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

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 out work are 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. $V(G)$ = 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

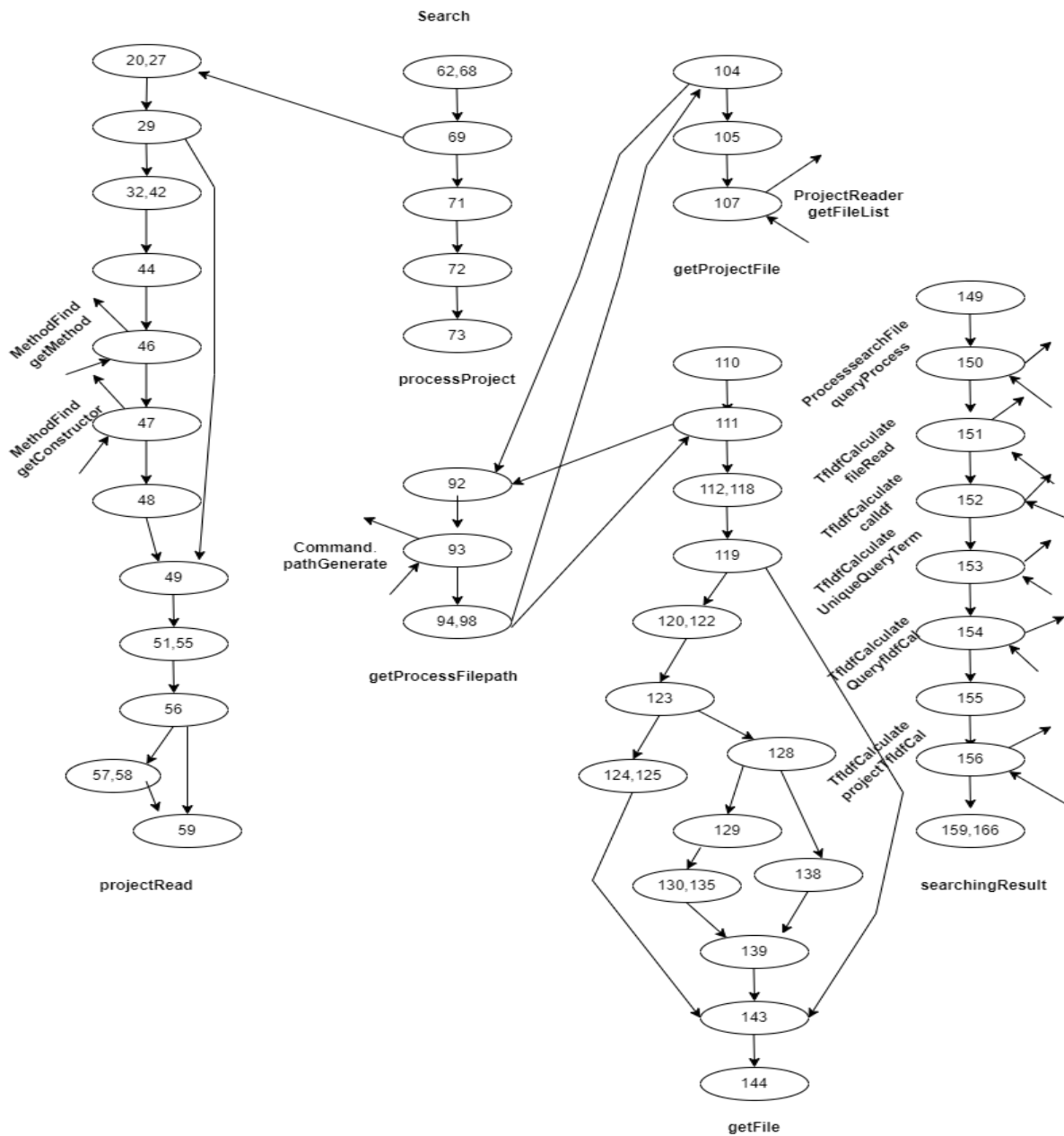Figure -1: DD Graph of Search

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

Now the total complexity

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

$\quad = 46-45+2*6$

$\quad = 13$

2. Here DD Graph of GrepContent.
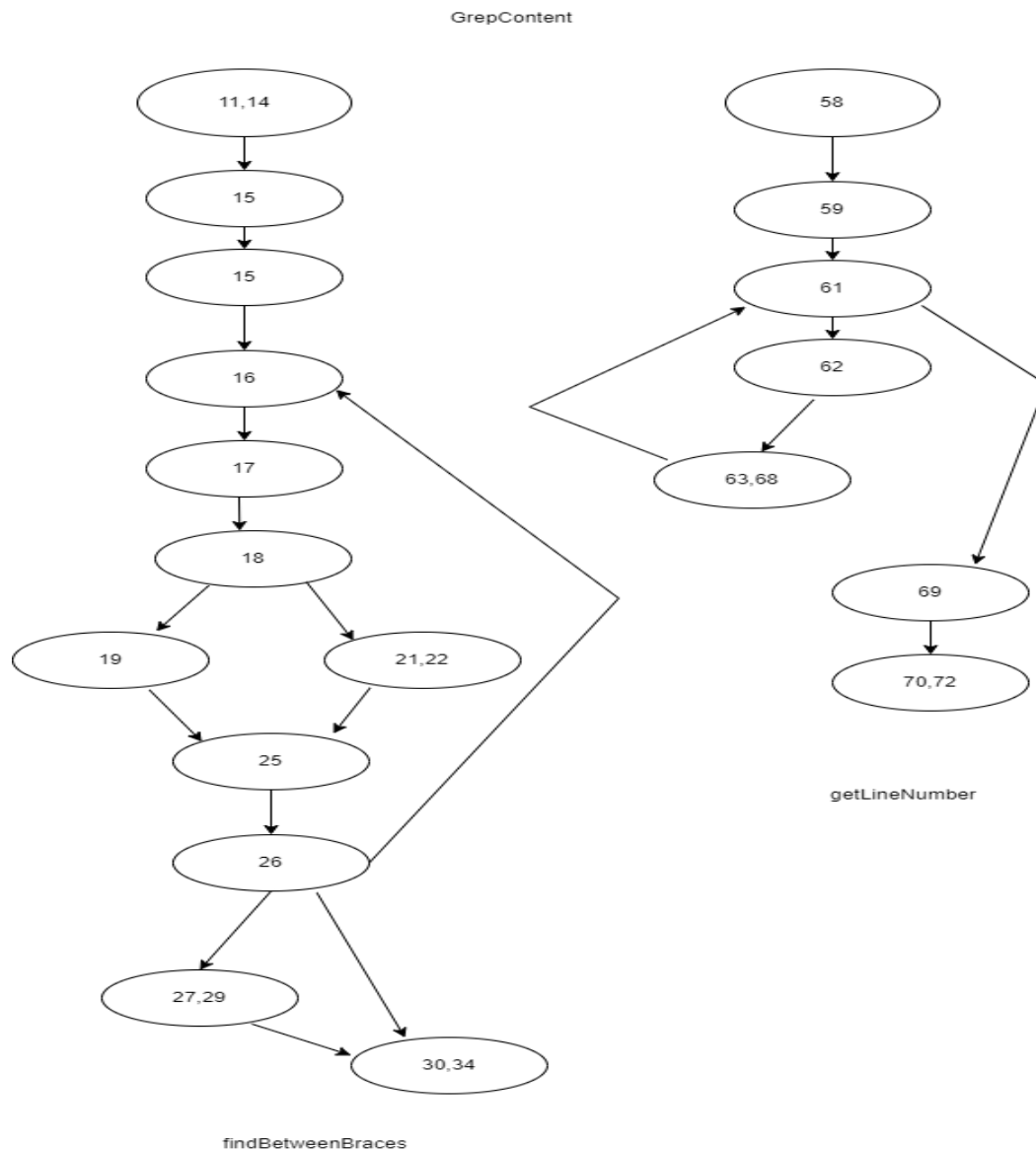
Figure -2: DD Graph of GrepContent

Now for Find between braces	Now for getlinenumber
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
$=12\text{-}14+2*1$	$=7\text{-}7+2*1$
$=0$	$=2$

