1 Introduction

For our report, we will review the paper "Optimal Quantile Approximation in Streams" [KLL16].

For a set of items[1] $x_1, ..., x_n$ in an ordered universe $\mathcal{U}$, the **rank** of an item x is the number of items in the stream such that $x_i \leq x$, and the **quantile** of x is the *fraction* of items such that $x_i \leq x$ (i.e. (quantile) = (rank)/n).

Quantiles are natural representations of non-parametric distributions, which cannot be summarized by statistics such as mean and variance [Wan+13]. They correspond to evaluations of the cumulative distribution function at different points which characterize the distribution and are pivotal in hypothesis testing. For example, the value of the $\frac{1}{2}$-quantile or the median is widely known as a robust statistic of the distribution that is less sensitive to outliers. Furthermore, items which have extreme quantile values can be detected as outliers.

Ideally, one summarizes a distribution via quantiles without storing all elements of the stream. Unfortunately, [MP80] has shown that any single-pass algorithm that *exactly* computes kth-largest element of a stream requires $\Omega(n)$ memory. Hence, we will consider the following *approximate* quantile estimation problem.

Definition 1

The **single quantile approximation** problem is about estimating the quantile of any item up to $\pm\varepsilon$ additive error with failure probability δ.

More explicitly, given $x_1, ..., x_n$ in streaming fashion, we produce an approximate (random) rank function $\tilde{R}$, with the following property:

For any item x, the estimate $\tilde{R}(x)$ approximates the true rank $R(x)$ to within $\pm\varepsilon n$ (additively) with probability at least $1 - \delta$ over the randomness of $\tilde{R}$.

If $\tilde{R}(x)$ is non-decreasing, the single quantile approximation problem can be leveraged to find the value of the α-th quantile as in the following remarks.

Remark 2. Given an approximate rank function $\tilde{R}$ from any single quantile approximation algorithm, we can compute an approximate value of the α-quantile with the same overall memory cost by querying elements until one finds an element $x \in \mathcal{U}$ such that $\tilde{R}(x) = \alpha n$.

Denoting $x_{\alpha\pm\varepsilon}$ as the true value of the $\alpha \pm \varepsilon$ quantiles, the guarantee in Definition 5 applied to $x_{\alpha\pm\varepsilon}$ ensures that $\tilde{R}(x_{\alpha-\varepsilon}) \leq \alpha n$ and $\tilde{R}(x_{\alpha+\varepsilon}) \geq \alpha n$ with probability $\geq 1 - 2\delta$. Thus, any element x we find with $\tilde{R}(x) = \alpha n$ must have a true quantile in $[\alpha - \varepsilon, \alpha + \varepsilon]$ (by the non-decreasing property of $\tilde{R}$) with probability $\geq 1 - 2\delta$.

Remark 3. The previous remark does not have any constraints on time complexity. It turns out that the sketch we will present computes $\tilde{R}(x)$ by comparing x to the elements stored in the sketch. Hence, we can find x from the stored sketch elements such that $\tilde{R}(x) = \alpha n$ in the previous remark in time on the order of the memory cost of our sketch.

The main result of the paper we are reporting on exhibits an optimal, comparison-based sketch and a matching lower bound.

Theorem 1 ([KLL16])

There exists a randomized, comparison-based algorithm for the single quantile approximation problem with a non-decreasing $\tilde{R}(x)$ and $O((1/\varepsilon)\log\log(1/\delta))$ space. This is optimal in the sense that any randomized, comparison-based algorithm must use $\Omega((1/\varepsilon)\log\log(1/\delta))$ space.

For the sake of simplicity, we will ignore mergeability concerns even though it was discussed in the original paper. It turns out that the bulk of the algorithm is mergeable except possibly the final part involving a sketch by [GK01]. Henceforth, we will use $\lesssim$ to mean inequalities up to constants.

[1] We can imagine these are real numbers, but it suffices for the items to be totally ordered.

1.1 Previous Results

[KLL16] gives a fairly complete overview of the relevant literature but we will just restate parts of it here for ease of reference. We also refer to [Wan+13] for a survey of previous research.

The best *deterministic* algorithm (the "GK sketch" from [GK01]) for the single quantile approximation problem takes space $O((1/\varepsilon)\log(n\varepsilon))$. A recent result in [CV20] proves a matching lower bound for such deterministic, comparison-based algorithms. A less optimal algorithm (the "MRL sketch", due to [MRL99]) uses $O((1/\varepsilon)\log^2(n\varepsilon))$, but is relevant in this report because it forms the base for the optimal randomized algorithm.

Allowing randomization, we can combine subsampling from the original stream with a deterministic algorithm to produce better results. If the size of the stream n is known before hand, we can take a uniform sample of size $n' = O((1/\varepsilon)^2\log(1/\varepsilon))$ and feed it into a GK sketch to obtain a space complexity of $O((1/\varepsilon)\log(1/\varepsilon))$ (observed in [MRL99]). Notably, this eliminates the dependence on n, and one needs a result of [FO17] to obtain this space complexity without prior knowledge of n. This was the best known space complexity for the randomized algorithm prior to [KLL16].

