Algorithms for Modern Data Models
MS&E 317/CS 263, Spr 2013-14, Stanford University
Instructor: Ashish Goel, Reza Zadeh
Homework 2. Given 5/5/2014, due 5/14/2014, in class.

Collaboration policy: Limited collaboration is allowed – you can discuss the problem with other students in this class but can not collaborate on writing the actual final answer. Please do not look at someone else's solution and do not share your solution with anyone else.

Please do the programming exercise in groups of 3 if possible.

Non-letter grade students: please do any two problems. If you do more, we will grade any two.

1. The edges of an undirected graph $G = (V, E)$ are given to you. Your goal is to create a sketch for each node so that given any set of nodes S, you can quickly estimate the number of nodes who have an edge to a node in S.

   (a) Write down the Map-Reduce algorithm for this sketch-creation.

   (b) Implement (in whichever high level language you like) the Map and Reduce functions, each as a stand-alone executable program that reads a sequence of records from the standard input and writes its output to the standard output. Use your implementation along with the simple map-reduce runner supplied by the instructor to produce this sketch for the supplied data file. Your mapper should accept the number of hash functions to use in the sketch as a command-line parameter.

   (c) Write a program to read the output of the above problem and repeatedly answer queries, where each query is a list of nodes and the answer is an estimator for the number of nodes adjacent to at least one node in the list.

   (d) Argue that your code is robust to repeated edges, to having an edge (u, v) listed as
   u v
   or
   v u
   or both, and to having nodes that are not adjacent to any edge.

   (e) Make sure the executables run on the Stanford corn machines. Submission instructions and all the code and instructions you need are in `http://www.stanford.edu/~ashishg/amdm/handouts/single_node_shuffle/`.

2. You are given a set of N sensors placed at integer positions $1, 2, \ldots N$ on a line. Sensor i observes a value $v(i, t)$ at time t. The sensors can pass messages to their neighbors infinitely fast, but passing and storing messages comes with some cost. Your goal is to create a randomized message-passing algorithm such that at any time t, any node i can draw a uniformly random sample from the values seen by nodes $i-d, \ldots, i$ during the last w time steps without passing any additional messages. Design an efficient algorithm for this problem, and analyze the expected number of messages passed by a node at every time step and the amount of storage space used at each node.

3. (a) Given a stream of values $a_0, a_1, \ldots, a_t$, your goal is to answer queries of the form "What is the median of the last w values?". Design a streaming algorithm for this problem and analyze the memory requirement and the accuracy.

   (b) The stream-sampling algorithm we have seen in this class suffers from the problem that while each query returns a uniformly random sample, different queries don't return independent samples. How would you address this problem in the above context? In other words, how can you tune your algorithm in (a) so that with high probability, you never return an estimate for the median which is outside the $[1/2 - \delta, 1/2 + \delta]$-quantile? You can assume that $t \leq T$ if you wish and express your algorithm and answers in terms of T.

4. Describe one problem (very briefly, eg. in 3-5 sentences) that you think would be important to discuss in this class.