First, you should create an input file that includes the rules. For instance, if your input file includes the following lines:

A,B,C,D->E
A->B
A,D->C
F
G

This means that node E will wait for the execution of nodes A, B, C, and D to complete. Similarly, Node B will wait for node A to complete. Nodes F and G will not wait for any node to complete, so they should be executed immediately.

Details:

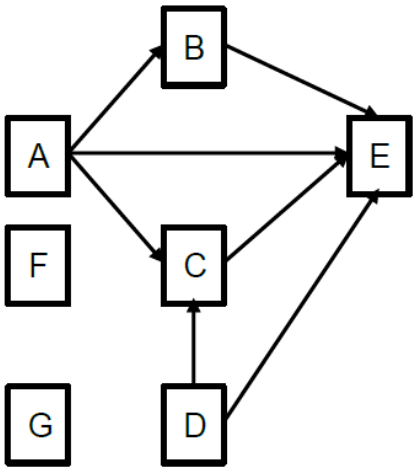
- The Node class should extend the Thread class.
- This class will have a perform() method, which simulates a job with a long execution time.
- The simulation can be done using Thread.sleep(time) method. Time should be passed by a randomly generated integer with a maximum value of 2000 milliseconds.

Sample Outputs for input1.txt

A,B,C,D->E
A->B
A,D->C
F
G

java -jar Project2.jar -i input1.txt

NodeD is being started NodeA is being started NodeB is waiting for A NodeF is being started NodeC is waiting for A,D NodeE is waiting for A,B,C,D NodeG is being started NodeD is completed NodeG is completed NodeA is completed NodeB is being started NodeC is being started NodeC is completed NodeF is completed NodeB is completed NodeE is being started NodeE is completed	NodeG is being started NodeD is being started NodeF is being started NodeA is being started NodeC is waiting for A,D NodeB is waiting for A NodeE is waiting for A,B,C,D NodeG is completed NodeF is completed NodeD is completed NodeA is completed NodeC is being started NodeB is being started NodeB is completed NodeC is completed NodeE is being started NodeE is completed



Graphical representation of those rules is illustrated in the figure.

You can easily see that E will wait for A, B, C, and D.
Similarly, C will wait for A and D.
B will wait for only A.
F and G will not wait for any Node to complete.

Note: The table shows the results of running the input file twice. As can be seen in the table, the order may change, but the completion of the nodes is in accordance with the rules.