1. Generate UML activity diagram using visual paradigm tool.
2. Generate XMI (XML Metadata Interchange) from UML activity diagram.
3. Generate adjacency matrix by using the extracted information by parsing the XMI code.
4.
    a. Initially ***soil (iwd)=0***, as initially intelligent water drop does not contain any amount of soil with it and Set initial velocity of iwd to 100.
    //soil(iwd) represents the soil present with intelligent water drop (iwd).
    b. Initially set **Vp(iwd) ={}**, and α={}. Also, set Status of all nodes in graph as undiscovered and Cost [path]=0.//**Cost[path] represents cost of test path.**
    c. Calculate cyclomatic complexity of each node as:
    $$CC(i)= \sum_{i=1 \text{ to } n} \text{(outdegree of vertex i)}$$
    d. Initially set Status [1] = visited. Store node '1' to α{}.
    e. set $a_{vel} = 1$.

5. While all the vertices in the adjacency matrix are not discovered (a vertex is said to be discovered if all its adjacent vertices have been visited) follow the following steps:
    5.1 Calculate ***soil(i)= soil (iwd)+ CC(i) + decision(i)*** // soil(i) represents the soil present at node 'i'. The value of decision (i) is '1' if it is a decision node otherwise the value is '0'.
    5.2 N(i) represents the number of nodes in the neighborhood(i).
        a. If N(i) is containing only single node i.e 'j', then set Status[j]= visited and also update the visited set α {}. Iwd will follow node 'j' in the path by calculating Soil(i,j) using the below mentioned formula.
        Soil(i,j)= {fan_in(i) + fan_out(j)}/μ(j)+1
        b. else N(i) is containing more than one node i.e j, k,. etc, then check the status of all the nodes in N(i).
            i) If more than one node is unvisited in N(i) then, 'iwd' at node 'i' will select the next node in its path on the basis of decisionfactor value of edges associated with node 'i'. Iwd will select the edge which is having less amount of soil in its way; therefore it will select the edge having less decision factor value. Set Status[j] = visited and also update the visited set α {}
            DecisionFactor_iwd (i,j)= soil(i,j)/{CC(j) + (N1- j)}
            ii) If N(i) is containing only one unvisited node then set status [j] = visited also update the visited set α {}. Calculate
            Soil(i,j)= {fan_in(i) + fan_out(j)}/μ(j)+1

    5.3 As iwd moves from one node to another node, the velocity of iwd will keep on changing on the basis of following formula:
    $$Vel\ (i,j,iwd)=\{vel(i,j,iwd)+(a_{vel}/soil(i)+soil(i,j))\}$$
    Velocity of iwd will depend upon the amount of soil present at current node and soil present at edge (i,j) in the path, in which iwd is flowing.

5.4 Calculate distance covered by iwd while moving from node i to node j. Distance covered by iwd specifies the total number of edges traversed by that particular water drop while flowing from root node to node j.
Dist_iwd= |α|-1

5.5 Calculate time taken by iwd in coming from node 'i' to node'j'.
Time (i,j, iwd)= Dist_iwd / vel(i, j, iwd)

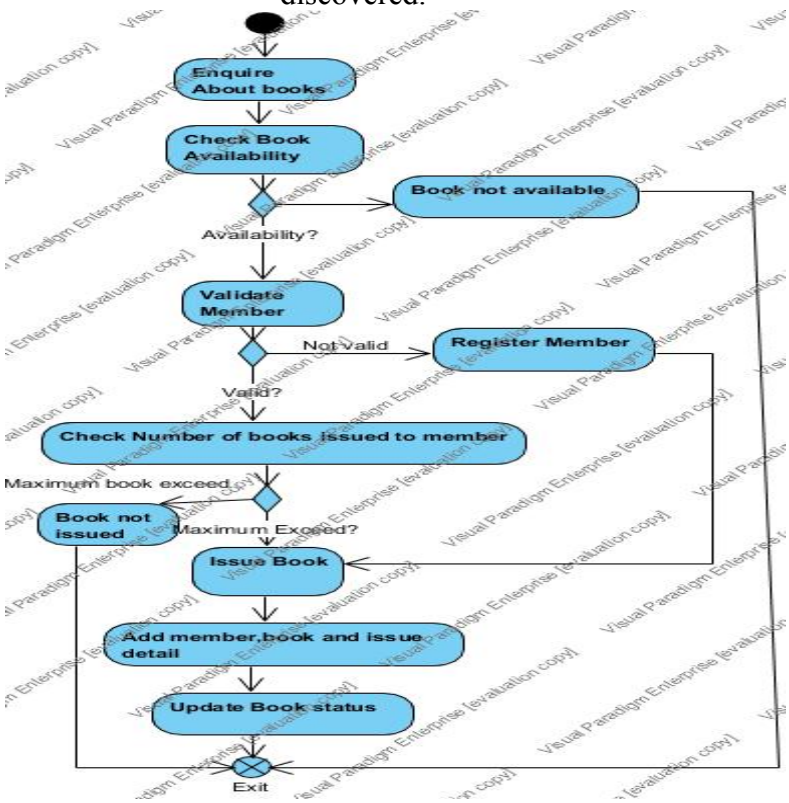
5.6 Intelligent water drop carries some amount of soil with it while moving from one node to another node in stream therefore calculate soil present at iwd as:
Soil(iwd)= Soil(i) / {Time (i,j, iwd)+ vel(i,j,iwd)}

