CS425 Fall 2015 – Homework 1
(a.k.a. “All the President’s Interns”)

Out: Aug 25, 2015. Due: Sep 17, 2015 (Start of Lecture)

**Topics:** Time and Synchronization, Snapshots, Clouds, Failure Detectors (Lectures 1-6)

**Instructions:**

1. **Attempt any 8 out of the 10 problems** in this homework (regardless of how many credits you’re taking the course for). If you attempt more, we will grade only the first 8 solutions that appear in your homework (and ignore the rest). Choose wisely!
2. Please hand in **hardcopy solutions that are typed** (you may use your favorite word processor. We will not accept handwritten solutions. Figures and equations (if any) may be drawn by hand.
3. Please **start each problem on a fresh sheet (not just page)**, and **type your name at the top of each sheet**.
4. Homeworks will be **due at the beginning of class on the day of the deadline. No extensions.**
5. Each problem has the same grade value as the others (10 points each).
6. Unless otherwise specified, the only resources you can avail of in your HWs are the provided course materials (slides, textbooks, etc.), and communication with instructor/TA via discussion forum and e-mail.
7. You can discuss lecture concepts and the questions on Piazza and with your friends, but you cannot discuss solutions or ideas. All work must be your own.

**Prologue:** In the year 2016 AD, the US will hold Presidential Elections. Most Presidential Campaigns today run distributed systems and cloud computing for storage and analytics of data such as the campaign, events, voters, schedules, etc., (e.g., both the Obama and Romney campaigns in 2012 built and used distributed systems, including cloud computing and mobile computing).

This homework uses fictitious stories and characters from the ongoing presidential campaigns to frame the homework problems. The choice of candidate names below is purely random. Any resemblance to persons, places, or events, living or dead, past, present or future, is purely coincidental. These stories and this homework is aimed neither at supporting or endorsing, nor at criticizing or disparaging, any candidate, campaign, people in campaigns, political parties or affiliations, or voters.
Problems:

1. An intern at the Scott Walker campaign is a fan of Cristian’s algorithm, and finds that the round-trip time for a synchronization round of messages is 1.29 ms. The intern would like to find the error in the run, and so he measures the minimum delays. On the client side, he finds that there is a delay of at least 13.7 microseconds for a packet to get from an application to the network interface and a delay of 19.3 microseconds for the opposite path (network interface to application buffer). Before the intern can measure the corresponding minimum delays for the server, he is called away to Iowa. Can you help the intern find the error, given the data just presented?

2. An intern in the Donald Trump Campaign sees the CS425 website, and likes that timeline at the top of the webpage. So he writes a distributed system that generates a similar timeline, and it is given below (Can you make out the timeline pattern from the CS425 website in this figure?). Arrows indicate messages and dots indicate steps/instructions. What Lamport timestamps did the intern mark? Mark timestamps at all events at all processes (i.e., the point of each message send and message receipt, and each step).

3. An intern in Jeb Bush’s campaign sees this, and says she can do one better by using vector timestamps. She repeats the above problem using vector timestamps instead of Lamport timestamps. What did she come up with it? Mark timestamps at all events at all processes (i.e., the point of each message send and message receipt, and each step).

4. An intern in Martin O’Malley’s Campaign thinks things are moving too fast, and they should snapshot their cluster. She implements the Chandy-Lamport
algorithm. While testing, she logs a particular run of the events in their cluster, and it is given in the figure below. In the figure, a-e are regular application messages, with S(a) denoting the send event of a and R(a) denoting its receipt event. Markers shown as dotted lines. Can you help the intern find the snapshot recorded by this run? Don’t forget to include both process states and channel states. For process states, you can use the name of the latest event at that process (For initial state, just say “Initial state”. Note that Markers don’t count as events).

5. A hip intern in Marco Rubio’s campaign decides that the way to beat the other candidates’ campaigns is to use Hadoop for data analytics, and run the latest version of Hadoop over YARN. He wants to train everyone in the campaign to use Hadoop. He creates a quiz to test the campaign’s knowledge of Hadoop. Would you qualify? Test for yourself! In Hadoop YARN, from among {AM, RM, NM, map task, reduce task, none}, choose which one entity fits each of the following descriptions.
   a. This entity writes the output to HDFS.
   b. Each server contains exactly one instance of this at all times.
   c. This entity receives and processes key-value pairs grouped by key.
   d. This entity runs makes scheduling decisions.
   e. This entity sends task scheduling requests to the scheduler.
   f. This entity sends shuffle traffic.
   g. This entity decides which job (among those waiting) gets a free container.
   h. This entity receives heartbeats from the other two entities.
   i. This entity reads input from HDFS.

6. But Hillary Clinton’s Campaign has caught on! Their intern teaches Mapreduce this way – In MapReduce, one writes a program for Map that processes one
input line at a time and outputs a (key, value) or nothing; and one writes a program for Reduce that processes an input of (key, all values for key). The iteration over input lines is done automatically by the MapReduce framework. The intern in Hillary Clinton’s Campaign would like to know who are the Twitter users most similar to her, and would like to use Hadoop for this. She uses an input file containing information from Twitter (which is an asymmetrical social network) about which users “follow” which other users. If user a follows b, the entry line is (a, b) – you can assume this data is already sharded HDFS and can be loaded from there. Can you help the intern? Write a MapReduce program (Map and Reduce separately) that outputs the list of all users U who satisfy the following three conditions simultaneously: U has at least a million followers, and U herself/himself follows at least 10 users, and U follows at least one user V who in turn has at least a million followers (e.g., @HillaryClinton satisfies this). You can chain Mapreduces if you want (but only if you must, and even then, only the least number). Your program must be as highly parallelizable as possible.

7. An intern in Bernie Sanders’ Campaign has designed a failure detection protocol that is a modified ring failure detection protocol. It works as follows: each process i selects k other processes at random and asks these k processes to send it (i) heartbeats. Heartbeats are not relayed (so this is not gossip, but more like ring failure detection), and process i times out if it doesn’t receive heartbeats.

   a. Another intern in the campaign claims that this algorithm provides completeness up to k failures. Are they right? If yes, argue why (informal proof). If no, give a counter-example, and also state what completeness the algorithm does provide.

   a. Another intern in the campaign claims this algorithm satisfies accuracy. Are they right? If yes, argue why (informal proof). If no, give a counter-example, and also state what kind of accuracy the algorithm does provide.

8. An intern in Jim Webb’s campaign thinks gossip is the way to increase the popularity of her candidate. To this extent, she decides to use a gossip-based membership protocol in their campaign cluster. However, she has a bug where she notices a failed node doesn’t seem to go away at all, and stays around forever in membership lists! Can you help the intern with your knowledge of CS425? What is the underlying bug that is likely causing this behavior?

9. An intern from the Lincoln Chafee campaign decides to map on a timeline the barbs traded among candidates during a presidential debate, and while analyzing this he would like to assign timestamps to events of only his candidate. Can you help the intern? The timeline is shown in the following figure. All five processes start with sequence numbers (as applicable) containing all zeroes. Only for process P2, mark the Lamport timestamps at the point of
each message send and message receipt. (If you wish you can mark other process’ timestamps, but those will not be graded.)

10. Unfortunately for the intern from the previous problem, his girlfriend works in the Carly Fiorina campaign. She is also analyzing the same debate timeline. She is smarter than her boyfriend, so she decides to repeat Problem 9 with vector timestamps instead of Lamport timestamps. Can you help the intern? Again mark the timestamps only for process P2. (If you wish you can mark other process’ timestamps, but those will not be graded.)