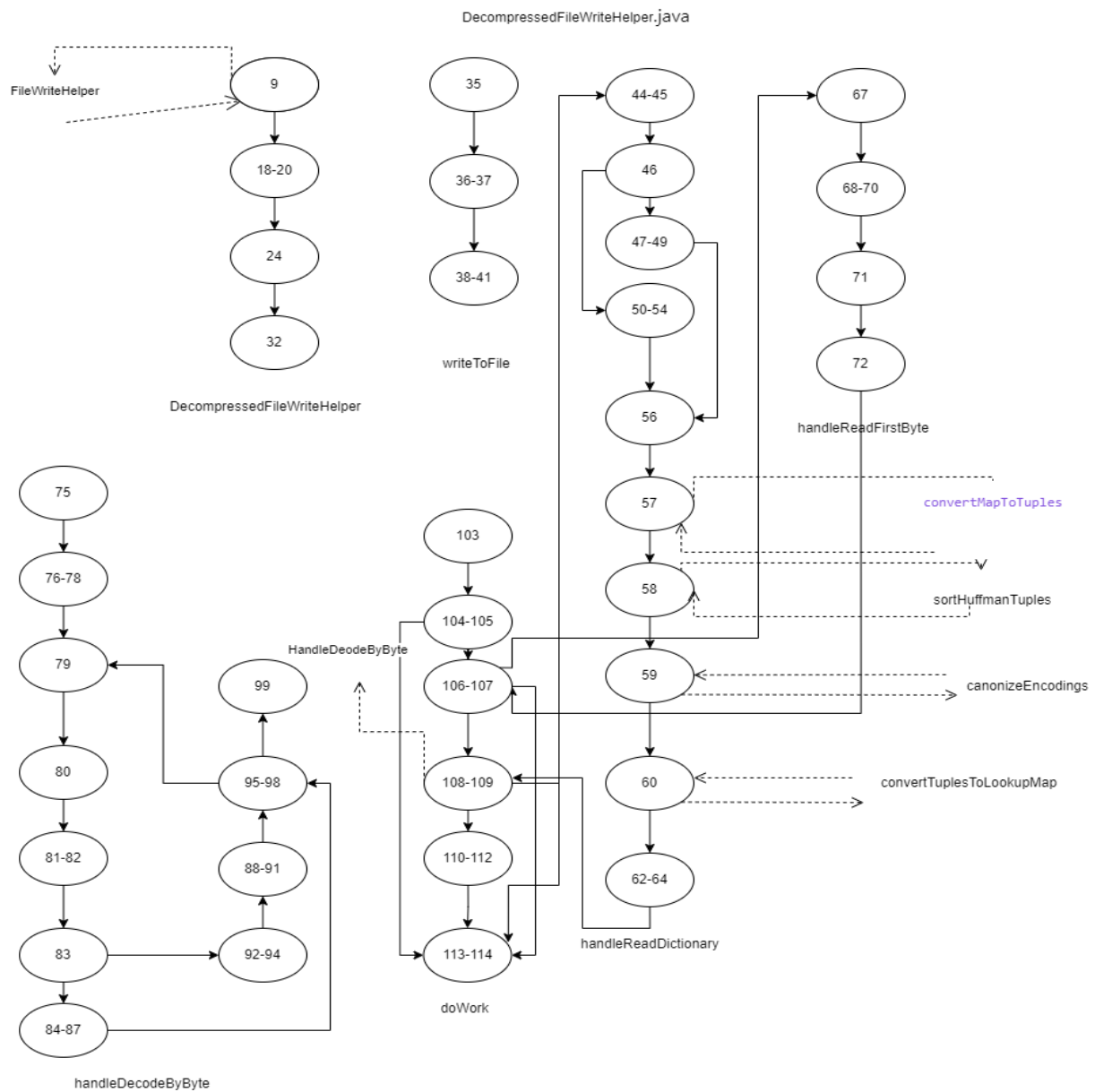


Figure-8:DD graph of decompressfilewritehelper.

decompressfilewritehelpe $V(G) = e - n + 2p$ $= 4 - 3 + 2 * 1$ $= 3$	Now for writetofile $V(G) = e - n + 2p$ $= 3 - 2 + 2 * 1$ $= 3$	handlewritedictionary $V(G) = e - n + 2p$ $= 10 - 10 + 2 * 1$ $= 2$
For handledecodebyte $V(G) = e - n + 2p$ $= 12 - 11 + 2 * 1$ $= 3$	Now for dowork $V(G) = e - n + 2p$ $= 7 - 6 + 2 * 1$ $= 3$	handlereadfirstbyte $V(G) = e - n + 2p$ $= 3 - 4 + 2 * 1$ $= 1$

This diagram there are 3 structure that are connected and 3 solo. Now solo have their own complexity but those that are connected have total complexity. And that is

