```
...
    public void calculatePageRank(nodes)
    {
        double dumpFactor = 0.85;
        // Setting default rank
        nodes.forEach((k, v) -> v.setRank(1 - dumpFactor/nodes.size()));
        // Firt node to visit
        Node node = nodes.entrySet().iterator().next().getValue();
        double nodeRank = 0;
        int totalInteractions = 0 ;
        while(true) {
            double followersRank = 0;
            for (Node follower : node.getIn()) {

                followersRank += follower.getRank()/follower.getOut().size();
            }
            nodeRank = node.getRank()+(dumpFactor* followersRank);
            node.setRank(nodeRank);
            if(Math.random() < dumpFactor && user.getOut().size() > 0)
            {
                node = walk(node);
            }else{
                node = randomWalk(nodes);
            }
            totalInteractions++;
            if(totalInteractions > 70_000_000)
            {
                System.out.println("Max interactions");
                break;
            }
        }
    }

    private static Node randomWalk(Map<Integer,Node> nodes)
    {
        Random random    = new Random();
        List<Integer> keys      = new ArrayList<Integer>(nodes.keySet());
        Integer       randomKey = keys.get( random.nextInt(keys.size()) );
        Node       node     = nodes.get(randomKey);
        return node;
    }

    private static Node walk(Node node)
    {
        Random random    = new Random();
        return node.getOut().get(random.nextInt(node.getOut().size()));

    }

}
```