

MapReduce

Use pseudocode to write MapReduce functions necessary to solve the problems below. Also, make sure to fill out the correct data types. Some tips:

- The input to each MapReduce job is given by the signature of the `map()` function.
- The function `emit(key k, value v)` outputs the key-value pair `(k, v)`.
- The `for(var in list)` syntax can be used to iterate through `Iterables` or you can call the `hasNext()` and `next()` functions.
- Usable data types: `int`, `float`, `String`. You may also use lists and custom data types composed of the aforementioned types.
- The method `intersection(list1, list2)` returns a list that is the intersection of `list1` and `list2`.

1. Given the student's name and the course taken, output each student's name and total GPA.

Declare any custom data types here: <code>CourseData:</code> <code>int courseID</code> <code>float studentGrade // a number from 0-4</code>	
<code>map(String student, CourseData value):</code> <code>emit(student, value.studentGrade)</code>	<code>reduce(<u>String</u> key,</code> <code>Iterable< <u>float</u> > values):</code> <code>totalPts = 0</code> <code>totalClasses = 0</code> <code>for (grade in values):</code> <code>totalPts += grade</code> <code>totalClasses++</code> <code>emit(key, totalPts / totalClasses)</code>

2. Given a person's unique `int` ID and a list of the IDs of their friends, compute the list of mutual friends between each pair of friends in a social network.

Declare any custom data types here: <code>FriendPair:</code> <code>int friendOne</code> <code>int friendTwo</code>	
<code>map(int personID, list<int> friendIDs):</code> <code>for (fID in friendIDs):</code> <code>if (personID < fID):</code> <code>friendPair = (personID, fID)</code> <code>else:</code> <code>friendPair = (fID, personID)</code> <code>emit(friendPair, friendIDs)</code>	<code>reduce(<u>FriendPair</u> key,</code> <code>Iterable< <u>list<int></u> > values):</code> <code>mutualFriends = intersection(values.next(),</code> <code>values.next())</code> <code>emit(key, mutualFriends)</code>

3. A. Given a set of coins and each coin's owner, compute the number of coins of each denomination that a person has.

Declare any custom data types here: CoinPair: String person String coinType	
map(String person, String coinType): coinPair = (person, coinType) emit(coinPair, 1)	reduce(____ <u>CoinPair</u> ____ key, Iterable< ____ <u>int</u> ____ > values): total = 0 for (count in values): total += count emit(key, total)

- B. Using the output of the first MapReduce, compute the amount of money each person has. The function `valueOfCoin(String coinType)` returns a float corresponding to the dollar value of the coin.

map(____ <u>CoinPair</u> ____ key, ____ <u>int</u> ____ amount): emit(coinPair.person, valueOfCoin(coinPair.coinType) * amount)	reduce(____ <u>String</u> ____ key, Iterable< ____ <u>float</u> ____ > values): total = 0 for (amount in values): total += amount emit(key, total)
---	--

Warehouse-Scale Computing

Power Usage Effectiveness (PUE) = (Total Building Power) / (IT Equipment Power)

Total Building Power = IT Equipment + Power supplies + Networking equipment + Cooling equipment

Sources speculate Google has over 1 million servers. Assume each of the 1 million servers draw an average of 200W, and that Google pays an average of 6 cents per kilowatt-hour for datacenter electricity.

- a) Estimate Google's annual power bill for its datacenters. Ignore the power cost of networking equipment. Assume 365 days (8760 hours) in a year.

1,000,000 servers * 0.2kW/server * 0.06 dollars/kW-hr * 8760 hrs/yr = \$105.12 M/yr

- b) Google reduced the PUE of a 50,000 machine datacenter from 1.5 to 1.25 without decreasing the power supplied to the servers. What's the cost savings per year?

50,000 servers * 0.2kW/server * (1.5 - 1.25) * 0.06 dollars/kW-hr x 8760 hrs/yr = \$1.314 M/yr