Table of Contents

In [ ]:

In [8]:
```
%%javascript
/*********************************************************************
*************
Known Mathjax Issue with Chrome - a rounding issue adds a border to the right of
mathjax markup
https://github.com/mathjax/MathJax/issues/1300
A quick hack to fix this based on stackoverflow discussions:
http://stackoverflow.com/questions/34277967/chrome-rendering-mathjax-equations-wi
th-a-trailing-vertical-line
*********************************************************************
*************/

$('.math>span').css("border-left-color","transparent")
```

In [1]:
```
%reload_ext autoreload
%autoreload 2
```

MIDS - w261 Machine Learning At Scale

Course Lead: Dr James G. Shanahan (**email** Jimi via James.Shanahan *AT* gmail.com)

Assignment - HW9

Name: Nilesh Bhoyar
Class: MIDS w261 (Section *Your Section Goes Here*, e.g., Fall 2016 Group 1)
Email: nilesh.bhoyar@iSchool.Berkeley.edu
StudentId 26302327 **End of StudentId**
Week: 9

NOTE: please replace `1234567` with your student id above
Due Time: HW is due the Tuesday of the following week by 8AM (West coast time). I.e., Tuesday, Mar 21, 2017 in the case of this homework.

Instructions

MIDS UC Berkeley, Machine Learning at Scale
DATSCIW261 ASSIGNMENT #9

Version 2017-3-16

IMPORTANT

Parts of this homework can be completed locally on your computer. For questions involving the wikipedia dataset, you will need to run your code in the cloud.

=== INSTRUCTIONS for SUBMISSIONS ===

Follow the instructions for submissions carefully.

Each student has a `HW-<user>` repository for all assignments.

Click this link to enable you to create a github repo within the MIDS261 Classroom:
https://classroom.github.com/assignment-invitations/3b1d6c8e58351209f9dd865537111ff8 (https://classroom.github.com/assignment-invitations/3b1d6c8e58351209f9dd865537111ff8)
and follow the instructions to create a HW repo.

Push the following to your HW github repo into the master branch:

- Your local HW6 directory. Your repo file structure should look like this:

```
HW-<user>
    --HW3
       |__MIDS-W261-HW-03-<Student_id>.ipynb
       |__MIDS-W261-HW-03-<Student_id>.pdf
       |__some other hw3 file
    --HW4
       |__MIDS-W261-HW-04-<Student_id>.ipynb
       |__MIDS-W261-HW-04-<Student_id>.pdf
       |__some other hw4 file
    etc..
```

HW Problems

HW 9 Dataset

Note that all referenced files life in the enclosing directory. Checkout the Data subdirectory on Dropbox (https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAAKsjQfF9uHfv-X9mCqr9wa?dl=0) or the AWS S3 buckets (details contained each question).

3. HW9.0 Short answer questions

Back to Table of Contents

What is PageRank and what is it used for in the context of web search?

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites. PageRank - a link analysis algorithm assigns a numerical weighting to each element of a hyperlinked World Wide Web documents, with the purpose of "measuring" its relative importance within the set.As such based on query content we can retrieve the important web pages for particular search.

What modifications have to be made to the webgraph in order to leverage the machinery of Markov Chains to compute the Steady State Distibution?

transition matrix should be aperiodic,irreducible and stochastuc for markov chains to achieve steady state distribution.

To satisfy above conditions , we first distribute the dangling mass ie. mass corresponding to nodes with not outlinks (this makes web graph irreducible), link from each page to every page and give each link a small transition probability controlled by a parameter d (web graph becomes aperiodic) and the transform final page rank equation to make it stochastic.

Reference http://www.dsi.unive.it/~calpar/New_HPC_course/AA12-13/project12-13.pdf (http://www.dsi.unive.it/~calpar/New_HPC_course/AA12-13/project12-13.pdf)

OPTIONAL: In topic-specific pagerank, how can we ensure that the irreducible property is satifsied? (HINT: see HW9.4)

Define new transition matrix where we add transition edges between every pair of node in G with probability alpha/N.

HW9.1 MRJob implementation of basic PageRank

$$P(n) = \alpha \left(\frac{1}{|G|} \right) + (1 - \alpha) \sum_{m \in L(n)} \frac{P(m)}{C(m)}$$

where $|G|$ is the total number of nodes (pages) in the graph, α is the random jump factor. $L(n)$ is the set of pages that link to n, and $C(m)$ is the out degree of the node m (the number of links on page m). The random jump factor α is sometimes called the "teleportation" factor; alternatively, (1 – α) is referred to as the "damping" factor

---

Write a basic MRJob implementation of the iterative PageRank algorithm that takes sparse adjacency lists as input (as explored in HW 7).

Make sure that you implementation utilizes teleportation (1-damping/the number of nodes in the network), and further, distributes the mass of dangling nodes with each iteration so that the output of each iteration is correctly normalized (sums to 1).

[NOTE: The PageRank algorithm assumes that a random surfer (walker), starting from a random web page, chooses the next page to which it will move by clicking at random, with probability d,one of the hyperlinks in the current page. This probability is represented by a so-called *damping factor* d, where d ∈ (0, 1). Otherwise, with probability (1 – d), the surfer jumps ("teleports") to any web page in the network. If a page is a dangling end, meaning it has no outgoing hyperlinks, the random surfer selects an arbitrary web page from a uniform distribution and "teleports" to that page]

As you build your code, use the data located here :

In the Data Subfolder for HW7 on Dropbox (same dataset as HW7) with the same file name.