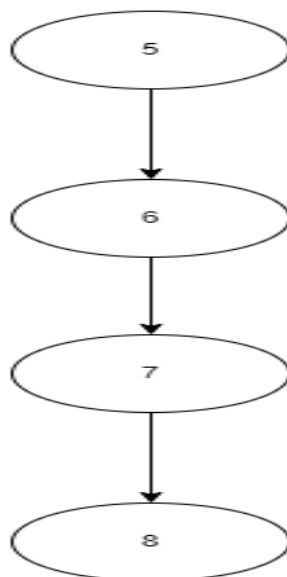
$$V(G) = e - n + 2p$$

$$= 20 - 20 + 2 * 3$$

$$= 6$$

9. Here DD Graph of FilecountReader.

FileContentReader.java



FileContentReader

Figure-9: DD graphs of FilecountReader

FilecountReader

$$V(G) = e - n + 2p$$

$$= 3 - 4 + 2 * 1$$

$$= 1$$

10. Here DD Graph of FilewriteHelper.

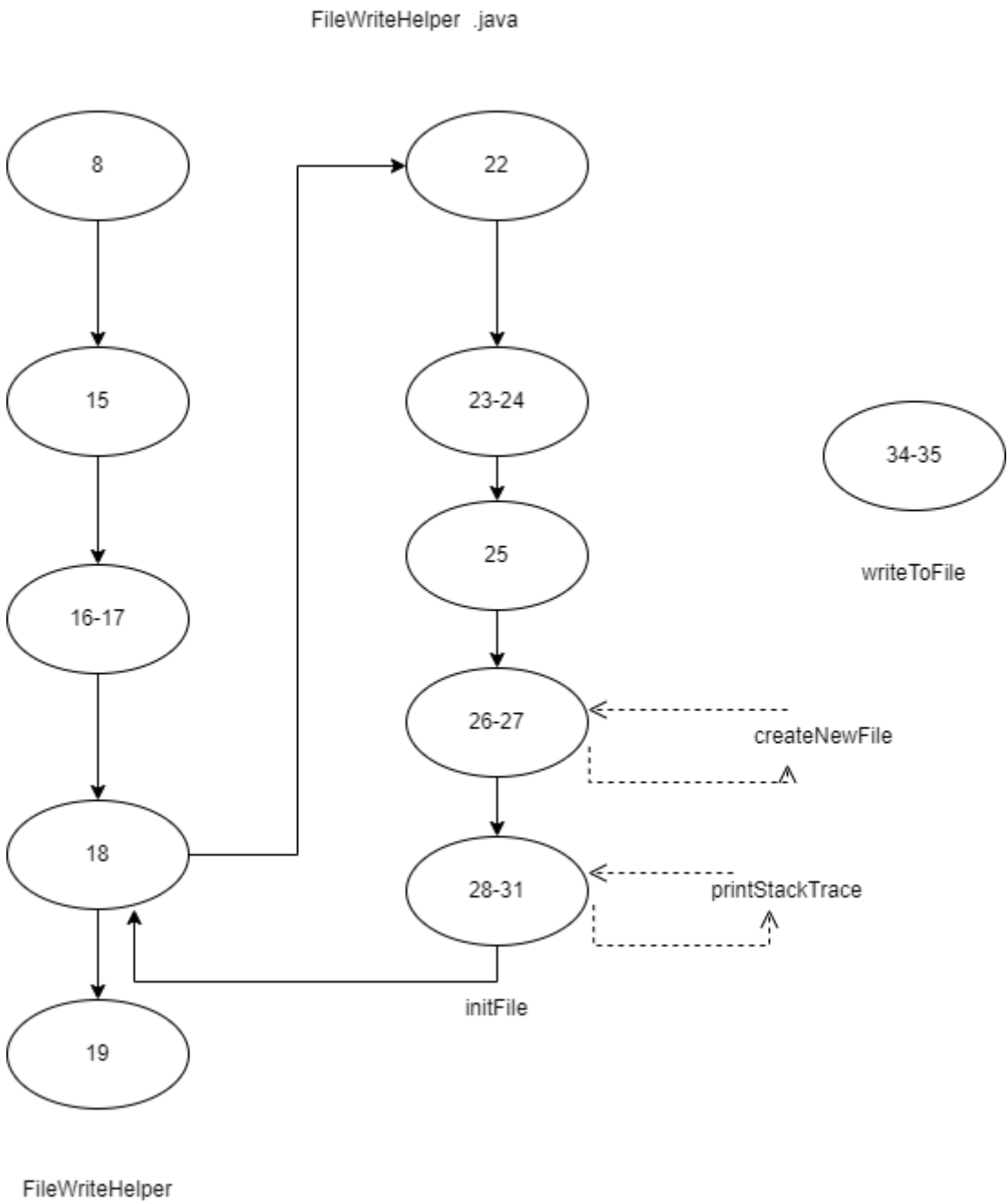


Figure-10:DD graphs of FilewriteHelper.

FilewriteHelper $V(G) = e - n + 2p$ $= 4 - 5 + 2 * 1$ $= 1$	intfile $V(G) = e - n + 2p$ $= 4 - 5 + 2 * 1$ $= 1$	Now for writetofile $V(G) = e - n + 2p$ $= 0 - 1 + 2 * 1$ $= 1$
--	--	--

Here two methods are connected and one is solo. Now total complexity will be divided into two parts. And those are.

$$\begin{aligned}
 V(G) &= e - n + 2p \\
 &= 8 - 10 + 2 * 2 \\
 &= 2
 \end{aligned}$$