2 Optimal Sketch

In this section, we detail the construction for an optimal sketch from [KLL16]. The optimal sketch builds on the basic deterministic algorithm of [MRL99] which can be thought of as a series of compactors, and optimizes various aspects of the basic algorithm.

2.1 Compactors

The fundamental block of the MRL sketch is known as a k-compactor. A k-compactor can either store k elements of identical weights w or compress them into $\frac{k}{2}$ elements of weight $2w$ by selecting every alternate element (elements at odd or even indices) in its sorted list—the latter is known as a compaction. Referring to the Figure 1, a compaction introduces at most w error to the rank estimation with k-stream elements, via the elements of the compressed list, while halving the memory cost. To compress a stream with mk-elements of weight w with $m \in \mathbb{N}$, a k-compactor performs a compaction operation whenever it already has k elements and is about to receive a new one (it also compacts the last k elements). Since the rank of an element x is the sum of the ranks of x within the m blocks of length k, the error in the rank of x is at most mw.

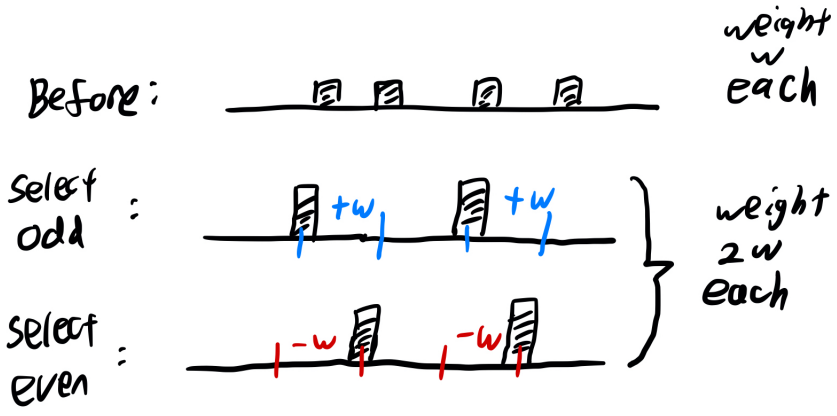

Figure 1: Original stream and the streams after odd and even compactions. The rank estimation errors and their associated intervals are labelled in blue and red.

2.2 Iterated compactors

We are ready to describe the MRL sketch. Suppose we have H compactors of capacity k each, and we connect them sequentially (i.e. the stream feeds elements of weight 1 into compactor 1, and any output from compactor 1 is fed into compactor 2 and so on). In the streaming model, a

compactor is only created when it receives the output of a previous compactor since the length of the stream n is unknown a priori. Items in the h-th compactor have weight $w_h = 2^{h-1}$.

At most $\lfloor n/2^{h-1} \rfloor$ items reach the h-th compactor, so we only need $H = \lfloor \log_2(n/k) \rfloor + 1$ compactors.

Now we try to estimate the total error of the rank estimate $\tilde{R}(x)$ computed as the rank of x within the weighted elements inside the compactors. Earlier on, we argued that one compact operation at weight w gives error w. The h-th compactor receives a total weight of at most n, so the number of compact operations it performs is at most $m_h = n/(kw_h)$ so by just naively adding the errors up, the worst-case total error is

$$\begin{aligned} \mathsf{Error} &\le \sum_{h=1}^{H} m_h w_h && (1) \\ &\le H \cdot (n/k) && (2) \\ &\lesssim \frac{n}{k} \log\left(\frac{n}{k}\right) . && (3) \end{aligned}$$

Setting $k = O((1/\varepsilon)\log(\varepsilon n))$ gives εn error with space $kH = O((1/\varepsilon)\log^2(\varepsilon n))$, since each compactor stores k elements and there are H compactors. Note that this deterministic algorithm falls short of the optimal space of $O\left((1/\varepsilon)\log(n\varepsilon)\right)$.

2.3 Odd/even randomness

With randomization, one significant improvement due to [Aga+13] is that one can independently select whether to discard the odd/even elements with equal probability. By the earlier discussion in Subsection 2.1, this has the effect that the errors from each compaction operations are independent $\pm w$ (or zero)—notice that the error intervals for the odd and even cases overlap in Figure 1. This allows us to use a version of Hoeffding's inequality (stated in the following convenient form):