> Dropbox: https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAAKsjQfF9uHfv-X9mCqr9wa?dl=0 (https://www.dropbox.com/sh/2c0k5adwz36lkcw/AAAAKsjQfF9uHfv-X9mCqr9wa?dl=0)

Or on Amazon:

> s3://ucb-mids-mls-networks/PageRank-test.txt

with teleportation factor alpha set to 0.15 (1-d, where d, the damping factor is set to 0.85), and crosscheck your work with the true result, displayed in the first image in the Wikipedia article (https://en.wikipedia.org/wiki/PageRank) and here for reference are the corresponding PageRank probabilities:

```
A, 0.033
B, 0.384
C, 0.343
D, 0.039
E, 0.081
F, 0.039
G, 0.016
H, 0.016
I, 0.016
J, 0.016
K, 0.016
```

In [20]:
```
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [21]:
```
!rm PageRank-test.txt

!wget https://www.dropbox.com/sh/2c0k5adwz36lkcw/AADxzBgNxNF5Q6-eanjnK64qa/PageRank-test.txt
```

```
--2017-07-17 02:52:34--  https://www.dropbox.com/sh/2c0k5adwz36lkcw/AADxzBgNxNF5Q6-eanjnK64qa/PageRank-test.txt
Resolving www.dropbox.com... 162.125.6.1
Connecting to www.dropbox.com|162.125.6.1|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://dl.dropboxusercontent.com/content_link/85ART6CmFjH344mtBc0g3JhyUBWr5lqObtjBxVYJb40BeGQRWDAKX1unNq8UOQRL/file [following]
--2017-07-17 02:52:35--  https://dl.dropboxusercontent.com/content_link/85ART6CmFjH344mtBc0g3JhyUBWr5lqObtjBxVYJb40BeGQRWDAKX1unNq8UOQRL/file
Resolving dl.dropboxusercontent.com... 162.125.6.6
Connecting to dl.dropboxusercontent.com|162.125.6.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 166 [text/plain]
Saving to: `PageRank-test.txt'

100%[======================================>] 166         --.-K/s   in 0s

2017-07-17 02:52:35 (39.0 MB/s) - `PageRank-test.txt' saved [166/166]
```

In [22]:
```
!hdfs dfs -copyFromLocal PageRank-test.txt hdfs:///user/nileshbhoyar/
```

```
copyFromLocal: `hdfs:/user/nileshbhoyar/PageRank-test.txt': File exists
```

HW 9.1 Implementation

First MRjob

Final Formulae to calculate the rank score is

$$P(n) = \alpha \left(\frac{1}{|G|} \right) + (1 - \alpha) \sum_{m \in L(n)} \frac{P(m)}{C(m)}$$

first MR job calculates the mass distribution for outgoing nodes i.e. component $\sum_{m\in{L(n)}}\frac{P(m)}{C(m)}$

We do this by:-
first initiatlizing probabilities and next iterations we would distribute that mass.

In reducer , we will combine that mass and feed it again to mapper for redistribution.

We also track dangling nodes for teleportation/dangling node mass destribution in next mrjob.

In [23]:
```
%%writefile pagerankalgo.py
#!/usr/bin/python
from mrjob.job import MRJob
from mrjob.step import MRStep

class pagerankalgo(MRJob):
    DEFAULT_PROTOCOL = 'json'

    def configure_options(self):
        super(pagerankalgo, self).configure_options()
        self.add_passthrough_option(
            '--i', dest='init', default='0', type='int',
            help='i: run initialization iteration (default 0)')

    # mapper of first pass of the file (initialization)
    def mapper_job_init(self, _, line):
        # parse line
        nid, adj = line.strip().split('\t', 1)
        nid = nid.strip('"')
        adj=eval(adj)
        # initialize node struct
        node = {'a':adj.keys(), 'p':0}
        rankMass = 1.0/len(adj)
        # emit graphs as it
        yield nid, node

        # emit pageRank mass
        for m in node['a']:
            yield m, rankMass

    # after initialization
    def mapper_job_iter(self, _, line):
        # parse line
        nid, node = line.strip().split('\t', 1)
        nid = nid.strip('"')
        node =eval(node)
        # distribute rank mass
        n_adj = len(node['a'])
        if n_adj > 0:
            rankMass = 1.0*node['p'] / n_adj
            # emit pageRank mass
            for m in node['a']:
                yield m, rankMass
        else:
            # track dangling mass with counter
            self.increment_counter('dangling_mass', 'mass', int(node['p']*1e10))
        # reset pageRank and emit node
        node['p'] = 0
        yield nid, node



    # reducer for initialization pass --> need to handle dangling nodes
    def reducer_job_init(self, nid, value):
        # increase counter for node count
        self.increment_counter('node_count', 'nodes', 1)
        rankMass, node = 0.0, None
        # loop through all arrivals
        for v in value:
            if isinstance(v, float):
                rankMass += v
            else:
                node = v
        # dangling node additions
```

```
Overwriting pagerankalgo.py
```

In [24]:
```
%%writefile distweight.py
#!/usr/bin/python
from mrjob.job import MRJob
from mrjob.step import MRStep

class distweight(MRJob):
    DEFAULT_PROTOCOL = 'json'

    def configure_options(self):
        super(distweight, self).configure_options()
        self.add_passthrough_option(
            '--s', dest='size', default=0, type='int',
            help='size: node number (default 0)')
        self.add_passthrough_option(
            '--j', dest='alpha', default=0.15, type='float',
            help='jump: random jump factor (default 0.15)')
        self.add_passthrough_option(
            '--m', dest='m', default=0, type='float',
            help='m: rank mass from dangling nodes (default 0)')

    def mapper_init(self):
        self.damping = 1 - self.options.alpha
        self.p_dangling = self.options.m / self.options.size



    def mapper(self, _, line):
        # parse line
        nid, node = line.strip().split('\t', 1)
        nid = nid.strip('"')
        node =eval(node)
        node['p'] = (self.p_dangling + node['p']) * self.damping + self.options.a
lpha
        yield nid, node

    def steps(self):

        return [MRStep(mapper_init=self.mapper_init
                       , mapper= self.mapper ,
                    jobconf = {
                       'mapred.map.tasks': 100,
                       'mapred.reduce.tasks':20,
                       'mapreduce.reduce.cpu.vcores':4,
                       }
                     )
                ]

if __name__ == '__main__':
    distweight.run()
```

```
Overwriting distweight.py
```

In [25]:
```
%%writefile topN.py
#!/usr/bin/python
from mrjob.job import MRJob
from mrjob.step import MRStep

class topN(MRJob):


    def configure_options(self):
        super(topN, self).configure_options()
        self.add_passthrough_option(
            '--s', dest='size', default=0, type='int',
            help='size: node number (default 0)')
        self.add_passthrough_option(
            '--n', dest='top', default=100, type='int',
            help='size: node number (default 100)')