11. Here DD Graph of FrequencyMapCreate.

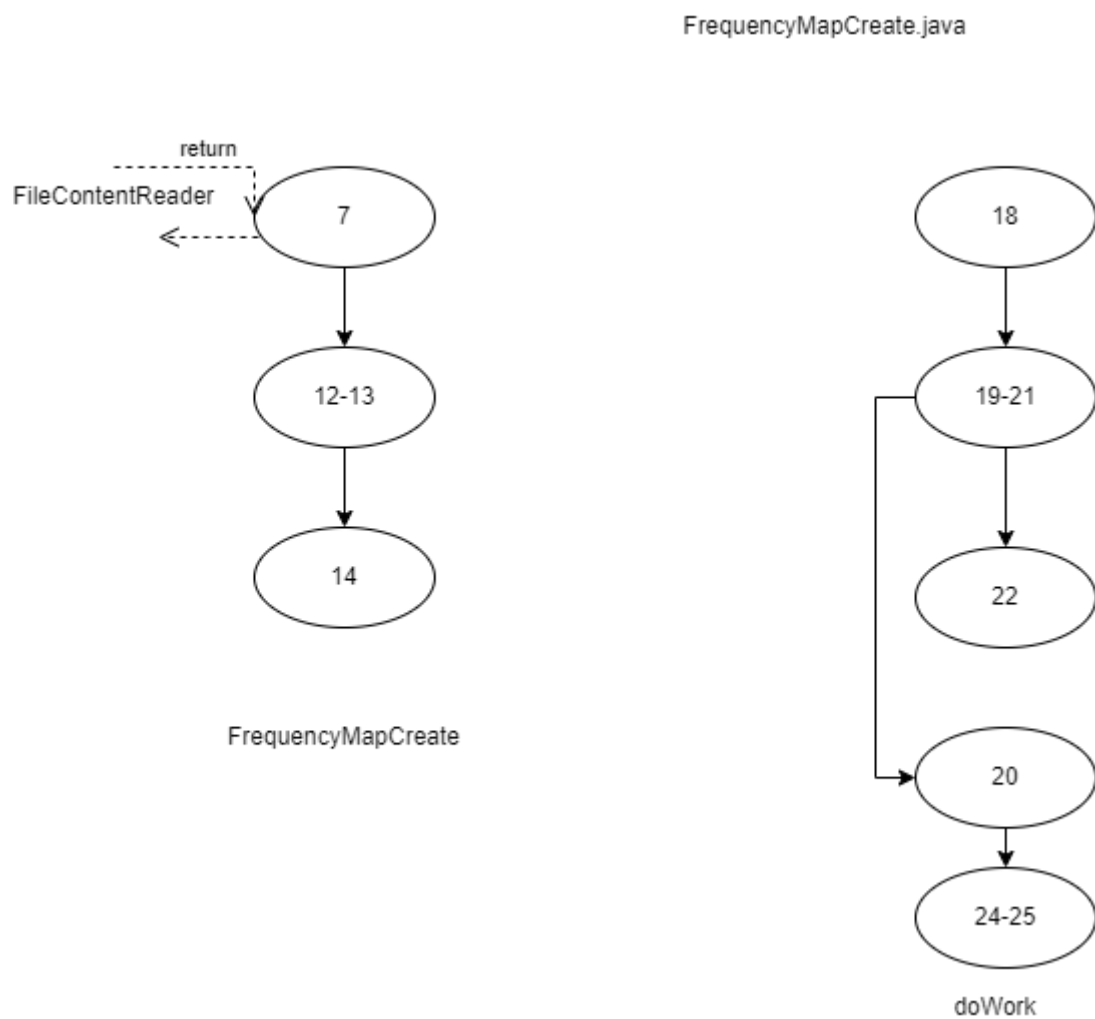


Figure-11: DD Graph of FrequencyMapCreate.

<p>FrequencyMapCreate</p> $V(G) = e - n + 2p$ $= 2 - 3 + 2 * 1$ $= 1$	<p>dowork</p> $V(G) = e - n + 2p$ $= 4 - 5 + 2 * 1$ $= 1$
---	---

Now the total complexity will different because above two graphs are not connected

12. Here DD Graph of Huffman.

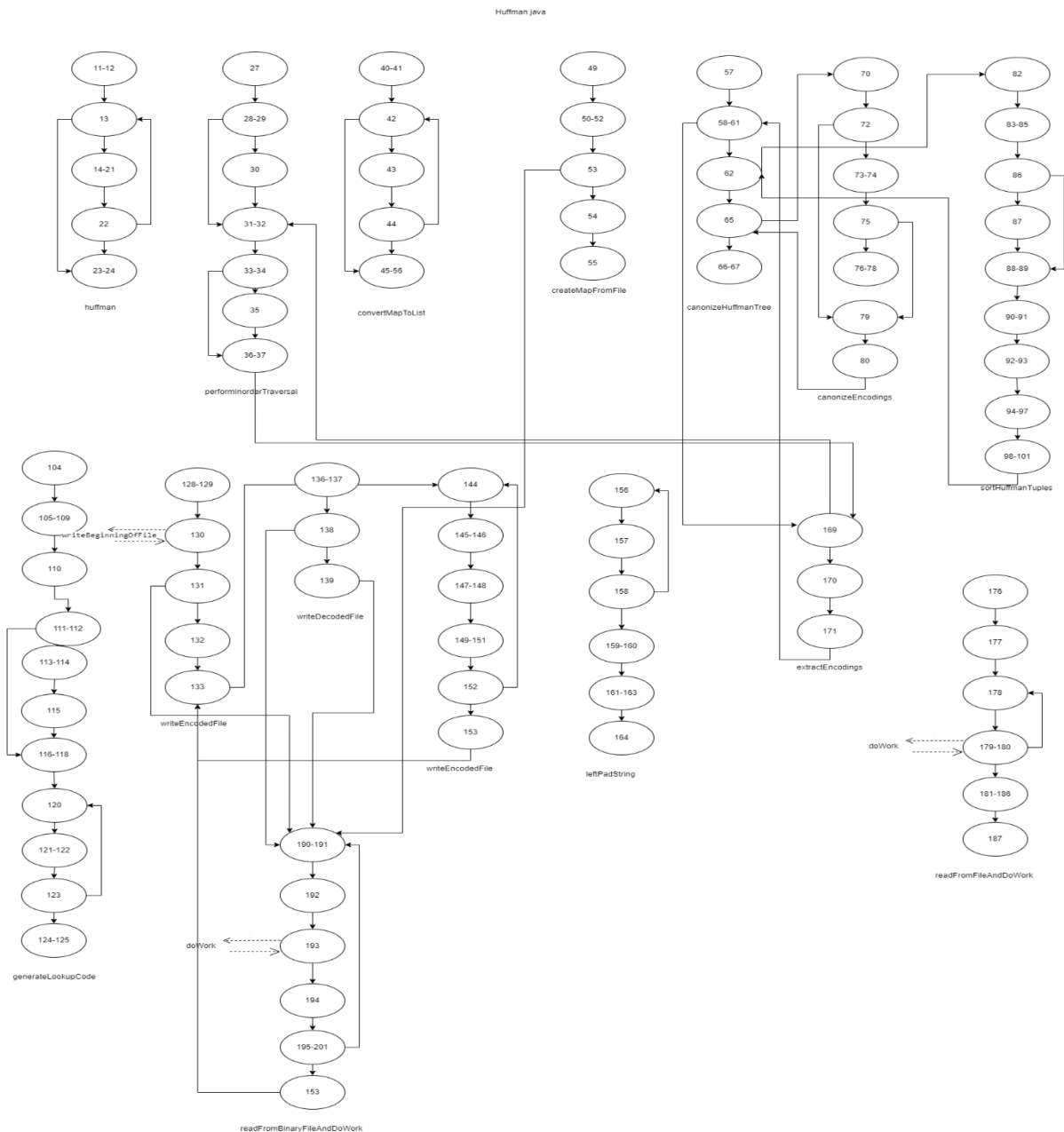


Figure-12:DD graphs of Huffman.