Now the total complexity will different because above two graphs are not connected.
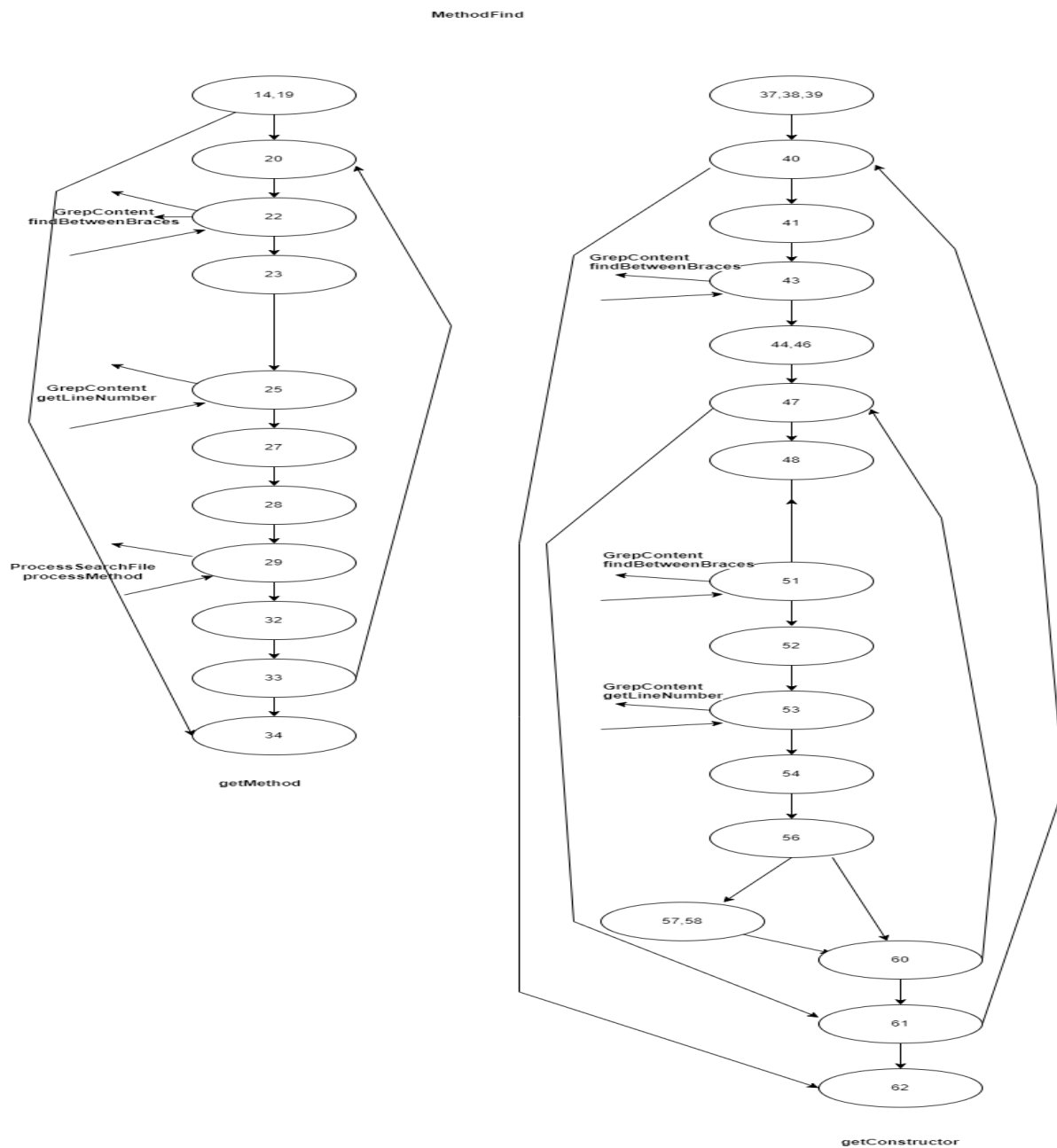
3. Here DD Graph of MethodFind .

Figure-3:DD graph of method find.

Now for getmethod	Now for getconstructor
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
$=11-12+2*1$	$=18-20+2*1$
$=1$	$=2$

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

4. Here DD Graph of processSearchfile.

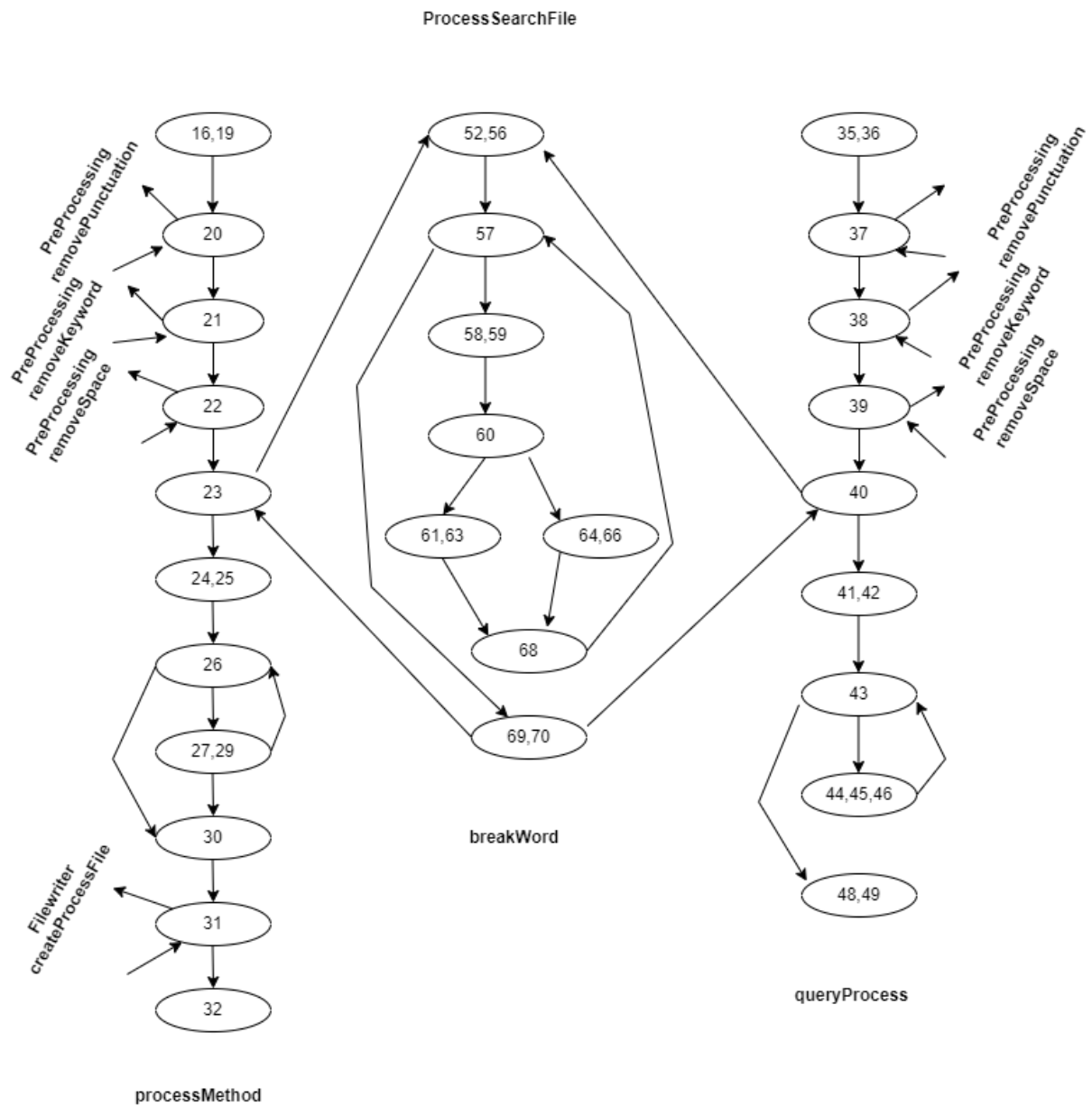Figure-4:DD graphs of processSearchfile.