    def mapper(self, _, line):
        # parse line
        nid, node = line.strip().split('\t', 1)
        cmd = 'node = %s' %node
        exec cmd
        yield node['p'], nid.strip('"')

    def reducer_init(self):
        self.i = 0
        self.total = 0

    def reducer(self, pageRank, nid):
        for n in nid:
            if self.i < self.options.top:
                self.i += 1
                self.total += pageRank
                yield n, pageRank/self.options.size

    def reducer_final(self):
        yield 'total mass: ', self.total/self.options.size

    def steps(self):
        jc = {
            'mapreduce.job.output.key.comparator.class': 'org.apache.hadoop.mapre
duce.lib.partition.KeyFieldBasedComparator',
            'mapreduce.partition.keycomparator.options': '-k1,1nr',
            'mapreduce.job.maps': '2',
            'mapreduce.job.reduces': '1',
        }
        return [MRStep(mapper=self.mapper, reducer_init=self.reducer_init,
                       reducer=self.reducer, reducer_final=self.reducer_final
                       , jobconf = jc
                      )
                ]

if __name__ == '__main__':
    topN.run()
```

```
Overwriting topN.py
```

In [13]:
```
%%writefile driver.py
#!/usr/bin/python

from pagerankalgo import pagerankalgo
from distweight import distweight
from topN import topN
from subprocess import call, check_output
from time import time
import sys, getopt, datetime, os

# parse parameter
if __name__ == "__main__":

    try:
        opts, args = getopt.getopt(sys.argv[1:], "hg:j:i:d:s:")
    except getopt.GetoptError:
        print "Issues"
        sys.exit(2)
    if len(opts) != 4:

        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':

            sys.exit(2)
        elif opt == '-g':
            graph = arg
        elif opt == '-j':
            jump = arg
        elif opt == '-i':
            n_iter = arg

        elif opt == '-s':
            n_node = arg

start = time()
FNULL = open(os.devnull, 'w')
n_iter = int(n_iter)
doInit = n_node=='0'


print "Initialization ..."
if doInit:
    # clear directory
    print str(datetime.datetime.now()) + ': clearing directory ...'
    call(['hdfs', 'dfs', '-rm', '-r', '/user/nileshbhoyar/in'], stdout=FNULL)
    call(['hdfs', 'dfs', '-rm', '-r', '/user/nileshbhoyar/out'], stdout=FNULL)

    # creat initialization job
    init_job = pagerankalgo(args=[graph, '--i', '1', '-r', 'hadoop', '--output-di
r', 'hdfs:///user/nileshbhoyar/out'])

    # run initialization job
    print str(datetime.datetime.now()) + ': running iteration 1 ...'
    with init_job.make_runner() as runner:
        runner.run()

    # checking counters
    n_node = runner.counters()[0]['node_count']['nodes']
    n_dangling = runner.counters()[0]['dangling_mass']['mass']/1e10

    print '%s: initialization complete: %d nodes, %d are dangling!' %(str(datetim
e.datetime.now()), n_node, n_dangling)
```

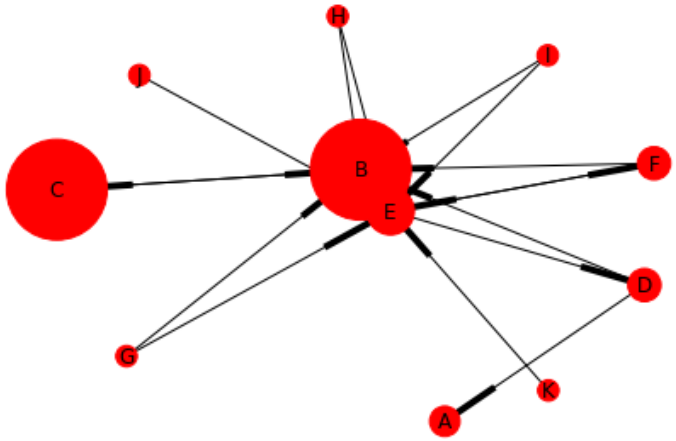
```
Overwriting driver.py
```

In [14]:
```
from matplotlib import pyplot as plt
import networkx as nx

def draw_graph(nSize=3500):
    # define the graph from adjacency matrix
    G = nx.DiGraph()
    with open('PageRank-test.txt') as f:
        for node in f.readlines():
            source, adj = node.strip().split('\t')
            adj =eval(adj)
            for d in adj:
                G.add_edge(source, d)
            G.node[source]['state'] = source
    G.add_node('A')
    G.node['A']['state'] = 'A'

    # define node size
    ranks = {}
    with open('pagerank_out.txt') as f:
        for line in f.readlines():
            nid, rank = line.strip().split('\t')
            nid = nid.strip('"')
            if len(nid) == 1:
                ranks[nid] = float(rank)
    norm = max(ranks.values())
    size =[ranks[n]*nSize/norm for n in G.nodes()]

    # draw the graph
    pos = nx.spring_layout(G)
    nx.draw(G, pos, node_size = size)
    node_labels = nx.get_node_attributes(G,'state')
    nx.draw_networkx_labels(G, pos, labels = node_labels)
    plt.show()
```

In [5]:
```
def draw_patterns(damping = 0.15):
    !python driver.py -g "hdfs:/user/nileshbhoyar/PageRank-test.txt" -j $damping
-i 10  -s '0'
    !hdfs dfs -cat hdfs:/user/nileshbhoyar/finalranks/* > pagerank_out.txt
    draw_graph()
```

In [ ]:
```
draw_patterns(0.15)
```

