Exercise Sheet 7

Due date: Monday, July 03 until 13:00

- Please upload your solutions to Moodle.
- Hand in your solutions in groups of **two to three students**.
- Hand in the solutions of your group as a single PDF file.
- The solutions for this exercise sheet will be published after the deadline.
- A discussion regarding this exercise sheet will take place on **Friday, July 07 14:30** in room AH II.
- Note that Exercise 1 gives 0 points and will not be corrected.

Exercise 1 (Star Join) 0 points

We consider a data structure that often arises in commercial data. A large company wants to keep track of all their sales. To do that they keep a fact table $F = (A_1, \dots, A_m)$, where every tuple represents a single sale. The key attributes A_i represent the important components of the sale, such as purchaser ID, the ID of the purchased item, the store etc. For every key item A_i, there is a dimension table $D_i = (A_i, B_i)$ which contains further information, like address of the purchaser or prize of the item. Usually the size f of the fact table is much larger than the sizes d_i of the dimension tables or the number ob key attributes. Analytic queries on these data bases usually join the fact table with several of the dimension tables and aggregate the result. Such a join is called *star join*.
Consider the following Map-Reduce algorithm for the star join $F \bowtie D_1 \bowtie \dots \bowtie D_m$. Let s be the number of reducers, let $s_1, \dots, s_m \in \mathbb{N}$ such that $\prod_{i=1}^{m} s_i = s$ be the shares of the key attributes and let $h_1, \dots, h_m$ be independently chosen truly random hash functions $h_i \colon V_i \to [s_i]$, where V_i is the set of values of the attribute A_i.

MAP:	On input $(F, (a_1, \dots, a_m))$,	emit $((h_1(a_1), \dots, h_m(a_m)), (a_1, \dots, a_m))$.
	On input $(D_i, (a_i, b_1))$,	emit all key-value pairs $((p_1, \dots, p_{i-1}, h_i(a_i), p_{i+1}, \dots, p_m), (a_i, b_1))$ for $p_j \in [s_j]$ for all $j \neq i$.
REDUCE:	On input $(\bar{p}, values)$,	compute $\mathcal{Q}(\bar{p}) := \mathcal{F}(\bar{p}) \bowtie \mathcal{D}_1(\bar{p}) \bowtie \dots \bowtie \mathcal{D}_m(\bar{p})$, where $\mathcal{F}(\bar{p}) := \{t \mid (F, t) \in values\}$ and $\mathcal{D}_i(\bar{p}) := \{t \mid (D_i, t) \in values\}$, and emit all pairs (Q, t) for $t \in \mathcal{Q}(\bar{p})$.

a) Compute the replication rate of this algorithm.

b) Compute the expected load of this algorithm.

c) Assume $s = \prod_{i=1}^{n} d_i$. Which choice of the s_i minimizes the values in (a) and (b)?

Hint: You may use the following inequality, known as AM-GM inequality, without proving it. Let $x_1, \dots, x_n \in \mathbb{R}^{\geq 0}$, then

$$\frac{\sum_{i=1}^{n} x_i}{n} \geq \sqrt[n]{\prod_{i=1}^{n} x_i}, \tag{1}$$

where equality holds if and only if $x_1 = x_2 = \dots = x_n$.

Exercise 2 (Random Elements from Stream) — 8 points

Suppose you receive a stream $a_1, \ldots, a_n$ of positive integers a_i for all $i \in [n]$. Let $z = a_1 + a_2 + \ldots a_n$ be the sum of all elements in the stream.
Describe a streaming sampling algorithm that has the probability $\frac{a_i}{z}$ to pick the element with the index i and prove its correctness. The algorithm shall use at most $O(log\ z)$ bits of space.
Note: Just like the length n of the stream, the value z is not known to your algorithm in advance.

Exercise 3 (Strongly-2-Universal Hashing) — 8+4=12 points

a) Consider the family

$$\mathcal{H} \coloneqq \{(ax + b) \mod p \colon 0 \leq a, b \leq p - 1\}$$

of hash function from $\mathbb{U} = \{0, \ldots, p-1\}$ to $\mathbb{T} = \{0, \ldots, p-1\}$ where p is a prime number. Show that $\mathcal{H}$ is strongly 2-universal.

- Show this directly. In particular, do **not** just apply (the more general) statement from the lecture.
- Point out where you use that p is a prime number.

b) Assume that $|\mathbb{U}| \geq 2$ and let $\mathbb{T}$ be arbitrary (non-empty). Does there exist a strongly 2-universal family $\mathcal{H}'$ of hash function from $\mathbb{U}$ to $\mathbb{T}$ such that $|\mathcal{H}'| < |\mathbb{T}|^2$?

If yes, describe a construction of such a family. If no, prove that no such family exists.