Exercise 1
Q1) [2pts] Which of the following ML algorithms has complex non-sequential data access patterns for training?

A. Tree-based methods  
B. GLMs trained with BGD  
C. K-Means clustering  
D. Both A and B  
E. All of A to C  
F. None of the above
Q2) [2pts] Which of the following ML model families is XGBoost primarily designed for?

A. GLMs
B. Tree-based methods
C. Deep learning
D. Bayesian networks
E. All of A to D
F. None of the above
Q3) [2pts] Which of the following ML systems was designed for the Hadoop/MapReduce backend?

A. MADlib
B. Spark ML/MLlib
C. Mahout
D. TensorFlow
E. Dask
F. None of the above
Exercise

Q4) [2pts] Which function in the RDBMS UDA API roughly corresponds to the role of Reduce in MapReduce?

A. Initialize
B. Transition
C. Merge
D. Finalize
E. Both A and C
F. None of the above
Q5) [6pts] Briefly discuss 2 advantages of in-RDBMS ML over Parameter Server.

Q6) [6pts] Briefly discuss 2 advantages of Parameter Server over Spark ML.

Q7) [6pts] Briefly discuss 2 disadvantages of in-RDBMS ML over Spark ML.

Q8) [6pts] Briefly discuss 2 reasons why SGD has become the optimization procedure of choice in large-scale ML.
**Exercise**

**Q9) [8pts]** Briefly discuss 2 systems-level advances made in XGBoost to improve scalability and/or efficiency of the training process and why they are effective.

**Q10) [10pts]** Briefly explain the core abstractions of 2 ML systems that were designed primarily to mitigate the developability concerns of large-scale ML.
Q11) [10pts] Write pseudocode for a MapReduce job to compute the column-wise sums of a given large matrix. Make sure to explain your assumption on how the dataset is stored/sharded to begin with.

Q12) [10pts] Assume you are given a large matrix stored as a table with rows as tuples and columns as attributes. Write pseudocode for a UDA to compute the column-wise sums of the matrix. Make sure to explain your aggregation state.
Exercise

Q13) [6pts] Suppose you are using SGD to train an ML model on a large dataset. You are given that the shuffle step to randomize the data order takes 5min, while running an epoch of SGD takes 2min.

Running SGD with shuffles before every epoch takes 20 epochs to converge, while running SGD with only one shuffle upfront takes 30 epochs to reach the same accuracy.

Which approach among the above two is faster from a total runtime standpoint?
Q14) True or False [2pts each]:

A. It is always possible for Parameter Server to be faster than a single-node scalable system for SGD if the former is given enough worker nodes.

B. XGBoost makes no algorithmic modifications to the standard GBDT algorithm published before that.

C. Model averaging-based SGD can work reasonably well for training convex models such as GLMs.