```
Initialization ...
2017-07-16 19:32:33.840169: clearing directory ...
17/07/16 19:32:35 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
rm: `/user/nileshbhoyar/out': No such file or directory
2017-07-16 19:32:38.500804: running iteration 1 ...
2017-07-16 19:33:26.046493: initialization complete: 11 nodes, 1 are dangling!
2017-07-16 19:33:28.439259: distributing loss mass ...
17/07/16 19:34:12 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:34:15.577003: running iteration 2 ...
17/07/16 19:35:08 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:35:11.485448: distributing loss mass 0.6523 ...
17/07/16 19:35:55 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:35:58.767484: running iteration 3 ...
17/07/16 19:37:08 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:37:11.500028: distributing loss mass 0.4174 ...
17/07/16 19:37:54 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:37:57.117315: running iteration 4 ...
17/07/16 19:38:46 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:38:49.556046: distributing loss mass 0.7042 ...
17/07/16 19:39:33 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:39:36.842429: running iteration 5 ...
17/07/16 19:40:24 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:40:27.589321: distributing loss mass 0.4136 ...
17/07/16 19:41:08 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:41:11.880274: running iteration 6 ...
17/07/16 19:42:02 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:42:05.595837: distributing loss mass 0.4254 ...
17/07/16 19:42:49 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:42:52.225275: running iteration 7 ...
17/07/16 19:43:45 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:43:48.415772: distributing loss mass 0.3753 ...
17/07/16 19:44:32 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:44:35.711877: running iteration 8 ...
17/07/16 19:46:59 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:47:02.784374: distributing loss mass 0.3812 ...
17/07/16 19:47:49 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:47:52.704482: running iteration 9 ...
17/07/16 19:48:43 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:48:46.731869: distributing loss mass 0.3659 ...
17/07/16 19:49:28 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:49:31.107580: running iteration 10 ...
17/07/16 19:50:22 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 19:50:25.220259: distributing loss mass 0.3660 ...
Sorting and Top N job
17/07/16 19:51:05 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
```



HW 9.1 Analysis

HW9.2: Exploring PageRank teleportation and network plots

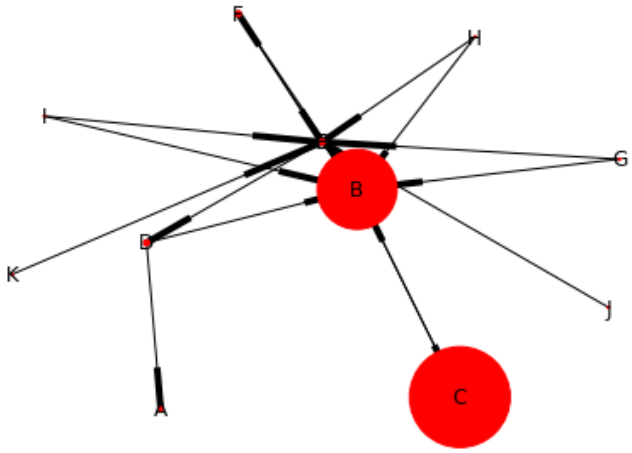
- In order to overcome problems such as disconnected components, the damping factor (a typical value for d is 0.85) can be varied.
- Using the graph in HW9.1, plot the test graph (using networkx, https://networkx.github.io/ (https://networkx.github.io/)) for several values of the damping factor, so that each nodes radius is proportional to its PageRank score.
- In particular you should do this for the following damping factors: [0,0.25,0.5,0.75, 0.85, 1].
- Note your plots should look like the following: https://en.wikipedia.org/wiki/PageRank#/media/File:PageRanks-Example.svg (https://en.wikipedia.org/wiki/PageRank#/media/File:PageRanks-Example.svg)

HW 9.2 Implementation

In [6]:
```
# START STUDENT CODE 9.2
# (ADD CELLS AS NEEDED)
draw_patterns(0)
# END STUDENT CODE 9.2
```

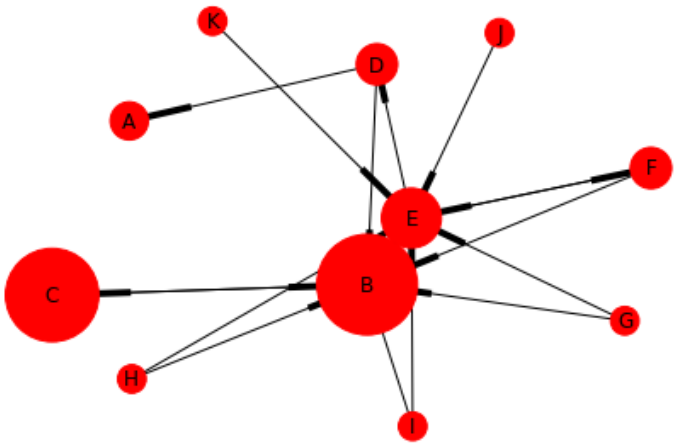
```
Initialization ...
2017-07-16 20:41:24.749237: clearing directory ...
17/07/16 20:41:26 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
17/07/16 20:41:29 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:41:29.794987: running iteration 1 ...
2017-07-16 20:42:16.068356: initialization complete: 11 nodes, 1 are dangling!
2017-07-16 20:42:18.534373: distributing loss mass ...
17/07/16 20:43:00 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:43:03.084750: running iteration 2 ...
17/07/16 20:45:21 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:45:24.676410: distributing loss mass 0.5909 ...
17/07/16 20:46:19 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:46:22.756850: running iteration 3 ...
17/07/16 20:47:21 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:47:24.352980: distributing loss mass 0.2658 ...
17/07/16 20:48:07 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:48:10.106146: running iteration 4 ...
17/07/16 20:49:01 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:49:04.047490: distributing loss mass 0.7328 ...
17/07/16 20:49:45 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:49:48.887972: running iteration 5 ...
17/07/16 20:50:38 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:50:41.298706: distributing loss mass 0.1760 ...
17/07/16 20:51:22 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:51:25.503848: running iteration 6 ...
17/07/16 20:52:14 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:52:17.790916: distributing loss mass 0.2028 ...
17/07/16 20:53:01 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:53:04.461214: running iteration 7 ...
17/07/16 20:53:55 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:53:58.953461: distributing loss mass 0.0699 ...
17/07/16 20:54:40 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:54:43.468554: running iteration 8 ...
17/07/16 20:55:33 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:55:36.719451: distributing loss mass 0.0882 ...
17/07/16 20:56:17 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:56:20.544636: running iteration 9 ...
17/07/16 20:57:11 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:57:14.434505: distributing loss mass 0.0322 ...
17/07/16 20:57:58 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:58:01.900882: running iteration 10 ...
17/07/16 20:58:51 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 20:58:53.901752: distributing loss mass 0.0324 ...
Sorting and Top N job
17/07/16 20:59:43 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
```