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

Now the total complexity

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

$=28-30+2*3$

$=4$

5. Here DD Graph of processSearchfile.
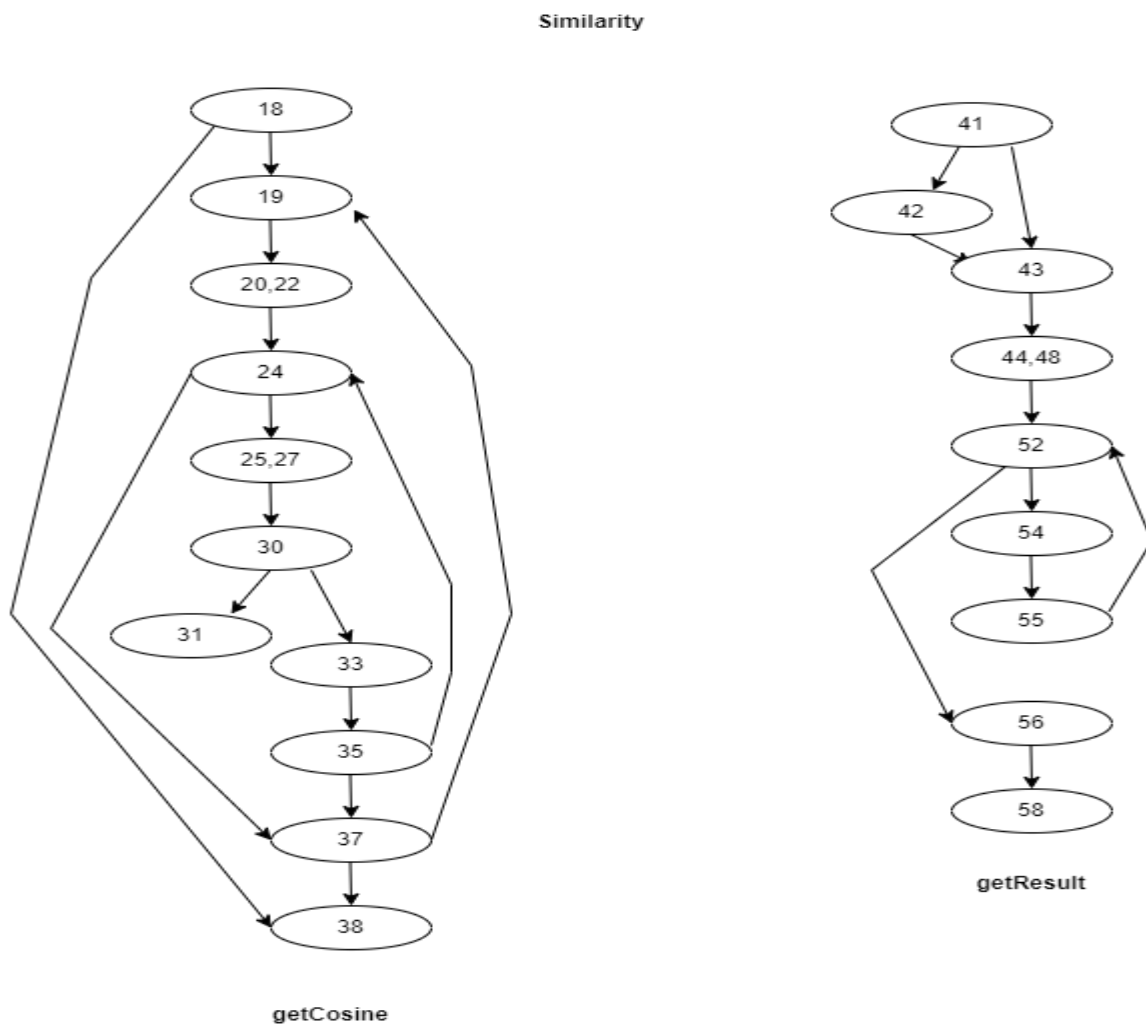
Figure-5:DD graphs of similarity.

Now for getcosine	Now for getresults
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
$=12-14+2*1$	$=9-10+2*1$
$=1$	$=1$

Now the total complexity will different because above two graphs are not connected.
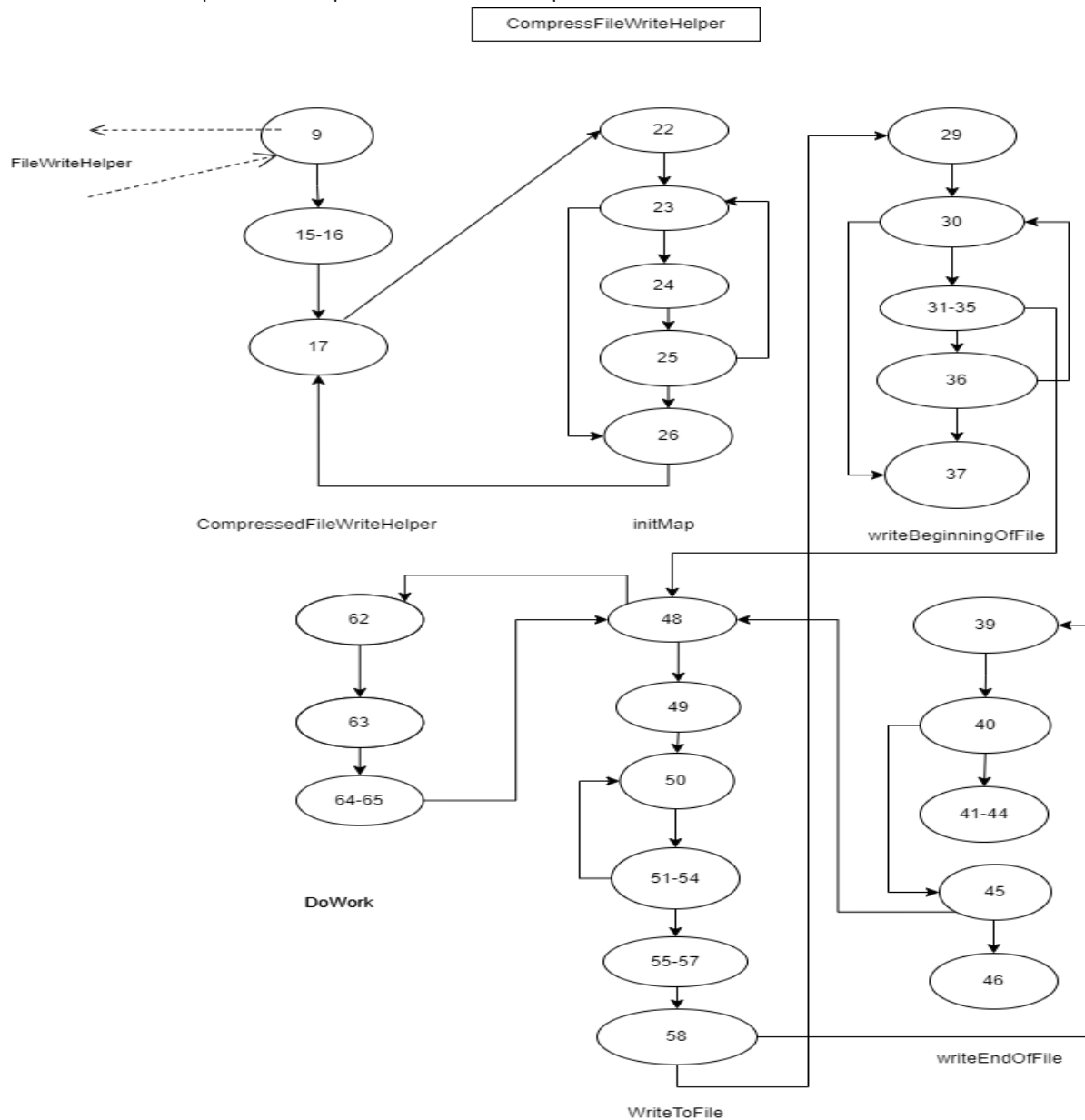
6. Here DD Graph of CompressFilewriteHelper.

Figure-6:DD graphs of CompressFilewriteHelper

Now for CompressFilewriteHelper $V(G) = e - n + 2p$ $\quad$ =3-2+2*1 $\quad$ =3	Now for intmap $V(G) = e - n + 2p$ $\quad$ =5-6+2*1 $\quad$ =1	Now for writebeginingfofile $V(G) = e - n + 2p$ $\quad$ =5-6+2*1 $\quad$ =1
For dowork $V(G) = e - n + 2p$ $\quad$ =3-2+2*1 $\quad$ =3	Now for getfile $V(G) = e - n + 2p$ $\quad$ =6-6+2*1 $\quad$ =2	Now for seasrchresult $V(G) = e - n + 2p$ $\quad$ =5-4+2*1 $\quad$ =3

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

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

$\quad$ =8-8+2*2

$\quad$ =4

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

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

$\quad$ =9

7. Here DD Graph of Compress.
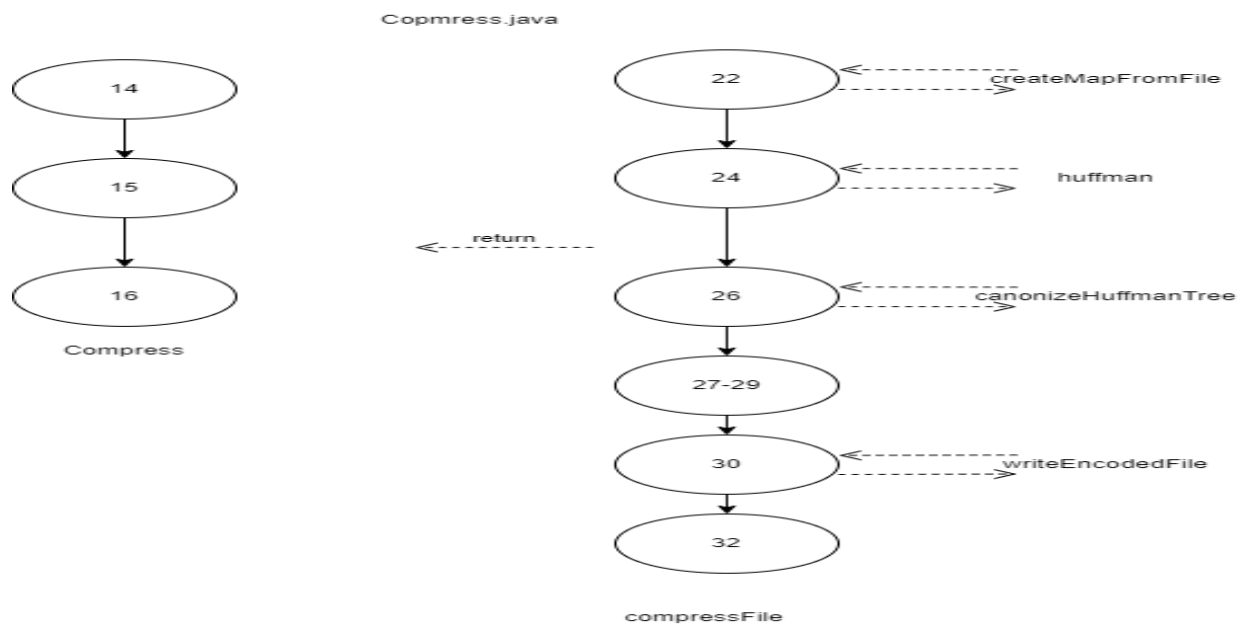
compressFile

Figure-7:DD graphs of Compress.

