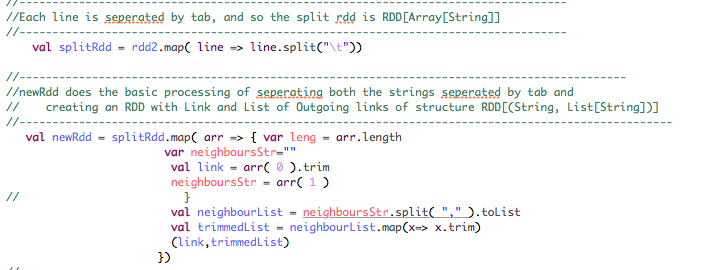
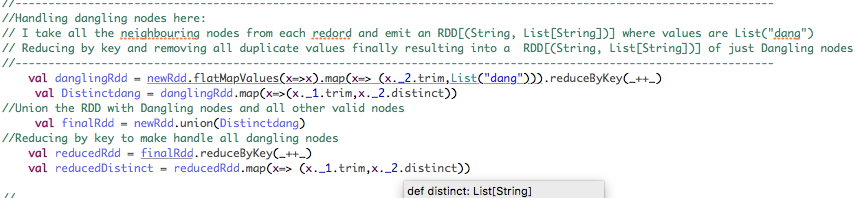
Report: HW4:

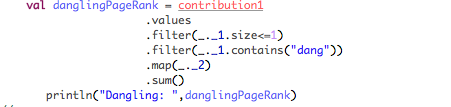
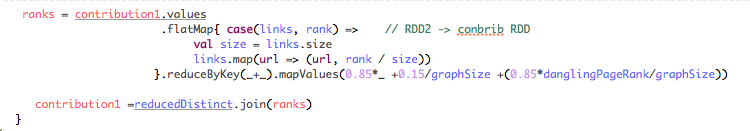
Name : Chandrika Sharma

Github <https://github.ccs.neu.edu/chandrika2311/MapReduce-CS-6240>

Structure of my program:

1. Initially the BZ2 parser returns a string to the Scala program.
2. This string is the link separated by the out links by ‘\t’
3. I transform this into an rdd of type: RDD[(String, List[String])]
4. Next is handling dangling nodes :I mapped all the outlinks with a list(“dang”)



1. Then I did the union of both the Rdds from point 3 and 4 and reduced by key.
2. Next was calculation of graph size and page rank mass of dangling dum.
3. 
4. Dangling sum of Pang ranks calculated by checking if the size of out links list was 1 or smaller, then checking if the list contained “dang” and then summing the page rank values of all such occurrences.
5. Finally: Page rank calculation
6. 
7. Here I took the size of each out link list and calculated the new pagerank, flat mapped each of the urls to that value and reduced it again to map new pageranks with dangling sum with it.
8. My program for Hadoop Map reduce Page rank implementation took 45 minutes for the 10 machines.
9. On spark however the program has taken 1 hour 50 minutes
10. My program for Hadoop Map reduce Page rank implementation took 1 hour for the 5 machines.
11. On spark however the program has taken 3 hour

This could be because of how I have written my program and needs improvement.