```
/home/nileshbhoyar/.conda/envs/py27/lib/python2.7/site-packages/networkx/drawing
/nx_pylab.py:126: MatplotlibDeprecationWarning: pyplot.hold is deprecated.
    Future behavior will be consistent with the long-time default:
    plot commands add elements without first clearing the
    Axes and/or Figure.
  b = plt.ishold()
/home/nileshbhoyar/.conda/envs/py27/lib/python2.7/site-packages/networkx/drawing
/nx_pylab.py:138: MatplotlibDeprecationWarning: pyplot.hold is deprecated.
    Future behavior will be consistent with the long-time default:
    plot commands add elements without first clearing the
    Axes and/or Figure.
  plt.hold(b)
/home/nileshbhoyar/.conda/envs/py27/lib/python2.7/site-packages/matplotlib/__ini
t__.py:917: UserWarning: axes.hold is deprecated. Please remove it from your mat
plotlibrc and/or style files.
  warnings.warn(self.msg_depr_set % key)
/home/nileshbhoyar/.conda/envs/py27/lib/python2.7/site-packages/matplotlib/rcset
up.py:152: UserWarning: axes.hold is deprecated, will be removed in 3.0
  warnings.warn("axes.hold is deprecated, will be removed in 3.0")
```



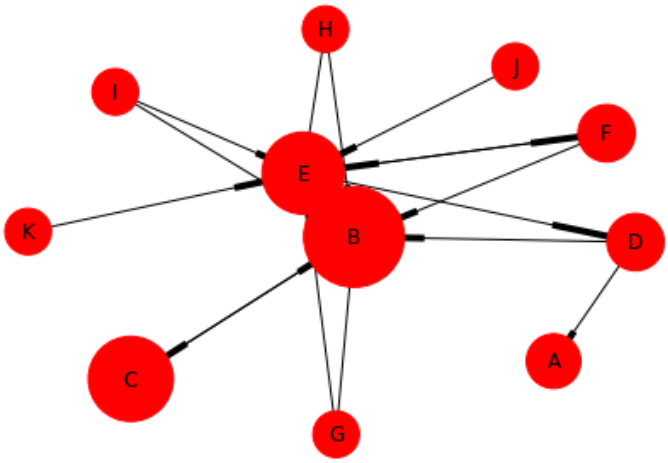
In [7]:
```
draw_patterns(0.25)
```

```
Initialization ...
2017-07-16 21:00:45.456629: clearing directory ...
17/07/16 21:00:47 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
17/07/16 21:00:49 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:00:50.268345: running iteration 1 ...
2017-07-16 21:01:38.877520: initialization complete: 11 nodes, 1 are dangling!
2017-07-16 21:01:41.319608: distributing loss mass ...
17/07/16 21:02:26 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:02:29.675613: running iteration 2 ...
17/07/16 21:03:20 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:03:23.185741: distributing loss mass 0.6932 ...
17/07/16 21:04:46 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:04:49.841042: running iteration 3 ...
17/07/16 21:05:39 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:05:42.939724: distributing loss mass 0.5103 ...
17/07/16 21:06:24 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:06:27.439762: running iteration 4 ...
17/07/16 21:07:14 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:07:17.921072: distributing loss mass 0.7073 ...
17/07/16 21:08:01 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:08:04.056388: running iteration 5 ...
17/07/16 21:08:54 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:08:57.802956: distributing loss mass 0.5312 ...
17/07/16 21:11:13 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:11:16.055932: running iteration 6 ...
17/07/16 21:12:14 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:12:17.222377: distributing loss mass 0.5375 ...
17/07/16 21:13:00 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:13:03.967241: running iteration 7 ...
17/07/16 21:13:54 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:13:57.515710: distributing loss mass 0.5139 ...
17/07/16 21:14:41 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:14:44.906394: running iteration 8 ...
17/07/16 21:15:36 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:15:39.438717: distributing loss mass 0.5163 ...
17/07/16 21:16:26 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:16:29.440823: running iteration 9 ...
17/07/16 21:17:23 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:17:26.131938: distributing loss mass 0.5107 ...
17/07/16 21:18:07 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:18:10.994422: running iteration 10 ...
17/07/16 21:19:06 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-16 21:19:09.125240: distributing loss mass 0.5107 ...
Sorting and Top N job
17/07/16 21:19:52 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
```


K
J
D
A
F
E
C
B
G
H
I

In [15]:
```
draw_patterns(0.5)
```

```
Initialization ...
2017-07-17 02:12:02.168046: clearing directory ...
17/07/17 02:12:04 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
17/07/17 02:12:06 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:12:07.042685: running iteration 1 ...
2017-07-17 02:13:36.847833: initialization complete: 11 nodes, 1 are dangling!
2017-07-17 02:13:39.219183: distributing loss mass ...
17/07/17 02:14:27 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:14:30.461836: running iteration 2 ...
17/07/17 02:15:53 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:15:56.198916: distributing loss mass 0.7955 ...
17/07/17 02:16:40 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:16:43.405766: running iteration 3 ...
17/07/17 02:17:39 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:17:42.499118: distributing loss mass 0.7142 ...
17/07/17 02:18:25 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:18:28.419988: running iteration 4 ...
17/07/17 02:19:17 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:19:20.720794: distributing loss mass 0.7726 ...
17/07/17 02:20:19 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:20:23.134170: running iteration 5 ...
17/07/17 02:21:23 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:21:26.433516: distributing loss mass 0.7378 ...
17/07/17 02:22:07 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:22:10.718445: running iteration 6 ...
17/07/17 02:22:58 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:23:01.768981: distributing loss mass 0.7386 ...
17/07/17 02:23:43 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:23:46.617112: running iteration 7 ...
17/07/17 02:24:36 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:24:39.191413: distributing loss mass 0.7365 ...
17/07/17 02:25:24 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:25:27.633773: running iteration 8 ...
17/07/17 02:26:29 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:26:32.479631: distributing loss mass 0.7367 ...
17/07/17 02:28:28 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:28:31.381934: running iteration 9 ...
17/07/17 02:29:40 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:29:43.468654: distributing loss mass 0.7364 ...
17/07/17 02:31:21 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:31:23.961569: running iteration 10 ...
17/07/17 02:32:43 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:32:46.553901: distributing loss mass 0.7364 ...
Sorting and Top N job
17/07/17 02:33:36 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
```



In [ ]:
```
draw_patterns(0.75)
```

In [ ]:
```
draw_patterns(0.85)
```

In [ ]:
```
draw_patterns(1.0)
```

HW 9.2 Analysis

HW9.3: Applying PageRank to the Wikipedia hyperlinks network

- Run your PageRank implementation on the Wikipedia dataset for 5 iterations, and display the top 100 ranked nodes (with damping factor = 0.85).
- Run your PageRank implementation on the Wikipedia dataset for 10 iterations, and display the top 100 ranked nodes (with teleportation (random jump) factor alpha of 0.15).
- Have the top 100 ranked pages changed? Comment on your findings.
- Plot the pagerank values for the top 100 pages resulting from the 5 iterations run. Then plot the pagerank values for the same 100 pages that resulted from the 10 iterations run.

HW 9.3 Implementation

Iterations 5 and Damping Factor 0.85

In [33]:
```
# START STUDENT CODE 9.3
# (ADD CELLS AS NEEDED)
!python driver.py -g "hdfs:/user/andrewlam/HW7/Data/wikipedia/all-pages-indexed-o
ut.txt" -j 0.15 -i 5  -s '0'
# END STUDENT CODE 9.3
```

