Problem 3

Part a

MapReduce for finding max integer (n_i) of a set of integers N:

Map 1: Hash the integers into g buckets of size $\frac{N}{g}$ integers, find max in each bucket. Group buckets into k groups of size $\frac{g}{k}$ buckets. Output the value of the max integer with key = group id of each bucket

Input: N

Output: (d, max n in j^{th} bucket) where $d \in \mathbb{Z}$, d $\in [1, k]$ and $j \in \mathbb{Z}$, j $\in [1, g]$

Reduce 1: Group results by group of buckets (d) per reducer, find max in that group of buckets and output it

Input:

(1, (max n in $bucket_{1,1}$, max n in $bucket_{1,2}$, ... , max n in $bucket_{1,j}$, ... , max n in $bucket_{1,g/k}$)) group 1

(2, (max n in $bucket_{2,1}$, max n in $bucket_{2,2}$, ... , max n in $bucket_{2,j}$, ... , max n in $bucket_{2,g/k}$)) group 2

...

(d, (max n in $bucket_{d,1}$, max n in $bucket_{d,2}$, ... , max n in $bucket_{d,j}$, ... , max n in $bucket_{d,g/k}$)) group d

...

(k, (max n in $bucket_{k,1}$, max n in $bucket_{k,2}$, ... , max n in $bucket_{k,j}$, ... , max n in $bucket_{k,g/k}$)) group k

Output:

(1, max in $group_1$)

(2, max in $group_2$)

...

(d, max in $group_d$)

...

(k, max in $group_k$)

Map 2:

Pool all outputs of Reduce 1, assign same key to all key-value pairs, keep value same.

Input: output of Reduce 1

Output: (0, max in $group_1$)

(0, max in $group_2$)

...

(0, max in $group_d$)

...

(0, max in $group_k$)

Reduce 2: Collect all in one reducer and return max integer

Input: (0, (max in $group_1$, max in $group_2$, max in $group_3$,...)) *Output:* n_i (max n in N)

Part b

Similarly to part a, except instead of max value, return a tuple of average value $\bar{X}_i$ and size of the population n_i that this average was created from. To combine N averages $\bar{X}_1$, $\bar{X}_2$, $\bar{X}_3$, etc that were calculated for n_1, n_2, n_3... n_N values respectively:

$$\frac{\bar{X}_1 \times n_1 + \bar{X}_2 \times n_2 + ... + \bar{X}_N \times n_N}{n_1 + n_2 + ... + n_N}$$

Part c

Map:

Input: Multiset of N integers

Output: key-value pairs: (n_i, n_i) - same key and value

Reduce: Collect non distinct integers per reducer and discard all but one instances.

Input: $(n_i, (n_i, n_i, n_i, ...))$

Output: n_i

Part d

Map 1:

Input: Multiset of N integers

Output: key-value pairs: $(n_i, 1)$

Reduce 1: Collect by integer keys per reducer and discard all but one values.

Input: $(n_i, (1, 1, 1, ...))$

Output: $(n_i, 1)$

Map 2:

Input: Set $S \subseteq N$ of distinct integers .

Output: key-value pairs: $(1, n_i)$

Reduce 2:

Collect all words and return size of the value tuple:

Input: $(1, (n_1, n_2, ...n_S))$

Output: $|S|$

If reducer size is not large enough - can partition words into buckets similar to part a.