5.7 Since, iwd carries some amount of soil with it, therefore it will decrease some amount of soil from its parent node in stream. Thus remaining soil at node 'i' is: ***Soil(i) = Soil(i)- Soil(iwd)***.

5.8 ***a. Add path 'i->j' in V_p(iwd).***
b. Cost[path]= Cost[path]+ updated_soil(i)+ soil(i,j)

5.9

a. Check if node 'j' is not a leaf node then set node 'j' as current node 'i' to be followed by 'iwd'. If all the nodes in N(i) has been visited, then set status[i]= discovered. Follow the same steps 5.1 to 5.9 in order to generate non-redundant paths.
b. If next node to be traversed by 'iwd' is a leaf node, then stop the procedure here and update the visited path list with the corresponding path and print the path. Backtrack in adjacency matrix until it found next undiscovered node in the matrix and follow the same procedure from steps 5.1 to 5.9 until all nodes in adjacency matrix have been discovered. If all the nodes in N(i) has been visited then set status[i]= discovered.



SNo.	Fan_ou	Fan_i	CC	Subgraph value of Node(i)

	t	n		
1	1	0	17	14
2	1	1	16	13
3	1	1	15	12
4	2	1	14	11
5	1	1	12	1
6	1	1	11	9
7	2	1	10	8
8	1	1	8	4
9	1	1	7	7
10	2	1	6	5
11	1	1	4	1
12	1	2	3	3
13	1	1	2	2
14	1	1	1	1
15	0	2	0	0

Iteration 1:
Initially soil at water drop is nil, therefore value of soil(iwd) is set to zero. Cost[path] is also set to zero. Set status[1] = visited. ‘α’ store the visited node ‘a’ and | α| is 1.
Soil(i) = 0+ 17+0 (as node 1 is not the decision node therefore its decision value is zero). N(i) represents the number of nodes adjacent to node 1. According to figure 5, N(1) = {2}. Set Status[2] = visited and {α}={1,2}
Soil(1,2)= {(0+1)/(1+13) }= 0.071
Vel(1,2,iwd) = {100+ 1/(17+0.071)}= 100.058
Dist_iwd= 2-1=1
Time= 1/100.058= 0.0099
Soil(iwd)= {17/(0.0099+100.058)}= 0.1698
Soil(1)= 17-0.1698= 16.83
Add path 1->2 in Vp{}, Vp={1,2}
Cost[path]= 0+16.83+0.071= 16.901
As iwd has reached now on node ‘2’ in the graph and it is not the leaf node therefore set status[1] = discovered, and for calculation of further nodes in the path it will again follow the same steps.
Soil[2] = 0.1698+16+0= 16.1698