```
Initialization ...
2017-07-18 03:08:00.604498: clearing directory ...
17/07/18 03:08:02 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
17/07/18 03:08:05 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 03:08:05.698878: running iteration 1 ...
No handlers could be found for logger "mrjob.compat"
2017-07-18 03:16:07.856566: initialization complete: 15192277 nodes, 9410987 are
dangling!
2017-07-18 03:16:10.400427: distributing loss mass ...
17/07/18 03:18:42 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 03:18:45.535910: running iteration 2 ...
17/07/18 03:30:12 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 03:30:16.055488: distributing loss mass 7608969.0125 ...
17/07/18 03:32:44 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 03:32:47.922636: running iteration 3 ...
17/07/18 03:41:30 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 03:41:33.869676: distributing loss mass 7103036.3030 ...
17/07/18 03:44:45 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 03:44:48.825008: running iteration 4 ...
17/07/18 04:04:36 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 04:04:39.947955: distributing loss mass 6940792.1440 ...
17/07/18 04:07:16 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 04:07:19.225942: running iteration 5 ...
17/07/18 04:16:43 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-18 04:16:46.385991: distributing loss mass 6884560.5221 ...
Sorting and Top N job
17/07/18 04:19:05 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
```

In [34]:
```
!hdfs dfs -cat hdfs:/user/nileshbhoyar/finalranks/* > pagerank_out_wiki5.txt
!cat pagerank_out_wiki5.txt
```

```
"13455888"      0.0014607248567696442
"1184351"       0.0006828280964932274 9
"4695850"       0.0006429807438275631 6
"5051368"       0.0005763793635101584 1
"1384888"       0.0004600329621555220 1
"7902219"       0.0004581331433058471 7
"6113490"       0.0004547314775405817 8
"2437837"       0.0004456250824910084 4
"6076759"       0.0004304041961846769 1
"13425865"      0.0004286518235660105 8
"4196067"       0.0004220348706988058 1
"6172466"       0.0004085202123451275 4
"14112583"      0.0003820633588030587 5
"10390714"      0.0003685343877331824 8
"15164193"      0.0003501337004836806 4
"3191491"       0.0003449496698251880 9
"7835160"       0.0003320829589600102 9
"6416278"       0.000331090957683057 4
"6237129"       0.0003280171616363163 7
"1516699"       0.0003248582134636595 4
"13725487"      0.0003211215467495405 6
"7576704"       0.0003144368431234253 7
"9276255"       0.0003126595224028241 4
"10469541"      0.0003110539659148065 7
"5154210"       0.0003052921596553504 4
"7990491"       0.0002798165075535850 5
"12836211"      0.0002772498249820734 1
"4198751"       0.0002676499953048684 3
"2797855"       0.000263014598992119 9
"11253108"      0.0002612684619553773 7
"3603527"       0.0002600149112462512 3
"3069099"       0.000253510659489969 5
"9386580"       0.0002533372828302826 5
"12074312"      0.000251071446662086 5
"14881689"      0.0002504393686362062 3
"2155467"       0.0002476649473588848 9
"1441065"       0.000240663056559329 9
"14503460"      0.0002349039914579231 2
"3191268"       0.0002208673132885714 6
"10566120"      0.0002204521258441458 6
"2396749"       0.0002201284587530667 7
"11147327"      0.0002149031255632435 2
"2614581"       0.0002148857467834241 9
"1637982"       0.0002124440452298093 7
"11245362"      0.0002070061121115468 3
"12430985"      0.0002043623786679339 8
"9355455"       0.0001963702114635725 8
"10527224"      0.000193277533739235 1
"6172167"       0.0001915416880695603 5
"2614578"       0.0001914614617880310 9
"981395"        0.0001898713696193868 9
"14112408"      0.0001886270178214922 2
"8697871"       0.0001885162227969237 4
"9391762"       0.0001860544665479014 7
"6171937"       0.0001828818140932501 1
"5490435"       0.0001801085405926872 4
"14725161"      0.0001734423295085281 3
"11582765"      0.0001719257889605150 7
"9562547"       0.0001688263436125669 9
"994890"        0.0001677306385066480 8
"12067030"      0.0001655683034951175 4
"10345830"      0.0001634053477278892 7
"9394907"       0.0001606955245647929 8
"13280859"      0.0001596068818144968 5
"9997298"       0.0001589917267962255 9
```

Iterations 10 and Damping Factor 0.85

In [27]:
```
!python driver.py -g "hdfs:/user/andrewlam/HW7/Data/wikipedia/all-pages-indexed-o
ut.txt" -j 0.15 -i 10 -s '0'
```

