*Assignment #4*

1. [Graph Query Processing] (20) The following questions test your understanding on basic graph algorithms
2. (10) Given a directed graph 𝐺(𝑉, 𝐸, 𝐿) with 𝑉 the node set, 𝐸 the edge set and 𝐿 a function that assigns to each edge 𝑒 ∈ 𝐸 a label 𝐿(𝑒). A label constrained reachability query 𝑄(𝑠, 𝑡, 𝑀) tests if there exists a path from a source node 𝑠 to a target node 𝑡, which consists of edges having a label from a label set 𝑀. Give an algorithm (pseudo-code) to answer query 𝑄.

(hint: A straightforward way is to revise BFS or DFS traversal)

1. (10) Consider a network 𝐺(𝑉, 𝐸) of servers, where each edge 𝑒 = (𝑢, 𝑣) represents a communication channel from a server 𝑢 to another server 𝑣. Each edge has an associated value 𝑟(𝑢, 𝑣), which is a constant in [0,1]. The value represents the reliability of the channel, i.e., the probability that the channel from server 𝑢 to server 𝑣 will not fail. Assume that these probabilities are independent. Give an algorithm (pseudo-code) to find the most reliable path between two given servers. Give a correctness proof and complexity (in Big O notation) of your algorithm.

(hint: Transform the weight to non-negative numbers and the problem may become very familiar to you).

1. [Approximate Query Processing] (30) This question continues our discussion on using data synopsis for query processing based on data-driven approximation. You are given a vector of numbers: [127, 71, 87, 31, 59, 3, 43, 99, 100, 42, 0, 58, 30, 88, 72, 130], each data point records the frequency of communication of a server in a 5-minute interval. For example, in the first 5 minutes, 127 contacts are observed. In the next 5 minutes, 71 contacts, ...

(hint: Discard the lowest level (high-resolution) coefficients (i.e. only keep the first 50% of coefficients).

1. (10) Give the Haar decomposition and draw a corresponding error tree for the contacts data vector.
2. (10) Give the process and result for reconstructing the frequency during time interval [15, 20] using Haar decomposition
3. (10) Use Haar decomposition and error tree to compute the total number of communications between time interval [15, 30].
4. [MapReduce] (30) This set of questions test the understanding and application of MapReduce framework.
5. (15) Facebook updates the “common friends” of you and response to hundreds of millions of requests every day. The friendship information is stored as a pair (Person, [List of Friends]) for every user in the social network. Write a MapReduce program to return a dictionary of common friends of the form ((User i, User j), [List of Common Friends of User i and User j]) for all pairs of i and j who are friends. The order if i and j you returned should be the same as the lexicographical order of their names. You need to give the pseudo-code of a main function, and both Map() and Reduce() function. Specify the key/value pair and their semantics (what are they referring to?).
6. (15) Top-10 Keywords. Search engine companies like Google maintains hot webpages in a set 𝑅 for keyword search. Each record 𝑟 ∈ 𝑅 is an article, stored as a sequence of keywords. Write a MapReduce program to report the top 10 most frequent keywords appeared in the webpages in 𝑅. Give the pseudo-code of your MR program.
7. [Graph Parallel Models] (20) This sets of questions relate to MapReduce for graph processing
8. (10) Consider the common friends problem in Problem 3.a. We study a “2-hop common contact problem”, where a list should be returned for any pair of friends i and j, such that the list contains all the users that can reach both i and j within 2 hops. Write a MR algorithm to solve the problem and give the pseudo code.
9. (10) We described how to compute distances with mapReduce. Consider a class of d-bounded reachability queries as follows. Given a graph 𝐺, two nodes 𝑢 and 𝑣 and an integer 𝑑, it returns a Boolean answer YES, if the two nodes can be connected by a path of length no greater than 𝑑. Otherwise, it returns NO. Write an MR program to compute the query 𝑄(𝐺, 𝑢, 𝑣, 𝑑) and give the pseudo code. Provide necessary correctness and complexity analysis