N(2)= {3}, ‘2’ is not the decision node therefore:
Soil(2,3)= (1+1)/ (1+13)=0.153. set status of ‘3’ = visited and α={1,2,3}
Vel(2,3,iwd)= 100.058 + {1/(16.1698+0.153)}= 100.0612
Dist_iwd= 3-1 =2
Time= 2/ 100.0612=0.019
Soil(iwd)= {16.1698/(0.019++100.0612)}= 0.1615
Soil(2) = 16.1698-0.1615 = 16.0082
Vp= 1->2->3
Cost[path]= 16.901+16.0082+0.153= 33.0622
Now iwd reached node’3’ in the graph and it is not the leaf node status[2]=discovered, again
Soil(3)=0.1615+15+0= 15.1615
N(3)={4}, containing only single node then
Soil(3,4)=3/(11+1)=0.25. Set status[4] = visited and α={1,2,3,4}
Vel(3,4,iwd)= {100.0612+ 1/(0.25+15.1615)}= 100.126
Dist_iwd= 4-1=3
Time=3/100.126= 0.029
Soil(iwd)= {15.1615/(0.029+100.1264)}= 0.1513.
soil(3)=15.1615-0.1513= 15.0101
path=1->2->3->4
cost= 33.0622+15.0101+0.25= 48.32
Status[3]= discovered. As node ‘4’ is not the leaf node then
Soil(4)=0.1513+14+1 // as node ‘4’ is the decision making node therefore its decision value is ‘1’.
N(4)= {5,6}, as it is contains two nodes in its neighborhood therefore it selects the next edge in path on the basis of decision_factor value of edges associated with it.
Soil(4,5)= 2/(1+1)= 1
Soil(4,6)= 2/(1+9)=0.2
Decision_factor(4,5)= 1/(12+10)= 1/22=0.045
Decision_factor(4,6)= 0.2/(11+9)= 0.0095
Edge (4->6) is having less decision _factor value therefore iwd will proceed to node ‘6’ in the graph and set status[6] = visited. Set α= {1,2,3,4,6}
Vel(4,6,iwd)= 100.1264+{1/15.15+0.2}=100.1911
Dist_iwd= 5-1=4
Time= 4/100.1911= 0.039
Soil(iwd) = 15.1513/(0.039+100.1911)= 0.1511
Soil(4)= 15.1513-0.1511= 15.000
Vp= 1->2->3->4->6
Cost= 48.32+15.000+0.2=63.52
As node ‘6’ is not the leaf node but all the nodes in the N(4) are not visited yet so status[4] = undiscovered.
Soil(6)= 0.1511+11+0= 11.1511
N(i)={7}
Soil(6,7) = {(1+2)/(1+8)}= 0.333, set status [7]= visited, α= {1,2,3,4,6,7}
Vel(6,7,iwd)= 100.1911+{1/(11.1511+0.333)}= 100.27
Dist_iwd= 6-1=5
Time= 5/ 100.27 = 0.049
Soil(iwd)= {11.1511/ (0.049+100.27)}= 0.111
Soil(7)= 11.1511-0.111= 11.03

Vp= 1->2->3->4->6->7
Cost[path]= 63.52+11.03+0.333= 74.88
All the nodes in N(6) has been visited therefore set status[6]= discovered. As node'7' is not the leaf node then:
Soil[7]= 0.111 +10+1= 11.111
N(7)= {8,9}
Soil(7,8)= 2/(1+4)= 0.4
Soil(7,9)= 2/7=0.28
Decision_factor(7,8)= 0.4/{8+(15-8)}= 0.4/15= 0.026
Decision_factor(7,9)= 0.28/(7+9)= 0.0175
Selected edge is (7->9), Set status[9]= visited and update α= {1,2,3,4,6,7,9}
vel(7,9,iwd)= 100.27+{1/(0.2+11.111)}= 100.35
Dist_iwd= 7-1 =6
Time= 6/ 100.35= 0.059
Soil(iwd) = {11.11/(0.059+100.35)}= 0.11
Soil(7)= 11.111-0.11=11
Cost[path] = 74.88+11+0.28= 86.1653
Vp=1-2->3->4->6->7->9
All nodes in N(7) are not visited therefore status of node [7] is still undiscovered. As node'7' is not the leaf node therefore node '9' would the next node to be traversed.
Soil(9) = 0.111+7+0 = 7.111
N(9) ={10}
Soil (9,10) = 3/ 1+5 = 0.5
Status[10] = visited and α= {1,2,3,4,6,7,9,10}
Vel(9,10,iwd) = 100.35 + {1/ (7.111+0.5)} = 100.48
Dist_iwd = 8-1=7
Time= 7/ 100.48 = 0.069
Soil(iwd) = 7.111/ (0.069+100.48) = 0.070
Soil(9) = 7.111- 0.070 = 7.041
Cost [path] = 86.1653+ 7.041+0.5 = 93.70
As all nodes in N(9) have been visited therefore status[9] is discovered. Node (10) is not the leaf node therefore next node to be traversed is node '10'.
Soil(10) = 0.070+1+6 = 7.070
N(10) = {11,12}
Soil(10,11) = 2/2 =1
Soil (10,12) = 2/4= 0.5
Decision_factor (10,11) = 1/ (4+4) = 0.125
Decision_factor (10,12) = 0.5/6 = 0.083
As decision factor of edge (10,12) is less therefore iwd will select the edge (10,12) first to traverse in graph and status[12] is visited and α ={1,2,3,4,6,7,9,10,12}
Vel(10,12,iwd) = 100.48 + {1/ (0.5+ 7.070)} = 100.61
Dist_iwd= 9-1 =8
Time = 8/ 100.61 = 0.079
Soil(iwd) = 7.070 /(0.079+100.61) = 0.070
Soil(10) = 7.070- 0.070= 7
Cost[path] = 93.70+ 7+0.5= 101.20
As all the nodes in N(10) are not visited yet therefore status[10] is still not discovered and next node to be traversed is node '12'.