```
Initialization ...
2017-07-17 02:53:08.642698: clearing directory ...
rm: `/user/nileshbhoyar/in': No such file or directory
17/07/17 02:53:12 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 02:53:13.350997: running iteration 1 ...
No handlers could be found for logger "mrjob.compat"
2017-07-17 03:09:28.335993: initialization complete: 15192277 nodes, 9410987 are
dangling!
2017-07-17 03:09:30.744802: distributing loss mass ...
17/07/17 03:12:01 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 03:12:04.361281: running iteration 2 ...
17/07/17 03:25:46 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 03:25:49.466782: distributing loss mass 7608969.0125 ...
17/07/17 03:28:33 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 03:28:36.846781: running iteration 3 ...
17/07/17 03:42:29 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 03:42:32.567278: distributing loss mass 7103036.3030 ...
17/07/17 03:46:31 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 03:46:34.694170: running iteration 4 ...
17/07/17 03:57:40 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 03:57:43.803284: distributing loss mass 6940792.1440 ...
17/07/17 03:59:58 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:00:01.486717: running iteration 5 ...
17/07/17 04:08:20 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:08:23.584747: distributing loss mass 6884560.5221 ...
17/07/17 04:10:41 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:10:44.360783: running iteration 6 ...
17/07/17 04:19:06 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:19:09.706080: distributing loss mass 6863177.9606 ...
17/07/17 04:21:22 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:21:25.841813: running iteration 7 ...
17/07/17 04:30:05 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:30:08.523374: distributing loss mass 6854533.8818 ...
17/07/17 04:32:19 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:32:22.787733: running iteration 8 ...
17/07/17 04:40:50 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:40:53.234664: distributing loss mass 6850834.1074 ...
17/07/17 04:43:10 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:43:12.995283: running iteration 9 ...
17/07/17 04:56:45 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:56:48.498767: distributing loss mass 6849174.4042 ...
17/07/17 04:59:08 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 04:59:11.121779: running iteration 10 ...
17/07/17 05:07:39 INFO fs.TrashPolicyDefault: Namenode trash configuration: Dele
tion interval = 5760 minutes, Emptier interval = 360 minutes.
2017-07-17 05:07:42.407385: distributing loss mass 6848394.4797 ...
Sorting and Top N job
```

In [29]:
```
!hdfs dfs -cat hdfs:/user/nileshbhoyar/finalranks/* > pagerank_out_wiki10.txt
!cat pagerank_out_wiki10.txt
```

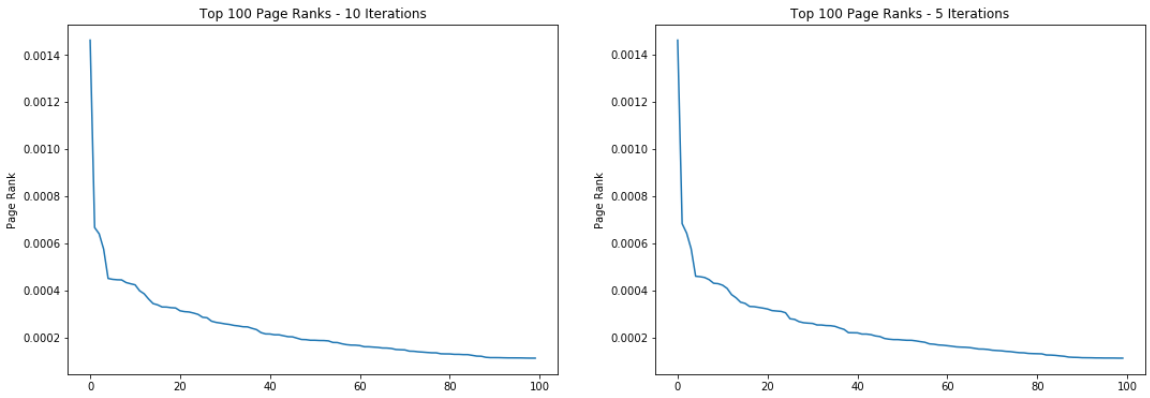
"13455888"	0.0014614491942438257
"1184351"	0.00066633173757986322
"4695850"	0.00063980518755069298
"5051368"	0.00057485388262867973
"1384888"	0.00045030471427850603
"2437837"	0.00044660099098875743
"6113490"	0.00044481824801658734
"7902219"	0.00044420373459828579
"13425865"	0.00043299524492651991
"6076759"	0.00042788533613083288
"4196067"	0.0004232759549062523
"6172466"	0.00039817029586923909
"14112583"	0.00038543367080922113
"10390714"	0.00036316706604628342
"15164193"	0.00034383110763123347
"3191491"	0.00033834786521938883
"6416278"	0.00032935246265791891
"6237129"	0.00032896996556416072
"7835160"	0.00032632071986421647
"1516699"	0.00032507588815577752
"13725487"	0.00031314344180306071
"9276255"	0.00030959412424563733
"7576704"	0.00030809546897992747
"10469541"	0.00030354256660877261
"5154210"	0.00029795335221032872
"12836211"	0.0002857902942672035
"7990491"	0.00028347554322906253
"4198751"	0.00026906211182606
"2797855"	0.00026401327504910974
"11253108"	0.00026106565574639988
"9386580"	0.00025755868482122058
"3603527"	0.0002550899370909592
"12074312"	0.00025104301138297659
"3069099"	0.00024879018334782112
"14881689"	0.00024545732885376207
"2155467"	0.00024484903184801446
"1441065"	0.00023872444275286914
"14503460"	0.00023335074718638856
"2396749"	0.00022060503331684984
"3191268"	0.00021509725578691455
"10566120"	0.00021468682890528893
"2614581"	0.0002113790965621929
"11147327"	0.00021132415993817689
"1637982"	0.0002071596350422885
"12430985"	0.00020338117266916012
"11245362"	0.00020262323395013983
"9355455"	0.00019701920174092094
"10527224"	0.00019142274072842333
"14112408"	0.00019074389254287479
"2614578"	0.00018818343629020061
"9391762"	0.00018809311929740352
"8697871"	0.0001871031700113492
"6172167"	0.00018685330248519746
"981395"	0.0001854013848557578
"6171937"	0.0001788500153761579
"5490435"	0.00017834740118143603
"11582765"	0.0001732578678348902
"14725161"	0.0001695498115129693
"12067030"	0.00016767695230721744
"9562547"	0.00016731685713821943
"994890"	0.00016548126934061452
"9997298"	0.00016067308336923595
"9394907"	0.00016052821854839472
"13280859"	0.00015904269685182109
"10345830"	0.00015776886151301641

In [35]:
```
def hasNumbers(inputString):
  return any(char.isdigit() for char in inputString)
```

In [41]:
```
#Load Page Ranks
pr1 = []
pr2 = []
with open('pagerank_out_wiki10.txt') as f:
    for l in f:
        t  = l.strip().split('\t')
        if hasNumbers(t[0]):
            pr1.append((t[0], t[1]))