Huffman $V(G) = e - n + 2p$ $=6-5+2*1$ $=3$	Now for performanceTravels $V(G) = e - n + 2p$ $=8-7+2*1$ $=1$	convertMaptolist $V(G) = e - n + 2p$ $=6-5+2*1$ $=3$
For CreateMapFromFile $V(G) = e - n + 2p$ $=4-5+2*1$ $=1$	Now for canonizehuffman $V(G) = e - n + 2p$ $=4-5+2*1$ $=1$	canonizeEncoded $V(G) = e - n + 2p$ $=7-7+2*1$ $=2$

For sorthuffmanTree $V(G) = e - n + 2p$ $=9-9+2*1$ $=2$	generalizelookofcode $V(G) = e - n + 2p$ $=12-11+2*1$ $=3$	writencodedfile $V(G) = e - n + 2p$ $=4-5+2*1$ $=1$
readfrombinaryanddowork $V(G) = e - n + 2p$ $=6-6+2*1$ $=2$	writedecodefile $V(G) = e - n + 2p$ $=2-3+2*1$ $=1$	leftpadString $V(G) = e - n + 2p$ $=6-6+2*1$ $=2$

For extractfileencode $V(G) = e - n + 2p$ $=2-3+2*1$ $=1$	readfromfileanddowork $V(G) = e - n + 2p$ $=6-6+2*1$ $=2$	readfilefrombinaryfile $V(G) = e - n + 2p$ $=6-6+2*1$ $=2$
--	--	---

Here some node are connected so total complexity can be calculated and those are

$$V(G) = e - n + 2p = 52 - 51 + 2 * 10 = 21$$

13. Here DD Graph of HuffmanTuple.

HuffmanTuple.java

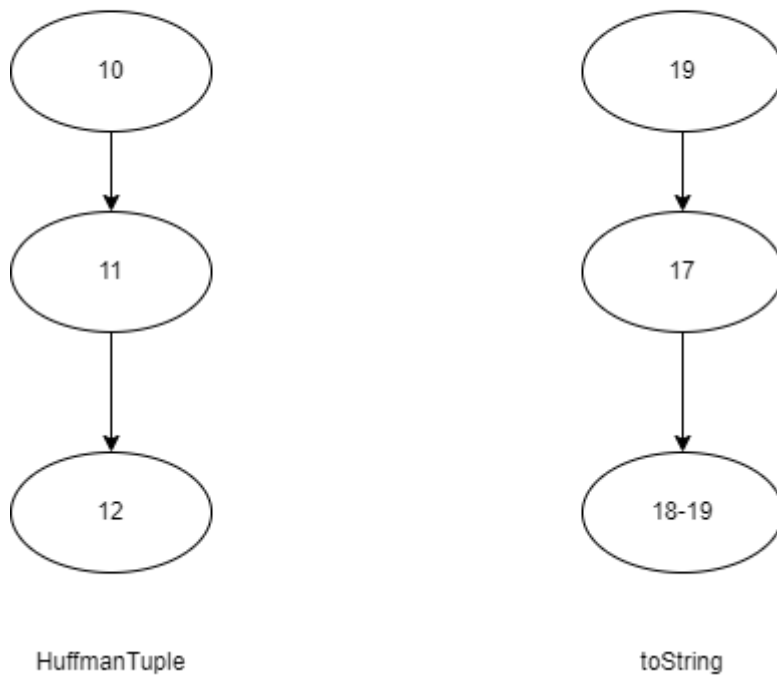


Figure-13: DD Graph of HuffmanTuple.

For HuffmanTuple	Tostring
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
$= 3 - 3 + 2 * 1$	$= 3 - 3 + 2 * 1$
$= 2$	$= 2$

No connected graph are exist so total complexity is not found. There are two complexities.