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

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

8. Here DD Graph of decompressfilewritehelper.

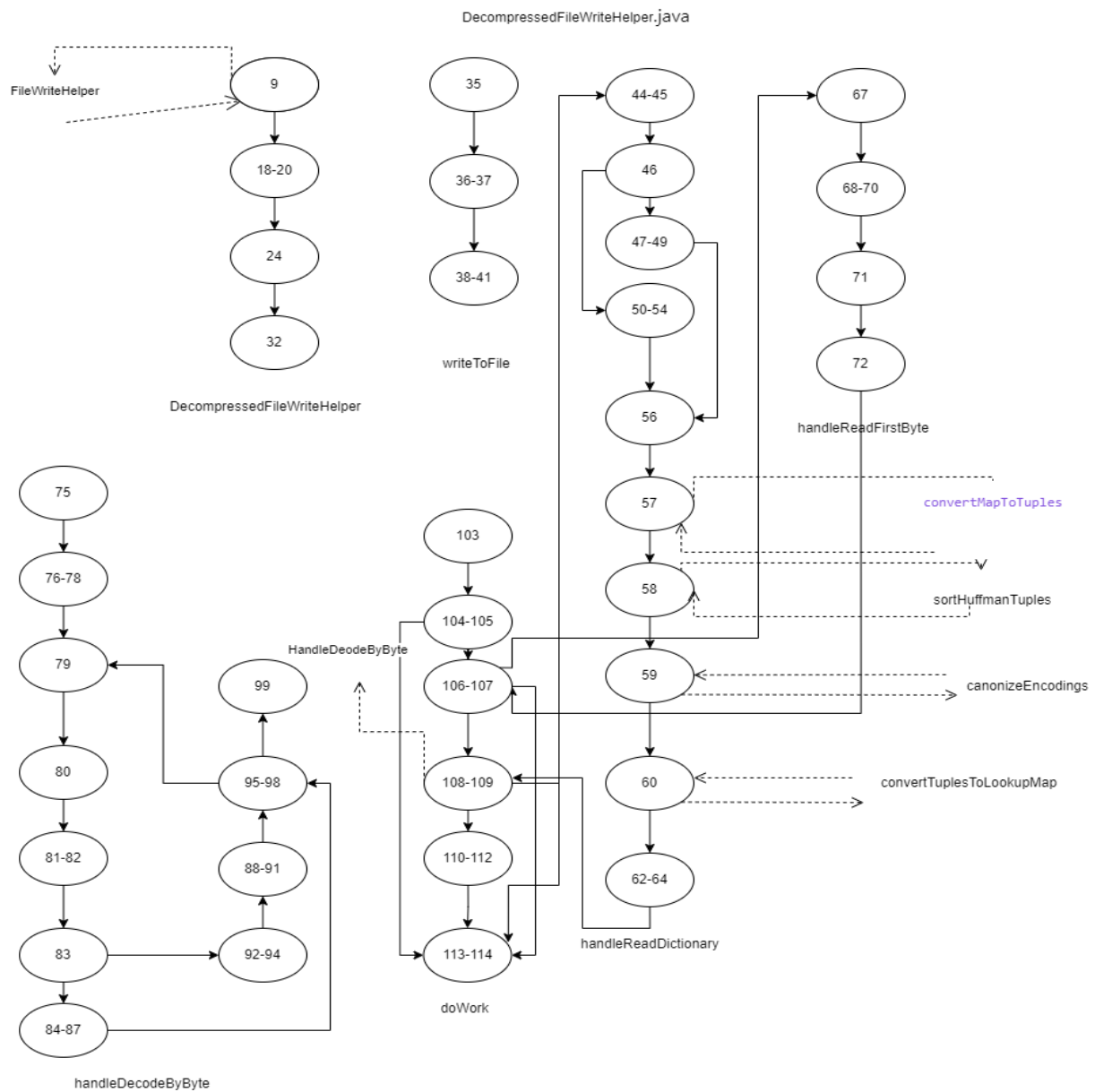Figure-8:DD graph of decompressfilewritehelper.

