1. Read the input file data into a python variable called **reader**.
2. Note that the input file will be in a matrix format having Web Nodes as the column headers.
3. Read each line of the variable to a list variable using step 4.
4. **for line belongs to reader { data.append(line) }**
5. Convert the **data** list variable to array using **data=Numpy.array(data)**
6. Numpy is a library that has all array functionalities
7. dup=data[slice the first row of array using slicing options].copy (creates a copy of first row of the array)
8. no_of_nodes=dup.size (returns the size of row which will be used as exit criteria for a loop which is used to store array index values in a separate list variable)
9. **v=0;li=list() ; while v<no_of_nodes { *li.append(v) ; v=v+1;* }**
10. Step 9 will assign index values in a separate list since python **for** loops are designed in such a way that it takes each values from a list during each iteration.
11. **for i belongs to li { *initial_rank.append(float((1.0/no_of_nodes))) ; out.append(0); sub_rank.append(0); rank.append(0)* }**
12. Convert all the above List variables initial_rank, out, sub_rank to array.
13. **for i belongs to li { for j belongs to li { if data[i,j] { out[j]=out[j]+1;} } }**
14. Step 13 calculates the number of out-going nodes for each Web Node.
15. Now since we have Initial Rank values and out- going nodes we can calculate the Rank for each Node in the 1st Iteration. Using the formula
    Rank[Node1]=initial_rank[Node1]/no_of_nodes + (1-initial_rank[Node1]) * (initial_rank[NodeX1 that is incoming to Node1]/No of outgoing Nodes for NodeX1+...
16. Below is the loop used for 1st Iteration alone
17. **for i belongs to li {for j belongs to li { if data[i,j] is not Empty { sub_rank[i]=initial_rank[j]/out[j] +sub_rank[i] ;} } }**
18. Step 17 calculates the sub_rank for Node i. Once we have the sub_rank for Node i we calculate Rank for Node i using below steps.
19. **for i belongs to li { rank[i]=initial_rank[i]/no_of_nodes+(1-initial_rank[i])*(sub_rank[i]) ; print "Node %s= %s" % ((i+1),rank[i]); }**
20. For rest of the Iterations repeat the steps 17 and 19 after performing step 21.
21. **for i belongs to li { initial_rank[i]=rank[i] ; sub_rank[i]=0 ;}**
22. We perform step 21 since we have to use the Ranks of previous iteration for current iteration and also we have to again assign **sub_ranks to 0** since it is a temporary variable.
23. Once we store the Ranks of all the Web Nodes in Rank Array, we have to sort the Index of the Rank Array based on the values of the Array using **argsort** function under **array** class and then display the Ranks of each Nodes by step 24 and 25
24. **nodes=rank.argsort() ; i=0; j=1;**
25. **while i<no_of_nodes { print "Rank-%s= Node-%s" %(i+1,nodes[no_of_nodes-j]+1); i=i+1 ; j=j+1; }**
26. The print statement in step 25 will have output as step 27.
27. **Rank-1= Node-2**
    Rank-2= Node-4
    Rank-3= Node-1
    Rank-4= Node-5
    Rank-5= Node-3