14. Here DD Graph of Node.

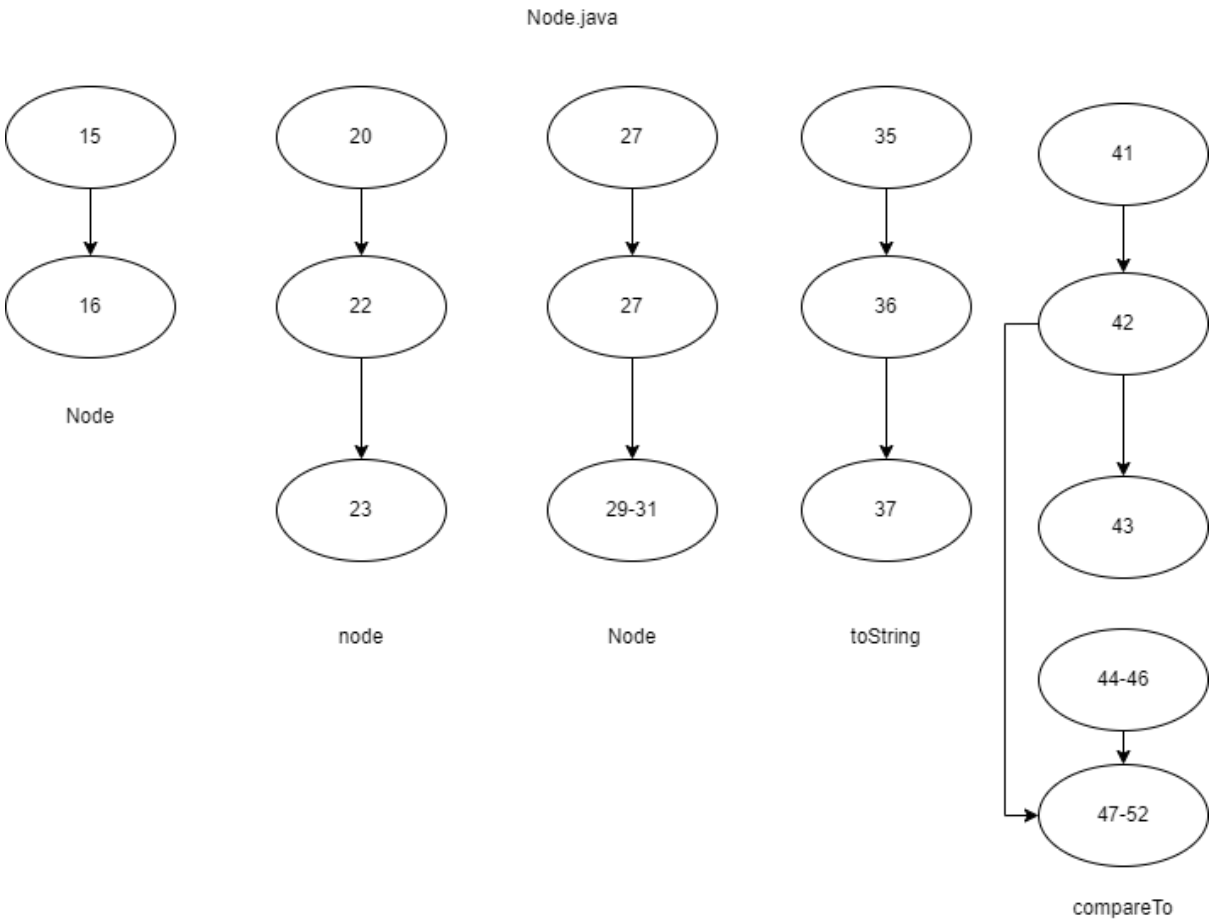


Figure-14:DD graphs for Node

<div>For node</div> <div>$V(G) = e - n + 2p$$= 1 - 2 + 2 * 1$$= 1$</div>	<div>node</div> <div>$V(G) = e - n + 2p$$= 2 - 3 + 2 * 1$$= 1$</div>	<div>node</div> <div>$V(G) = e - n + 2p$$= 2 - 3 + 2 * 1$$= 1$</div>
<div>Tostring</div> <div>$V(G) = e - n + 2p$$= 2 - 3 + 2 * 1$$= 1$</div>	<div>compareTo</div> <div>$V(G) = e - n + 2p$$= 4 - 5 + 2 * 1$$= 1$</div>	

No connected graph are exist so total complexity is not found. There are two complexities.