Soil(12) = 3+0+0.070= 3.070
N(12) = {13}, set status[13] = visited and α= {1,2,3,4,6,7,9,10,12,13}
Soil(12,13) = 3/(1+2)= 1
Vel(12,13,iwd) = 100.61+{ 1/(1+3.070)} = 100.85
Dist_iwd= 10-1 = 9
Time= 9/ 100.85 = 0.089
Soil(iwd) = 3.070/(0.089+100.85) = 0.0304
Soil(12) = 3.070- 0.0304 = 3.039
Cost[path] = 101.20+3.039+1 = 105.23
As all the nodes in N(12) has been visited therefore status[12] = discovered. As node '13' is not the leaf node then
Soil(13) = 2+0+0.0304 = 2.0304
N(13) = {14}, Status[14] = visited and α = { 1,2,3,4,6,7,9,10,12,13,14}
Soil(13,14) = 2/(1+1) =1
Vel(13,14,iwd) = 100.85 + {1/ (1+2.0304)} = 101.17
Dist_iwd= 11-1 = 10
Time = 10/101.17 = 0.098
Soil(iwd) = 2.0304 / (0.98+ 101.17) = 0.0200
Soil(13) = 2.0304 – 0.0200 = 2.010
Cost[path] = 105.23+ 2.010+1 = 108.24
As all the nodes in N(13) has been visited therefore status[13] = discovered. And as node '14' is not the leaf node therefore next node would be discovered by iwd is 14.
Soil(14) = 0+1+0.0200= 1.0200
N(14) ={15}, set status[15] = visited and α= { 1,2,3,4,5,6,7,9,10,12,13,14,15}
Soil(14,15) = 1/1= 1
Vel(14,15,iwd) = 101.17+ { 1/(1+ 1.0200)} = 101.66
Dist_iwd= 12-1 = 11
Time = 11/101.66 = 0.108
Soil(iwd) = 1.020/ (0.108+101.66) = 0.0100
Soil(14) = 1.009
Cost[path] = 108.24 +1+1.009 = 110.258
As all the nodes in N(14) has been visited therefore set status[14]= discovered. As node 15 is the leaf node therefore the process will stop here and will print the path 1->2->3->4->5->6->7->9->10->12->13->14->15 with cost = 110.258
Iwd will now backtrack in adjacency matrix and search undiscovered node to be traversed next. First undiscovered node found is node '10'. Iwd will check for unvisited children of node '10'. N(10) = {11,12}. There is only one child node '11' which is not visited by iwd.
Soil(10) = 7.070
Set status[11] = visited. Update α= {1,2,3,4,6,7,9,10,11,12}
Soil(10,11) = 2/(1+1) = 1
Vel(10,11,iwd) = 100.48 + {1/ 1+7.070} = 100.60
Dist_iwd = 10-1 = 9
Time = 9 / 100.60 = 0.089
Soil(iwd) = 7.070/ (0.089+100.61) = 0.070
Soil[10] = 7.070-0.070 = 7
Cost[path] = 93.70 + 7+ 1 = 101.7

As all the nodes in N(10) have been visited by iwd therefore set staus[10] = discovered. As node ‘11’ is not the leaf node therefore next node to be traversed by iwd is node ‘11’.
Soil(11) = 4+0+0.070= 4.07
N(11) = {15} and α= {1,2,3,4,6,7,9,10,11,12,15}, set status [15] = visited.
Soil(11,15) = 1/1 = 1
Vel(11,15,iwd) = 100.60 + {1/(1+4.07)} = 100.7
Dist = 11-1 = 10
Time= 10/ 100.7 = 0.099
Soil(iwd) = 4.07/ (0.085+100.77) = 0.040
Soil(11) = 4.07-0.040= 4.03
Cost[path] = 101.7+4.03+1= 106.73
Since all nodes in N(11) has been visited therefore set status[11] as discovered. As node ‘15’ is a leaf node therefore iwd will stop its traversing here print the path as : 1->2->3->4->6->7->9->10->11->15 with cost = 106.73.
In similar way iwd will print all other paths also.

Table 1: Number of paths obtain by M_Iwd

S.No.	Test Paths	Cost	Priority
1	1->2->3->4->6->7->9->10->12->13->14->15	110.25	1
2	1->2->3->4->6->7->9->10->11->15	106.73	2
3	1->2->3->4->6->7->8->12->13->14->15	102.60	3
4	1->2->3->4->5->15		4