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

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

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

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

$\quad$ =6

9. Here DD Graph of FilecountReader.
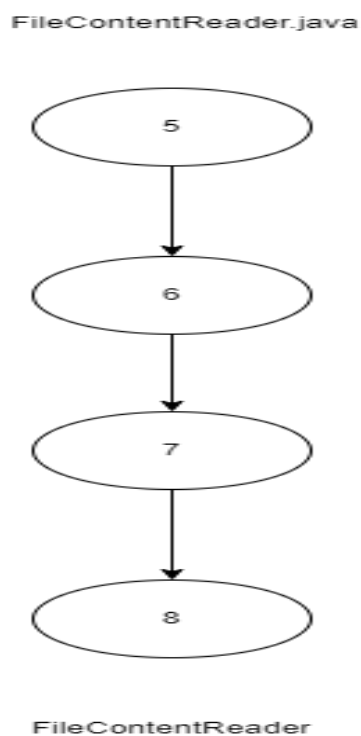
Figure-9:DD graphs of FilecountReader

10. Here DD Graph of FilewriteHelper.

FileWriteHelper .java
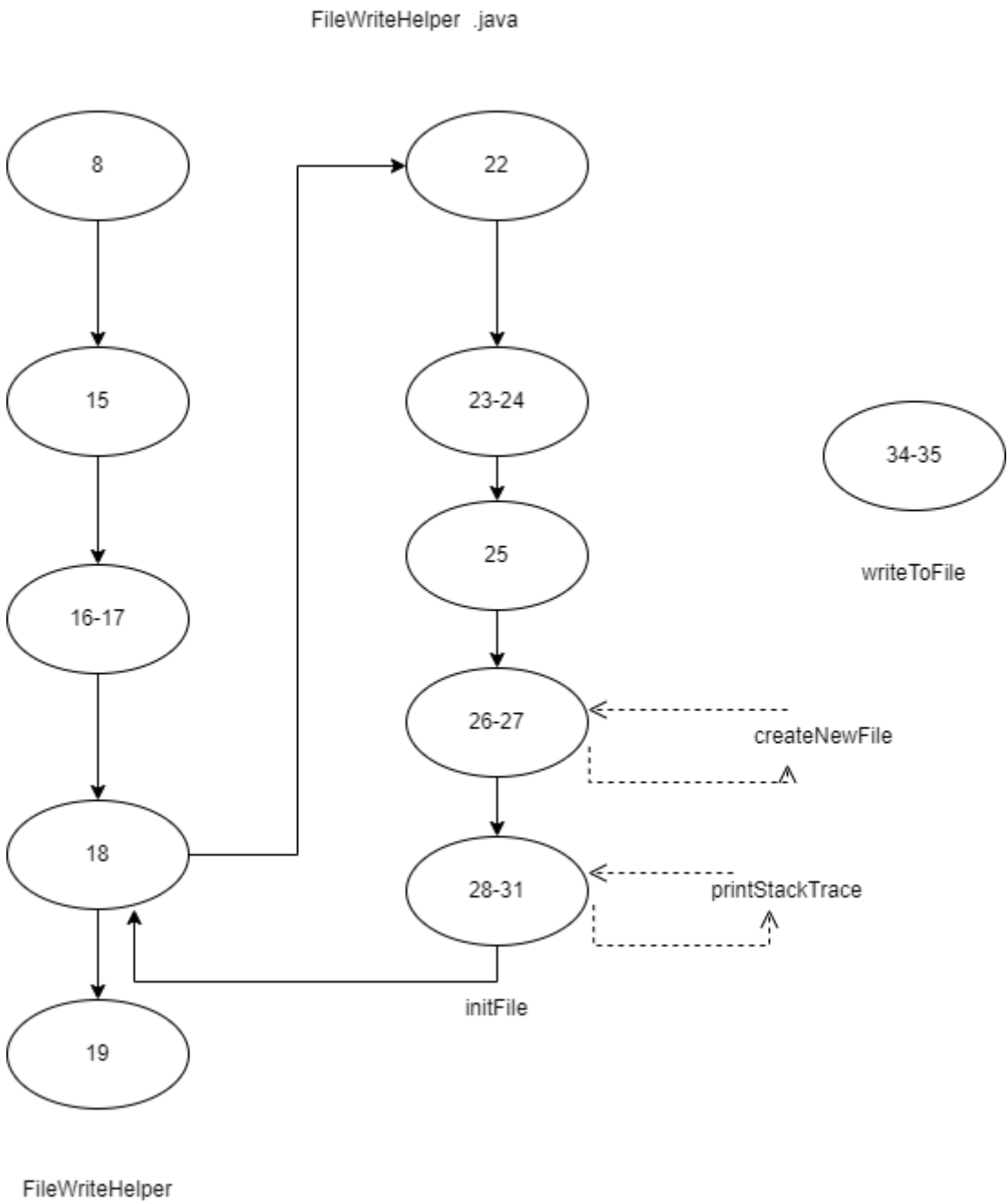
FileWriteHelper

Figure-10:DD graphs of FilewriteHelper.

FilewriteHelper	intfile	Now for writetofile
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
=5-4+2*1	=5-4+2*1	=1-0+2*1
=2	=2	=3

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

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

=10-8+2*2

=6

11. Here DD Graph of FrequencyMapCreate.

FrequencyMapCreate.java
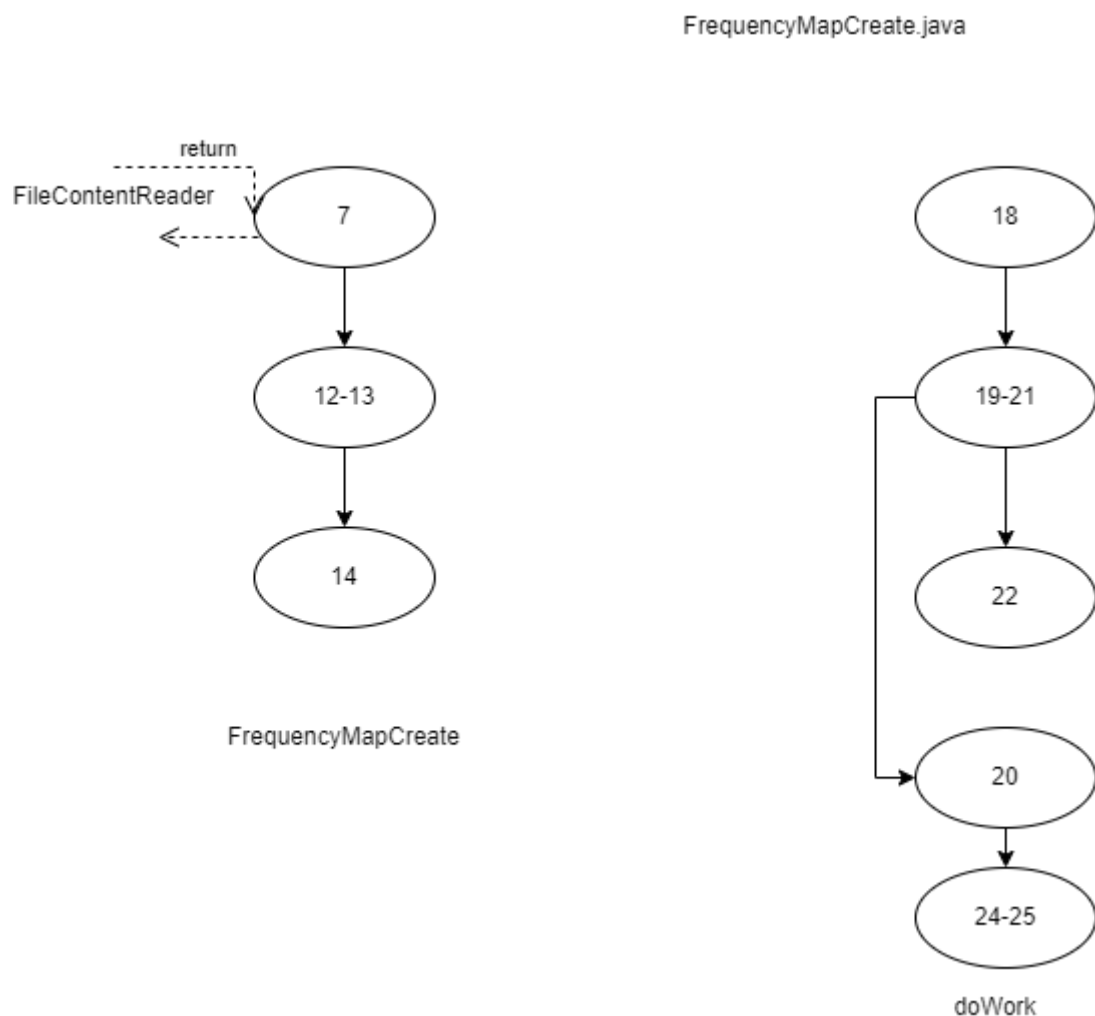
Figure-11: DD Graph of FrequencyMapCreate.

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

Now the total complexity will different because above two graphs are not connected
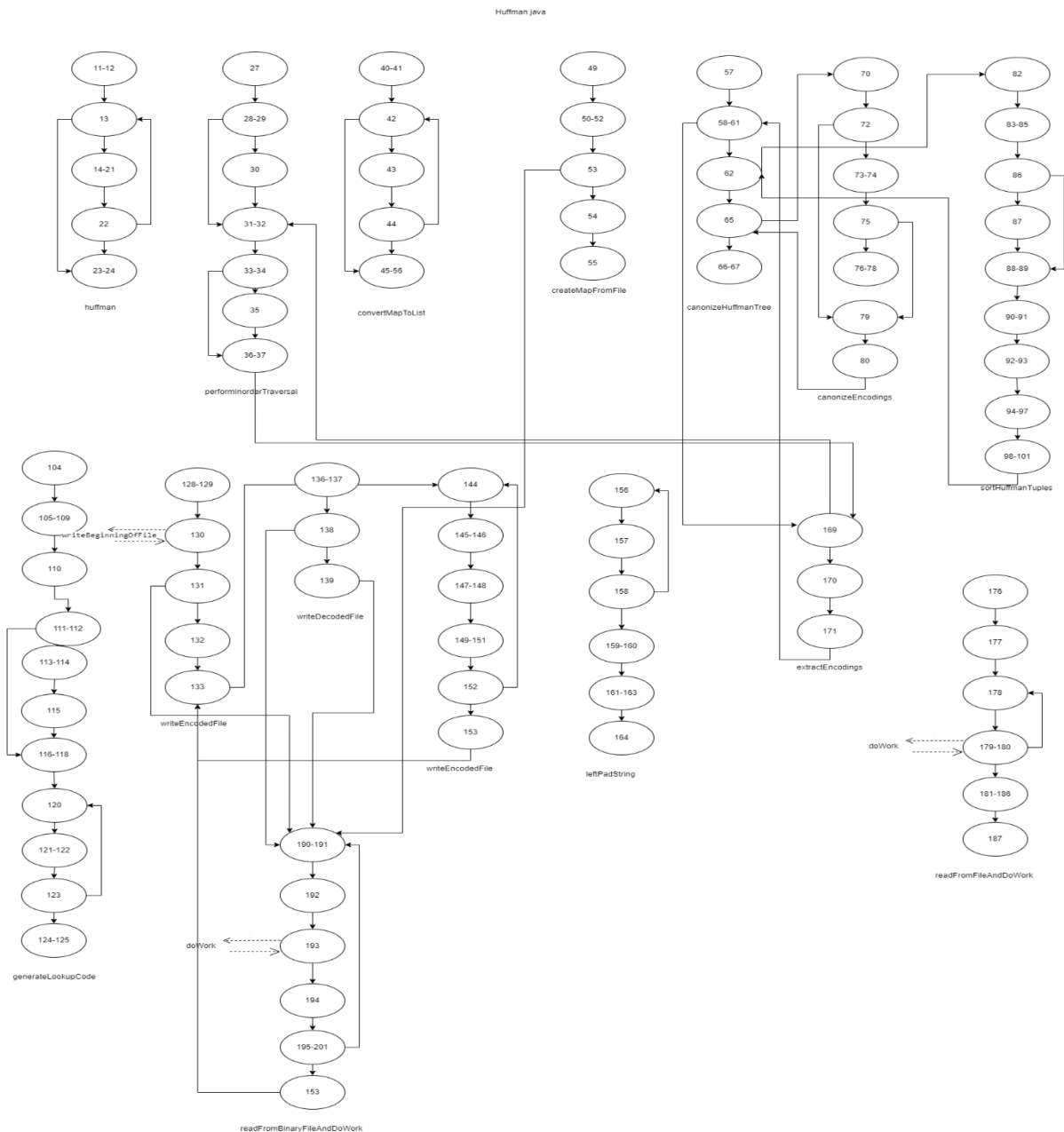
12. Here DD Graph of Huffman.