15. Here DD Graph of Code clone

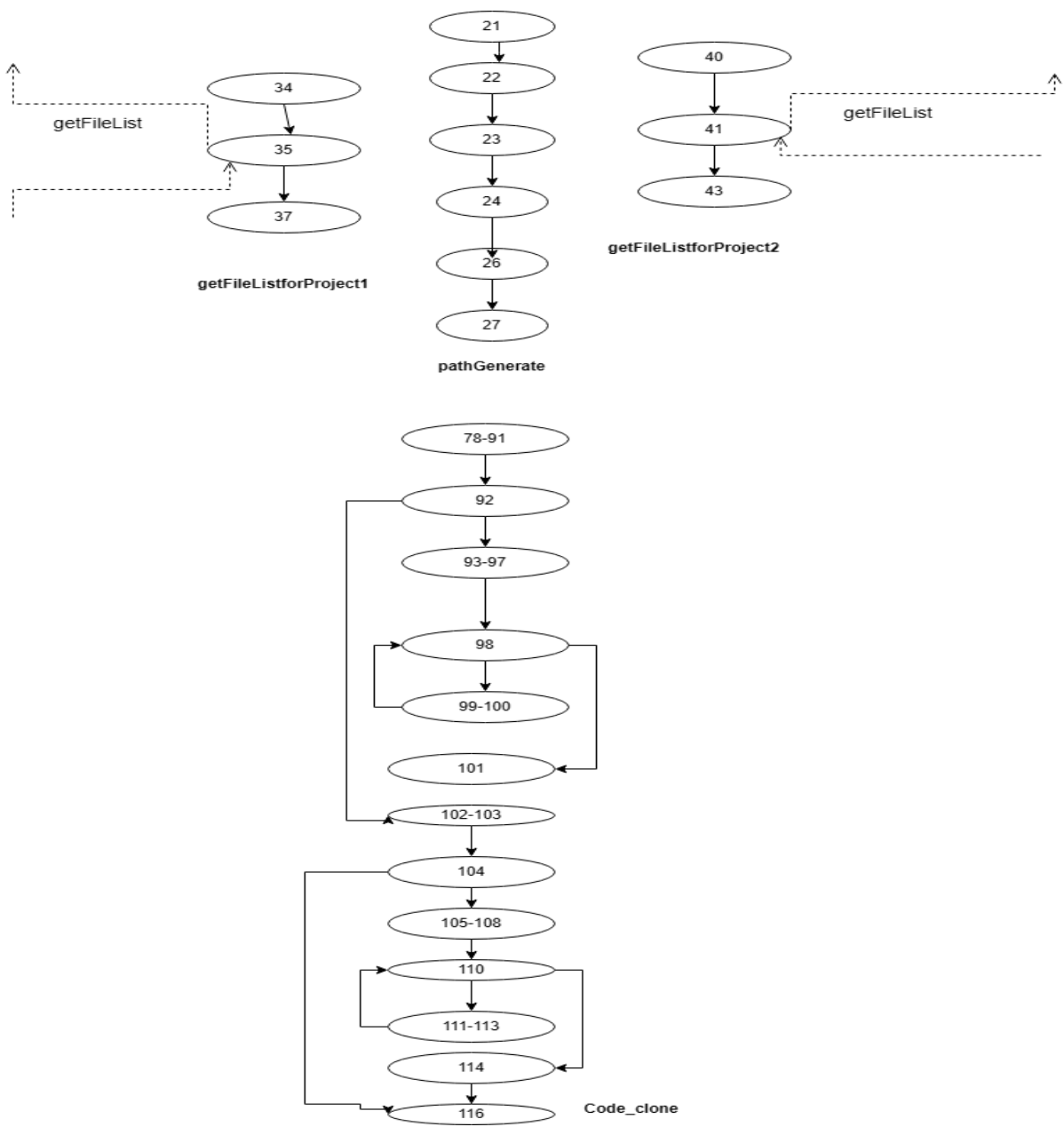


Figure-15:DD graphs Code clone

For getfilelistforproject1	pathgenerate	Getfilelistforproject2	Code_clone
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$	$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
$= 2 - 3 + 2 * 1$	$= 5 - 6 + 2 * 1$	$= 2 - 3 + 2 * 1$	$= 13 - 15 + 2 * 1$
$= 1$	$= 1$	$= 1$	$= 0$

Here no relation are found with more method so complexity will own for each module.

16. Here DD Graph of cosine similarity

CosineSimilarity

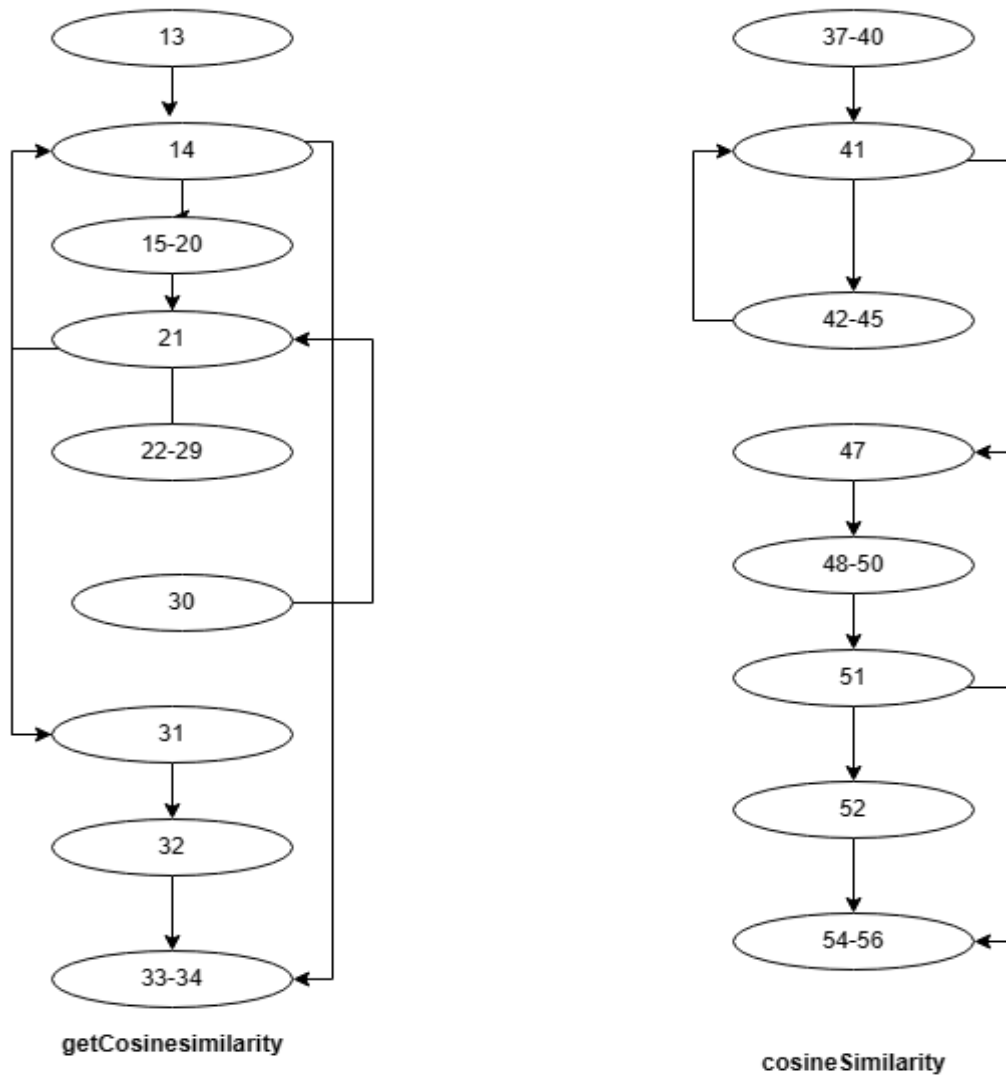


Figure-16:DD graphs for cosine similarity.

<p>For getcosinesimilarity</p> $V(G) = e - n + 2p$ $= 10 - 9 + 2 * 1$ $= 3$	<p>pathgenerate</p> $V(G) = e - n + 2p$ $= 9 - 8 + 2 * 1$ $= 3$
---	---

Here no relation are found with more method so complexity will own for each module.