with open('pagerank_out_wiki5.txt') as f:
    for l in f:
         t  = l.strip().split('\t')
         if hasNumbers(t[0]):
            pr2.append((t[0], t[1]))
```

In [43]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

plt.figure(figsize=(18,6))
plt.subplot(121)
plt.title("Top 100 Page Ranks - 10 Iterations")
plt.ylabel('Page Rank')
plt.plot([float(pr[1])for pr in pr1])

plt.subplot(122)
plt.title("Top 100 Page Ranks - 5 Iterations")
plt.ylabel('Page Rank')
plt.plot([float(pr[1]) for pr in pr2])
```

Out[43]: [<matplotlib.lines.Line2D at 0x7f9216250750>]



HW 9.3 Analysis

HW9.4: Topic-specific PageRank implementation using MRJob

Back to Table of Contents

Modify your PageRank implementation to produce a topic specific PageRank implementation, as described in:

http://www-cs-students.stanford.edu/~taherh/papers/topic-sensitive-pagerank.pdf (http://www-cs-students.stanford.edu/~taherh/papers/topic-sensitive-pagerank.pdf)

Note in this article that there is a special caveat to ensure that the transition matrix is irreducible.
This caveat lies in footnote 3 on page 3:

> A minor caveat: to ensure that M is irreducible when p
> contains any 0 entries, nodes not reachable from nonzero
> nodes in p should be removed. In practice this is not problematic.

and must be adhered to for convergence to be guaranteed.

Run topic specific PageRank on the following randomly generated network of 100 nodes:

s3://ucb-mids-mls-networks/randNet.txt (also available on Dropbox)

```
wget http://ucb-mids-mls-networks.s3.amazonaws.com/randNet.txt
```

which are organized into ten topics, as described in the file:

s3://ucb-mids-mls-networks/randNet_topics.txt (also available on Dropbox)

```
wget http://ucb-mids-mls-networks.s3.amazonaws.com/randNet_topics.txt
```

Since there are 10 topics, your result should be 11 PageRank vectors (one for the vanilla PageRank implementation in 9.1, and one for each topic with the topic specific implementation). Print out the top ten ranking nodes and their topics for each of the 11 versions, and comment on your result. Assume a teleportation factor of 0.15 in all your analyses.

One final and important comment here: please consider the requirements for irreducibility with topic-specific PageRank. In particular, the literature ensures irreducibility by requiring that nodes not reachable from in-topic nodes be removed from the network.

This is not a small task, especially as it it must be performed separately for each of the (10) topics.

So, instead of using this method for irreducibility, please comment on why the literature's method is difficult to implement, and what what extra computation it will require.

Then for your code, please use the alternative, non-uniform damping vector:

```
vji = beta*(1/|Tj|); if node i lies in topic Tj

vji = (1-beta)*(1/(N - |Tj|)); if node i lies outside of topic Tj
```

for beta in (0,1) close to 1.

With this approach, you will not have to delete any nodes. If beta > 0.5, PageRank is topic-sensitive, and if beta < 0.5, the PageRank is anti-topic-sensitive. For any value of beta irreducibility should hold, so please try beta=0.99, and perhaps some other values locally, on the smaller networks.

HW 9.4 Implementation

In [6]:
```
# START STUDENT CODE 9.4
# (ADD CELLS AS NEEDED)

# END STUDENT CODE 9.4
```

HW 9.4 Analysis

--------- OPTIONAL QUESTIONS SECTION --------

HW9.5: (OPTIONAL) Applying topic-specific PageRank to Wikipedia

Here you will apply your topic-specific PageRank implementation to Wikipedia, defining topics (very arbitrarily) for each page by the length (number of characters) of the name of the article mod 10, so that there are 10 topics.

- Once again, print out the top ten ranking nodes and their topics for each of the 11 versions, and comment on your result. Assume a teleportation factor of 0.15 in all your analyses. Run for 10 iterations.
- Plot the pagerank values for the top 100 pages resulting from the 5 iterations run in HW 9.3.
- Then plot the pagerank values for the same 100 pages that result from the topic specific pagerank after 10 iterations run.
- Comment on your findings.

HW 9.5 Implementation

In [5]:
```
# START STUDENT CODE 9.5
# (ADD CELLS AS NEEDED)

# END STUDENT CODE 9.5
```

HW 9.5 Analysis

HW9.6: (OPTIONAL) TextRank

- What is TextRank? Describe the main steps in the algorithm. Why does TextRank work?
- Implement TextRank in MrJob for keyword phrases (not just unigrams) extraction using co-occurrence based similarity measure with with sizes of N = 2 and 3. And evaluate your code using the following example using precision, recall, and FBeta (Beta=1):

```
"Compatibility of systems of linear constraints over the set of natural numbers
Criteria of compatibility of a system of linear Diophantine equations, strict
inequations, and nonstrict inequations are considered. Upper bounds for
components of a minimal set of solutions and algorithms of construction of
minimal generating sets of solutions for all types of systems are given.
These criteria and the corresponding algorithms for constructing a minimal
supporting set of solutions can be used in solving all the considered types of
systems and systems of mixed types."
```

- The extracted keywords should be in the following set:

```
linear constraints, linear diophantine equations, natural numbers, non-strict i
nequations, strict inequations, upper bounds
```

HW 9.6 Implementation

In [7]:
```
# START STUDENT CODE 9.6
# (ADD CELLS AS NEEDED)

# END STUDENT CODE 9.6
```

HW 9.6 Analysis

------- END OF HWK 9 --------