Fogure-12:DD graphs of Huffman.

Huffman $V(G) = e - n + 2p$ =5-6+2*1 =1	Now for performanceTravels $V(G) = e - n + 2p$ =7-8+2*1 =1	convertMaptolist $V(G) = e - n + 2p$ =5-6+2*1 =1
For CreateMapfromFIle $V(G) = e - n + 2p$ =5-4+2*1 =2	Now for canonizehuffman $V(G) = e - n + 2p$ =5-4+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 $V(G) = e - n + 2p$ =11-12+2*1 =2	writencodedfile $V(G) = e - n + 2p$ =5-4+2*1 =2
readfrombinaryanddowork $V(G) = e - n + 2p$ =6-6+2*1 =2	writedecodefile $V(G) = e - n + 2p$ =3-2+2*1 =3	leftpadString $V(G) = e - n + 2p$ =6-6+2*1 =2

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

$V(G) = e - n + 2p$=51-52+2*10=19
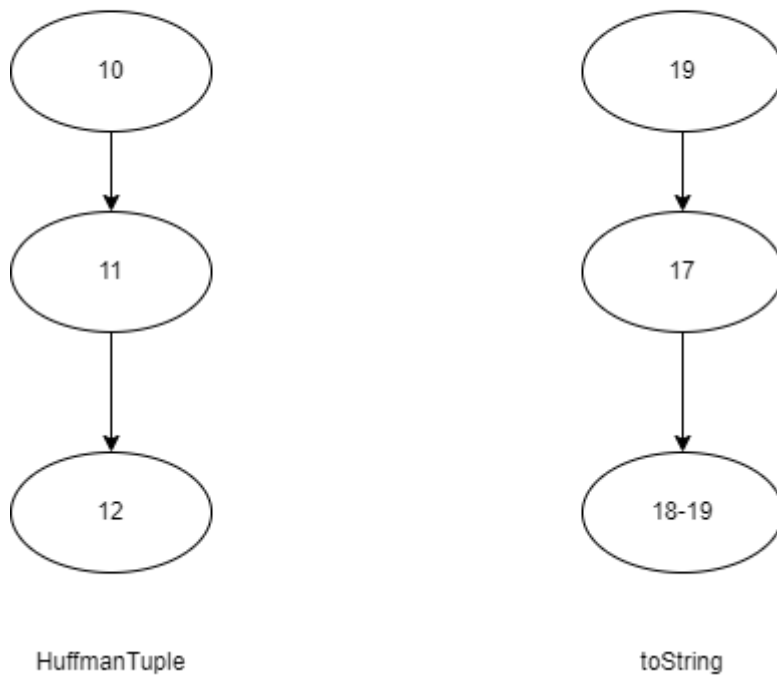
13. Here DD Graph of HuffmanTuple.

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.
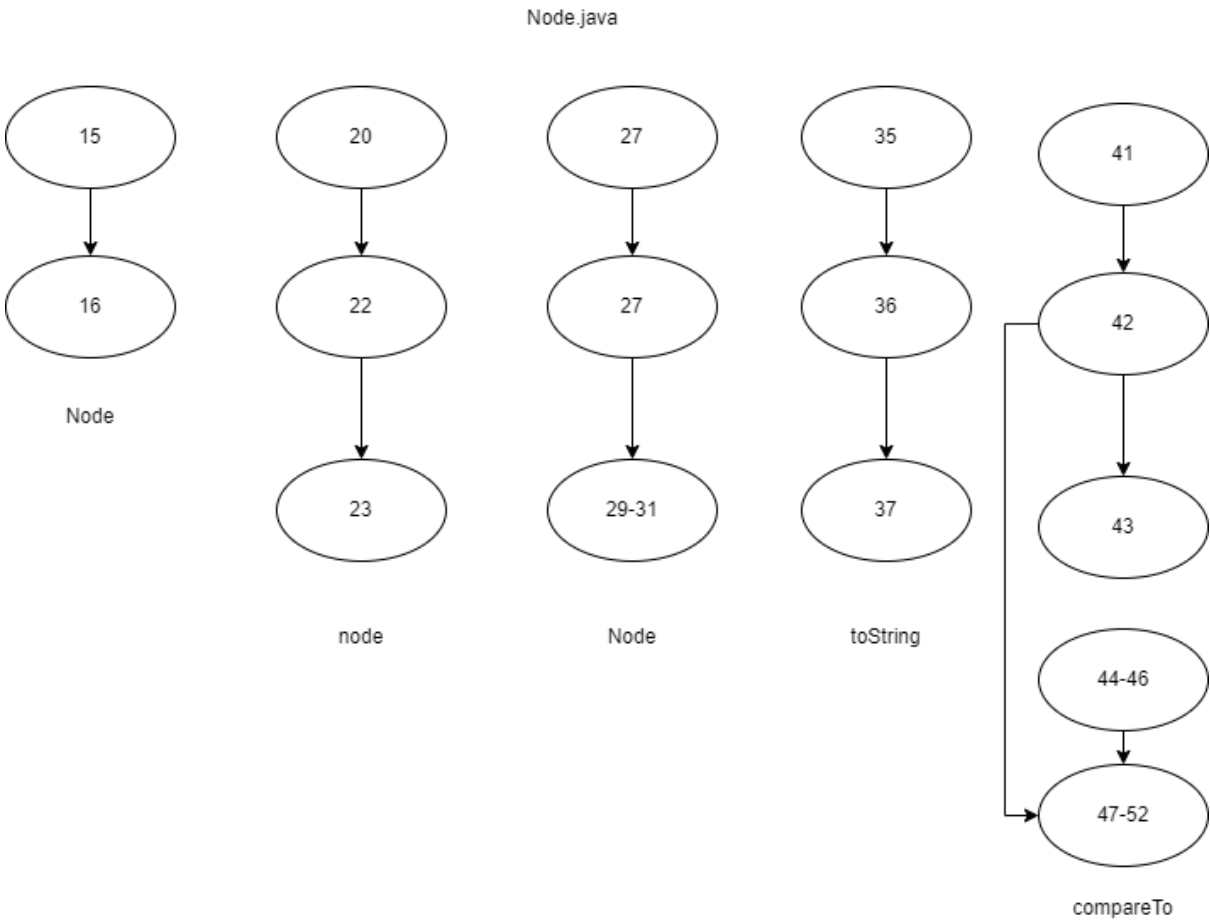
14. Here DD Graph of Node.