17. Here DD Graph of getidfield.

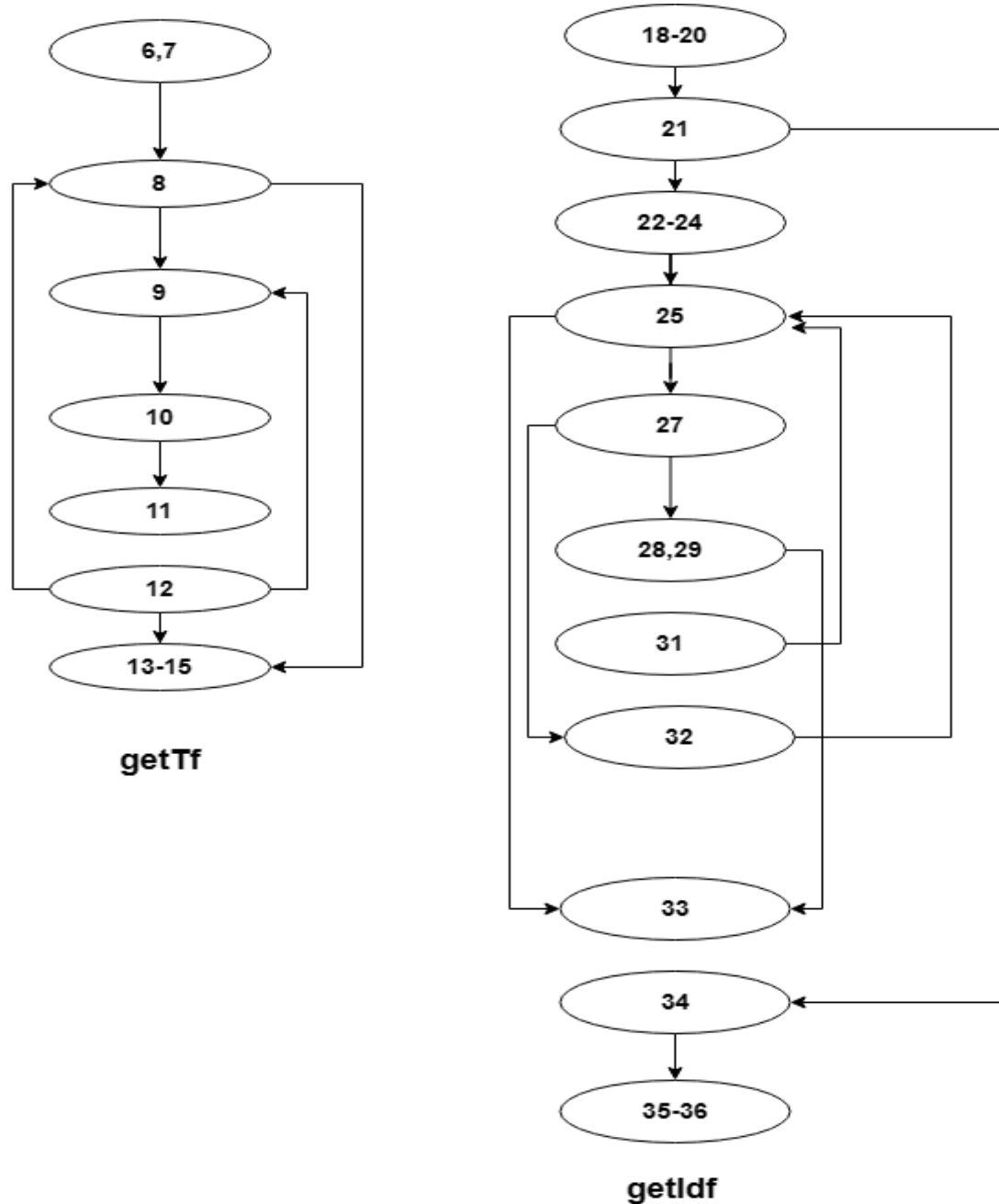


Figure-17:DD graphs for getidfield

<p>For getcosinesimilarity</p> $V(G) = e - n + 2p$ $= 8 - 7 + 2 * 1$ $= 3$	<p>pathgenerate</p> $V(G) = e - n + 2p$ $= 12 - 11 + 2 * 1$ $= 3$
--	---

Here no relation are found with more method so complexity will own for each module.

18. Here DD Graph of preprocessing.

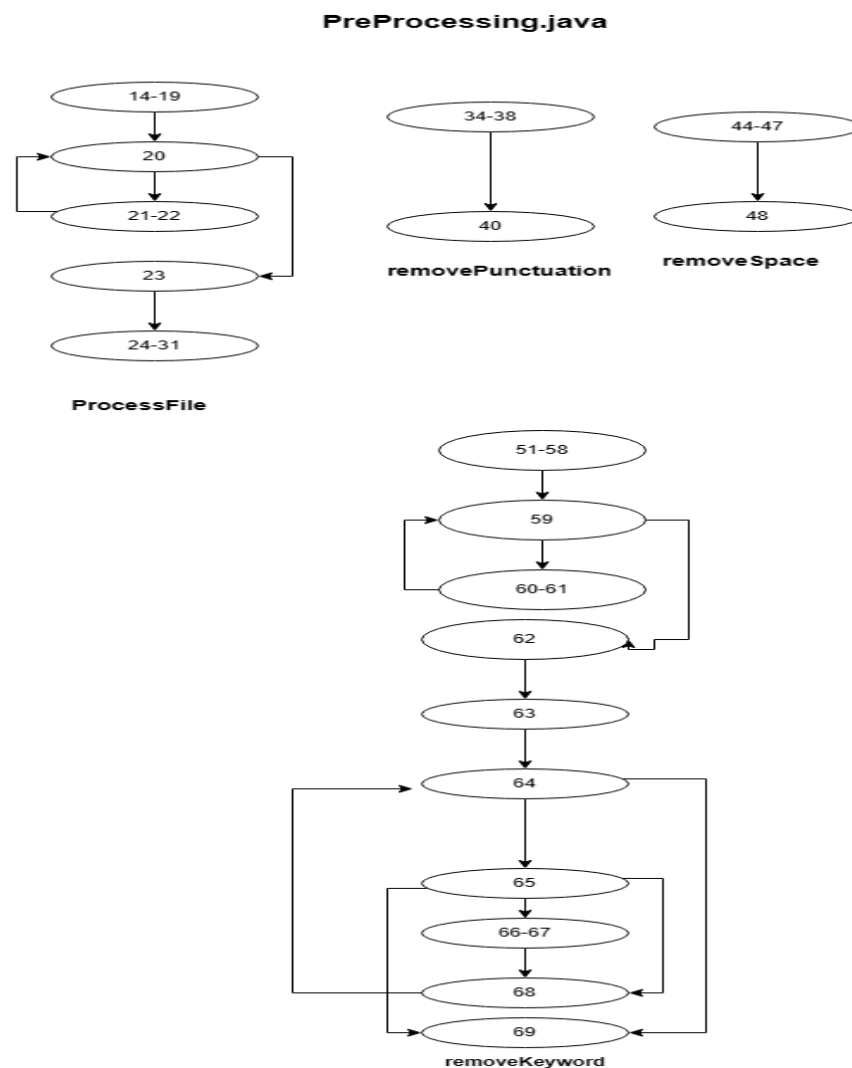


Figure-18:DD graphs for preprocessing.

