

# Homework #5: Graph Partitioning

*Fall 2018*

120 points (note 20 points for extra credit question below)

Due: 11:59pm, 11/27/2018

## A. Problem

Please use python 2.7 in this assignment

In this assignment, you will implement the Girvan-Newman (GN) algorithm in Python (gn.py) for computing the betweenness of edges in a graph. Here, we consider the betweenness of an edge  $e$  as the sum of the fraction of shortest paths between nodes  $x$  and  $y$  that pass through  $e$ , over all pairs of nodes in the graph. The algorithm takes as the input a graph and outputs the betweenness for each edge in the graph. You may assume that the input graph is **connected**.

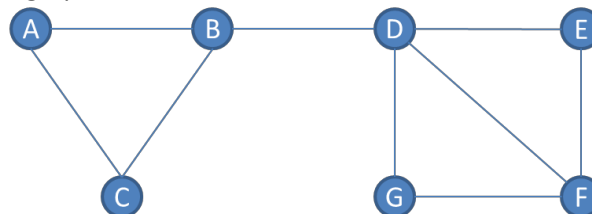
We cannot use any specific libraries except the basic libraries python itself has.

Recall that the GN algorithm works as follows. And we need to write a BFS algorithm here for Step A.

- **[Step A]** for each node  $X$  in the graph  $G$ ,
  1. Run BFS, starting at the node  $X$ ; form a DAG graph  $G'$  that contains edges between different levels of BFS.
  2. For each node  $Y$  in the graph, compute the number of shortest paths from  $X$  to  $Y$ . Recall that this is done by a top-down traversal of  $G'$ .
  3. Based on the results in step 2, for each edge  $e$  in  $G'$ , compute the sum of the fraction of shortest paths from  $X$  that pass through  $e$ . Recall that this is done by a bottom-up traversal of  $G'$ .
- **[Step B]** for each edge  $e$  in the graph  $G$ ,
  1. Sum up the fractions obtained in Step A for  $e$ .
  2. Divide the sum by 2 to give the betweenness of  $e$ .

## A. Input format

The input graph will be provided in a JSON file where each line represents an edge in the graph. For example, the input for the graph below is as follows.



["a","b"]

["a","c"]

["b","c"]

["b","d"]

...

**B. Output format**

You should print the betweenness of edges to output.txt in a format as follows.

(a, b), 5.0

(a, c), 1.0

(b, c), 5.0

(b, d), 12.0

(d, e), 4.5

(d, f), 4.0

(d, g), 4.5

(e, f), 1.5

(f, g), 1.5

**C. Execution format**

python gn.py input-file output-file

**D. Submission**

- Submit the code gn.py

**Extra credit: (20 points)** But note that the maximum point of all homework remains to be 500.

- Implement a program in Spark that computes the edge betweenness in parallel. You can utilize your implementation of the GN algorithm above. Name your script gn-spark.py.
  - Execution format: spark-submit gn-spark.py input-file output-file
- Write a document that explains how your program computes the betweenness scores in parallel.
- Submit the code and the document.
- The library we need here is pyspark.
- Do not use Dataframe here.