Figure-14:DD graphs for Node

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

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

15. Here DD Graph of Code clone

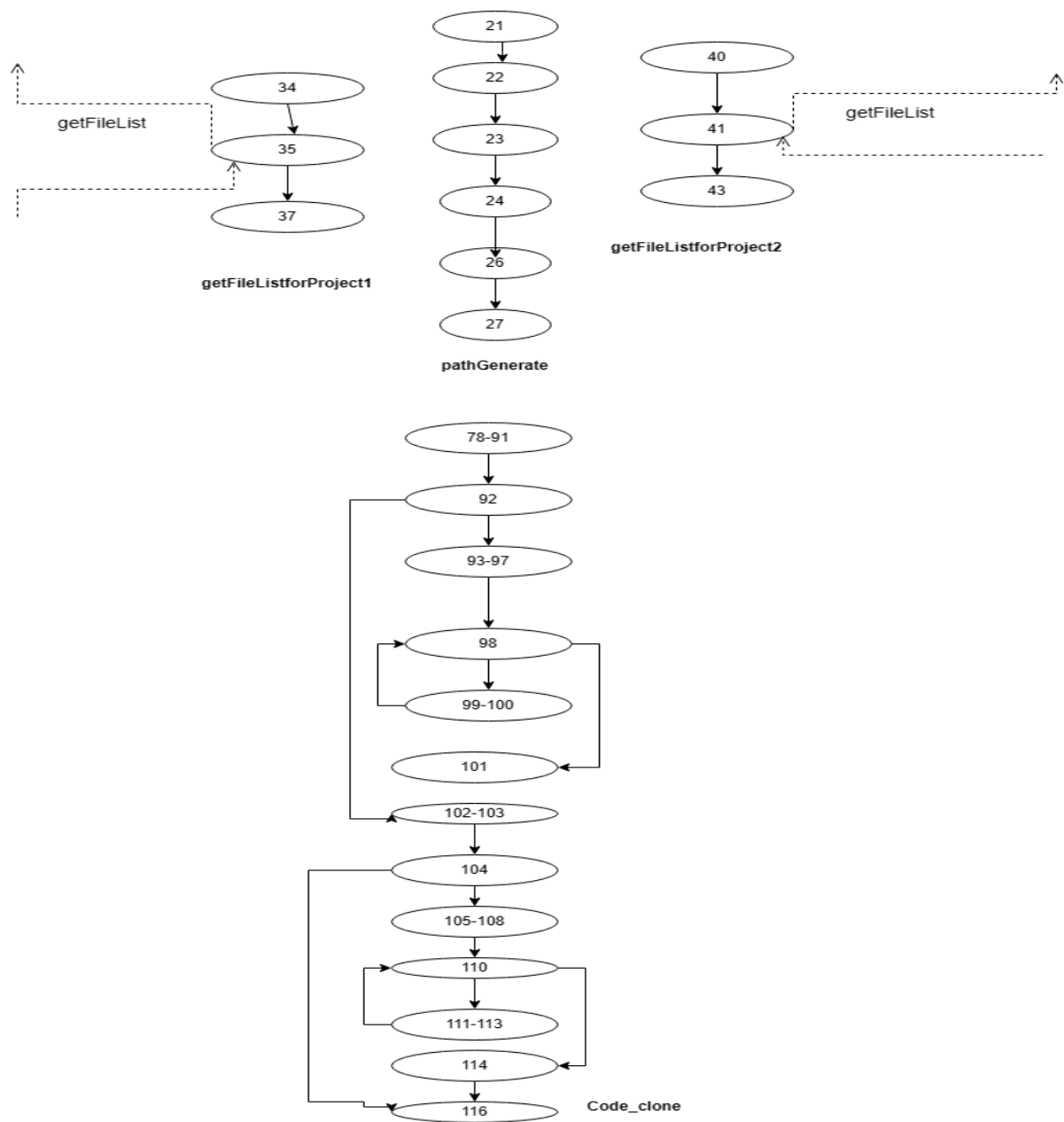Figure-15:DD graphs Code clone

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

Here no relation are fount with more method so complexity will own for each module.
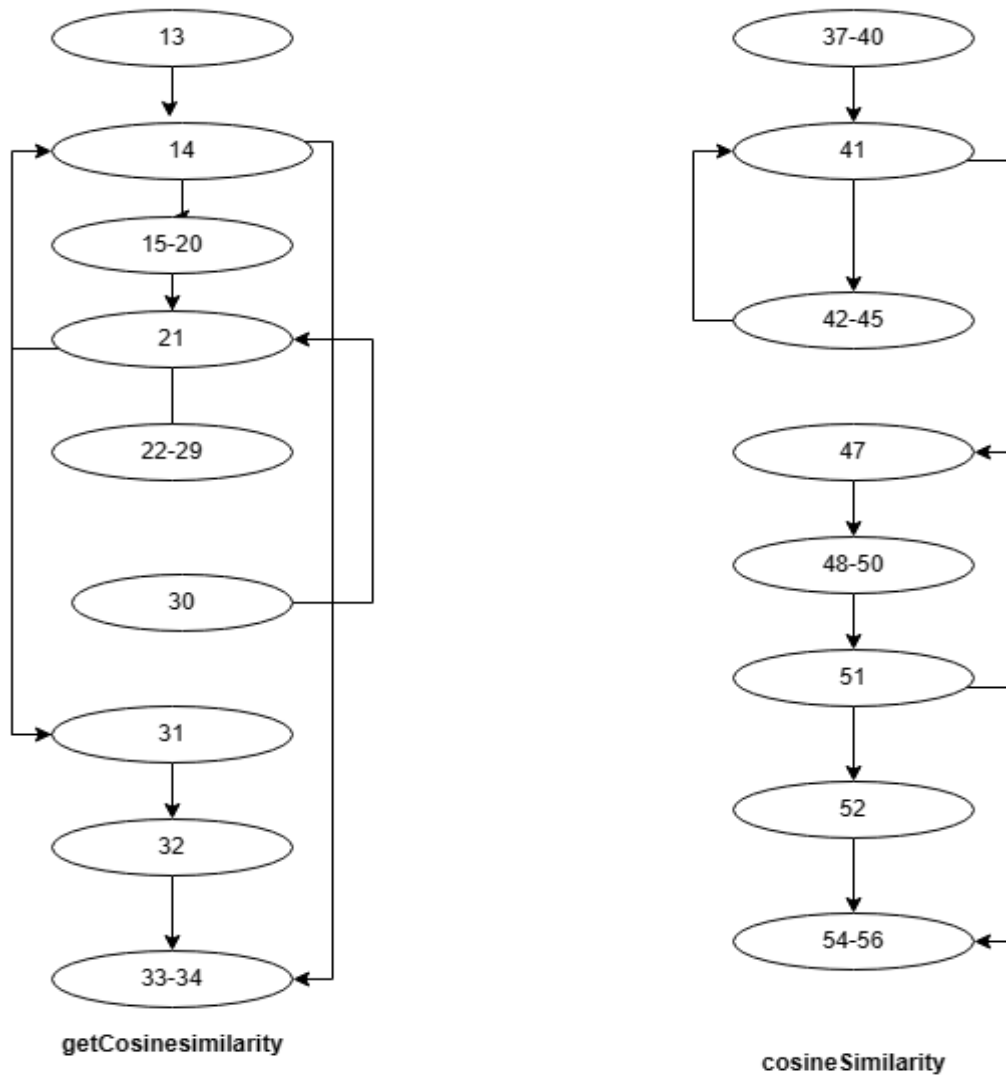
16. Here DD Graph of cosine similarity

Figure-16:DD graphs for cosine similarity.

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

Here no relation are fount with more method so complexity will own for each module.
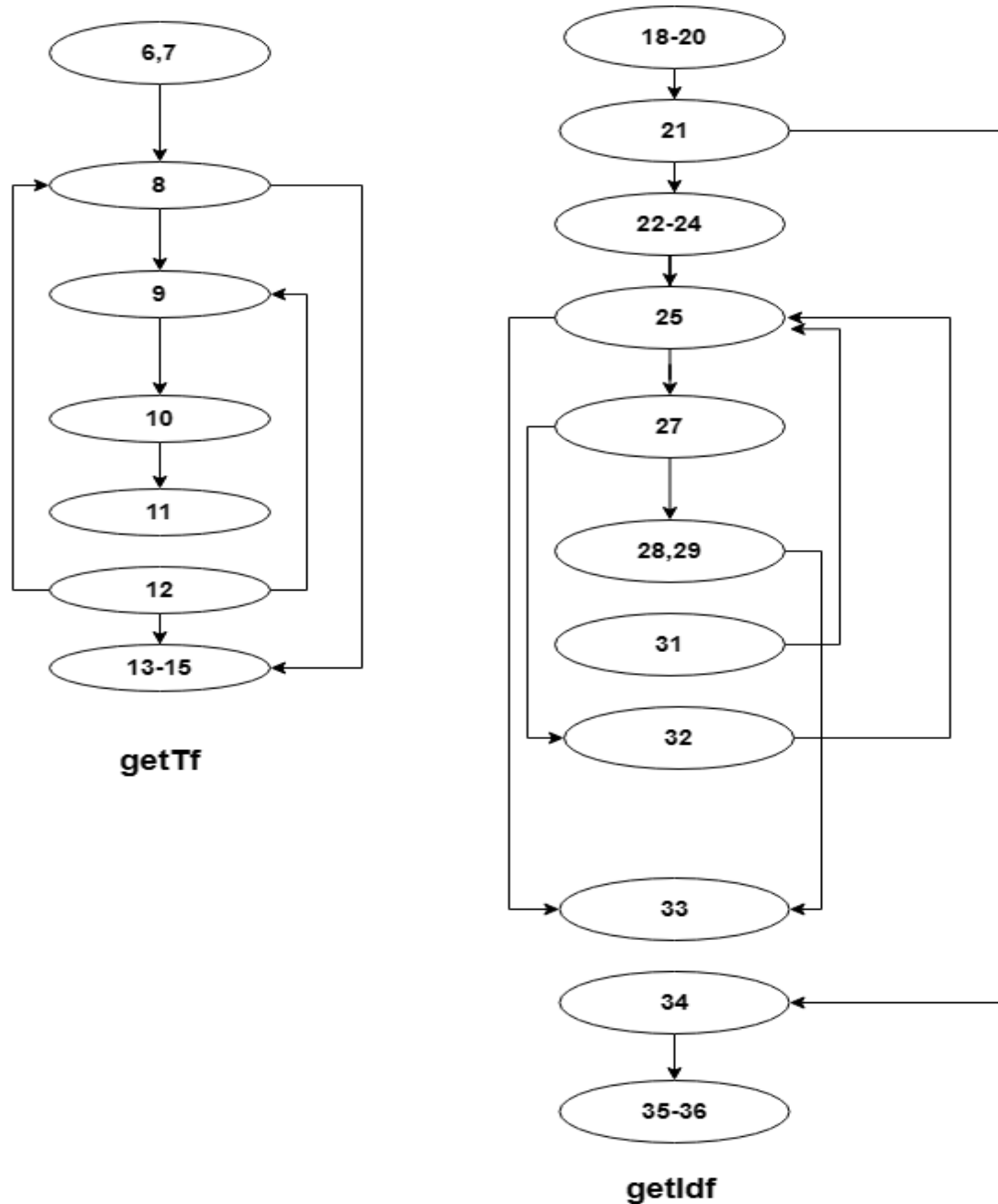
17. Here DD Graph of getidfield.