<p>For processfile</p> $V(G) = e - n + 2p$ $= 5 - 5 + 2 * 1$ $= 2$	<p>removepunctuation</p> $V(G) = e - n + 2p$ $= 1 - 2 + 2 * 1$ $= 1$	<p>removespace</p> $V(G) = e - n + 2p$ $= 1 - 2 + 2 * 1$ $= 1$
<p>For removekeyword</p> $V(G) = e - n + 2p$ $= 13 - 12 + 2 * 1$ $= 3$		

Here no relation are found with more method so complexity will own for each module.

19. Here DD Graph of tfidfcalculate.

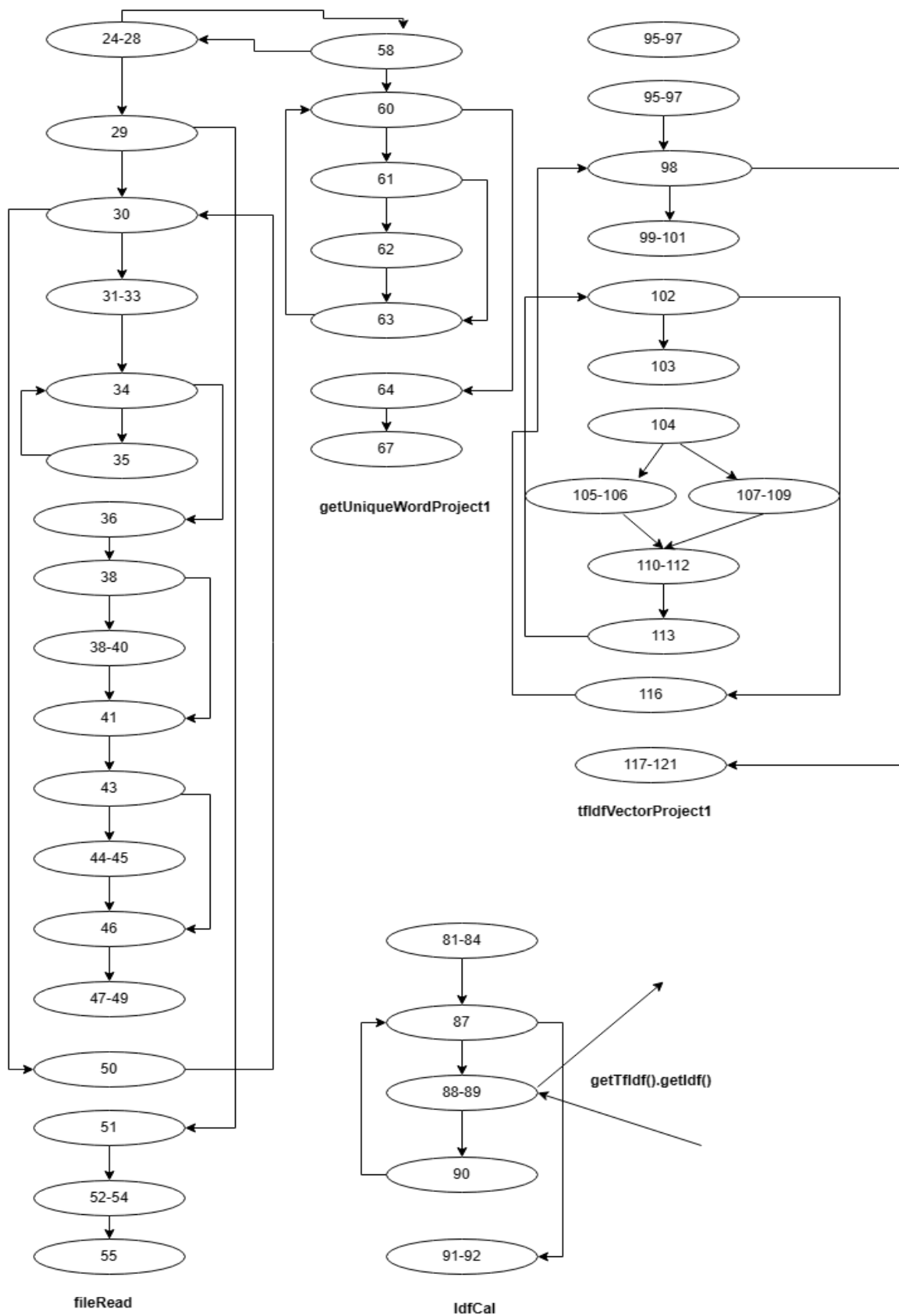


Figure-20: DD Graph of tfidfcalculate.

For fileread

$$\begin{aligned}V(G) &= e - n + 2p \\&= 21 - 18 + 2 * 1 \\&= 5\end{aligned}$$

getuniquewordprojectone

$$\begin{aligned}V(G) &= e - n + 2p \\&= 8 - 7 + 2 * 1 \\&= 3\end{aligned}$$

tfidfVectorProject1

$$\begin{aligned}V(G) &= e - n + 2p \\&= 15 - 13 + 2 * 1 \\&= 4\end{aligned}$$

For idcal

$$\begin{aligned}V(G) &= e - n + 2p \\&= 5 - 5 + 2 * 1 \\&= 2\end{aligned}$$

Here two components are connected other are solo. Now the group toatal complexity

$$\begin{aligned}V(G) &= e - n + 2p \\&= 29 - 25 + 2 * 2 \\&= 12\end{aligned}$$