getTf

getIdf

Figure-17:DD graphs for getidfield

For getcosinesimilarity	pathgenerate
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
$=7\text{-}8\text{+}2\text{*}1$	$=11\text{-}12\text{+}2\text{*}1$
$=1$	$=1$

Here no relation are fount with more method so complexity will own for each module.
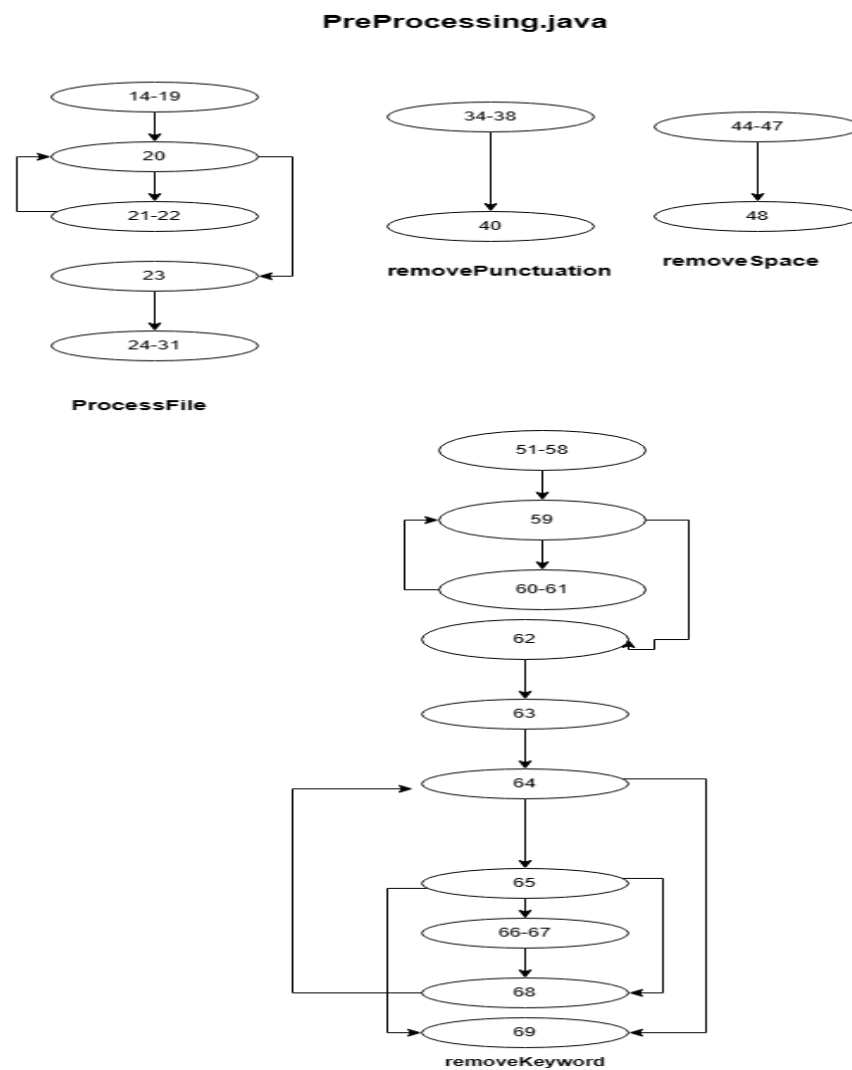
18. Here DD Graph of preprosessing.

Figure-18:DD graphs for preprosessing.

For processfile	removepuntuation	removespace
$V(G) = e - n + 2p$	$V(G) = e - n + 2p$	$V(G) = e - n + 2p$
=5-5+2*1	=2-1+2*1	=2-1+2*1
=2	=3	=3

For removekeyword
$V(G) = e - n + 2p$
=12-13+2*1
=1

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

19. Here DD Graph of tfidfcalculate.
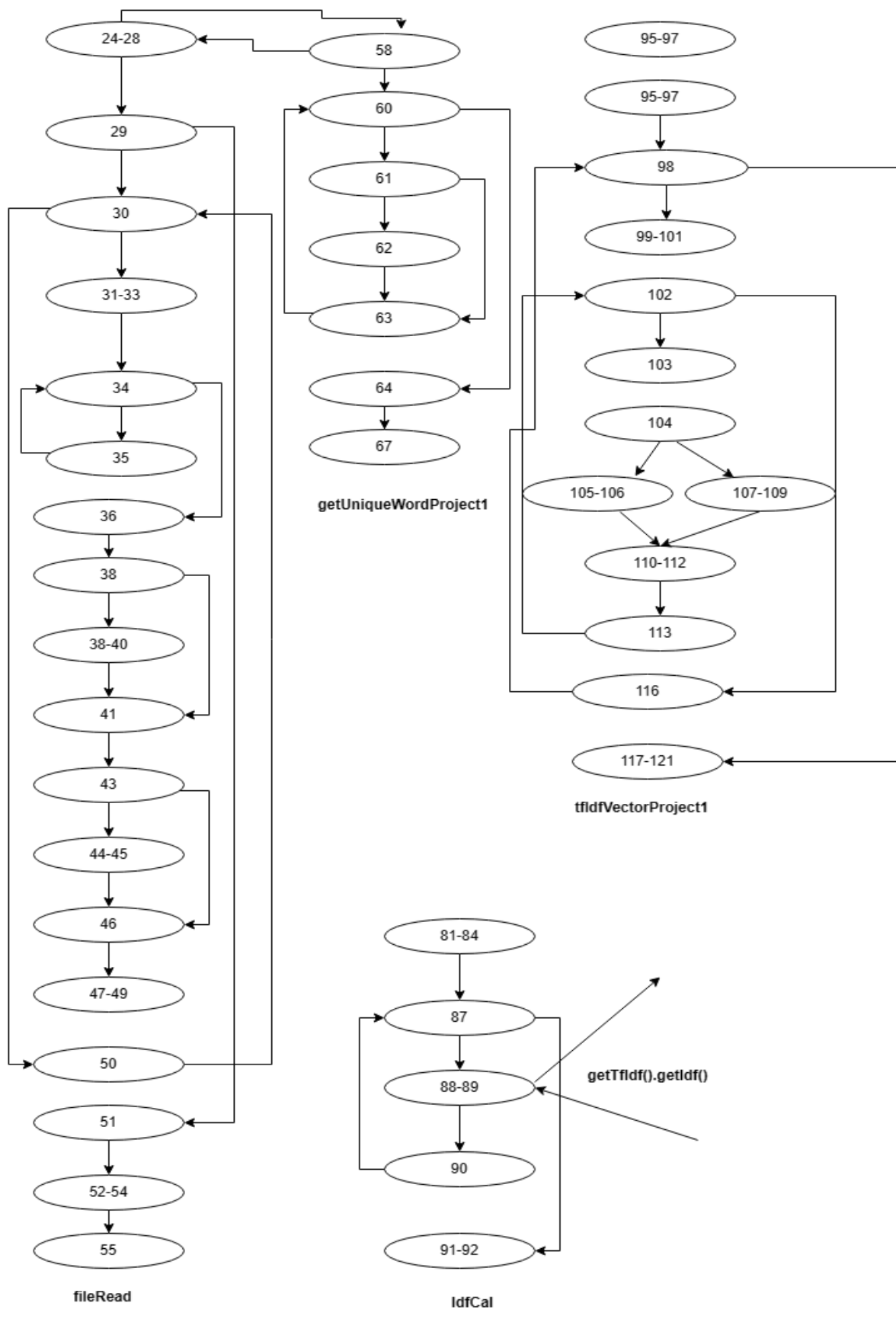

Figure-20: DD Graph of tfidfcalculate.

For fileread

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

$= 19 - 21 + 2*1$

$= 0$

getuniquewordprojectone

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

$= 7 - 8 + 2*1$

$= 1$

tfidVectorProject1

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

$= 13 - 15 + 2*1$

$= 0$

For idcal

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

$= 5 - 5 + 2*1$

$= 2$

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

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

$= 25 - 29 + 2*2$

$= 20$