Lemma 4 (Hoeffding, restated)
Let S be a linear combination of independent Rademacher variables. Then, with probability $1 - \delta$,

$$|S| \le \sqrt{(\operatorname{Var} S) \cdot \log \frac{1}{\delta}}. \tag{4}$$

Now, we can apply the above Lemma to the compactor errors which are $\sum_{h=1}^{H} m_h$ independent variables that are $\pm w$ (or zero) to obtain with probability $\ge 1 - \delta$:

$$\begin{aligned} \mathsf{Error} &\le \sqrt{\sum_{h=1}^{H} m_h w_h^2 \log \frac{1}{\delta}} && (5) \\ &\le \sqrt{\frac{n}{k} \sum_{h=1}^{H} w_h \log \frac{1}{\delta}} && (6) \\ &\lesssim \frac{n}{k} \sqrt{\log(1/\delta)} . && (w_h = 2^{h-1}, \quad 2^H \le \tfrac{2n}{k}) \end{aligned}$$

Setting $k = O((1/\varepsilon)\sqrt{\log(1/\delta)})$ gives $\mathsf{Error} \le \varepsilon n$ and $\mathsf{Space} \le kH = O\left((1/\varepsilon)\log(\varepsilon n)\sqrt{\log(1/\delta)}\right)$.

2.4 Compactor size decay

Note that the higher levels contribute exponentially larger values to the sum in Eq. (6). Hence, one novel idea of [KLL16] exploits this observation to reduce the memory cost by decreasing the compactor capacity k for lower levels (introducing more compaction operations and thus errors) since they do not contribute much to the error anyway.

Suppose that the h-th compactor instead has capacity k_h. The error and space complexity bounds become

$$\mathsf{Error} = \sqrt{\left(\sum_{h=1}^{H} m_h w_h^2\right) \log \frac{1}{\delta}} \tag{7}$$

$$\leq \sqrt{\left(\sum_{h=1}^{H} \frac{n w_h}{k_h}\right) \log \frac{1}{\delta}} \ , \tag{8}$$

$$\mathsf{Space} = \sum_{h=1}^{H} k_h \ . \tag{9}$$

We also require $n \leq \sum_{h=1}^{H} k_h w_h$ to prevent overflow.

Previously when k_h was constant, the summation in the error term was exponentially increasing in h due to $w_h = 2^{h-1}$, so the result is a constant multiple of the last term at $h = H$. However, we can make k_h increase exponentially at a rate < 2 up to the last term k_H which is a parameter we will choose in our sketch (e.g. $k_h \approx k_H(2/3)^{H-h}$) so that the summation in the space term is still exponential. Thus, ignoring constants, we have the same error bound if we pick $k_H = O((1/\varepsilon)\sqrt{\log(1/\delta)})$, but the space bound is now $\sum_{h=1}^{H} k_h = O(k_H)$ instead of $O(k_H H)$, so $\mathsf{Space} = O\left((1/\varepsilon)\sqrt{\log(1/\delta)}\right)$. In particular, the dependence on n has been eliminated.

In the above space analysis, we have a slight subtlety—in order for a compactor to be effective, it must have a capacity that is at least 2, so we seemingly have $k_h = \max\{2, k_H(2/3)^{H-h}\}$ and the exponential decay does not technically hold for all h so

$$\mathit{Space} = \sum_{h=1}^{H} k_H = O(k_H + H) \ , \tag{10}$$

instead of our previous $O(k_H)$ upper bound. This is remedied by noticing that the lower $H'' = H - \lceil \log(k_H)/\log(3/2) \rceil$ levels of compactors of capacity 2 can be simulated by selecting one item uniformly from a continuous block of $2^{H''}$ items, which can be done in $O(1)$ memory using standard reservoir sampling.

2.5 Using a deterministic sketch for the final layers

The final stretch is to improve the dependence on δ in the space complexity from $\sqrt{\log(1/\delta)}$ to $\log\log(1/\delta)$. This will come from one final optimization over the later compactors, which take up most of the memory usage.

Consider truncating the compactor sequence at the h^*-th compactor before level H. This turns the original stream of n items into a stream of at most $\frac{n}{2^{H-h^*}} \leq k \cdot 2^{h^*}$ (using $H = \lfloor \log_2(n/k) \rfloor + 1$) items of weight 2^{H-h^*}. Since we know that w_h/k_h is exponentially increasing with common ratio $4/3$ and k_h is exponentially increasing with common ratio $3/2$, we conclude that

$$\mathsf{Error} \lesssim \left(\frac{2}{\sqrt{3}}\right)^{h^*} \frac{n}{k_H} \sqrt{\log \frac{1}{\delta}} \tag{11}$$

$$\mathsf{Space} \lesssim \left(\frac{2}{3}\right)^{h^*} k_H \tag{12}$$

Since we want an improved dependence on δ, one reasonable option is to feed the outgoing items into a deterministic sketch—this is another crucial improvement by [KLL16]. Suppose we channel the remaining elements into a GK sketch with error parameter ε'. Roughly speaking, increasing h^* decreases the space usage by a factor of α while increasing the number of outgoing items by $\mathit{poly}(\alpha)$, and the space complexity of the GK sketch depends logarithmically on the number of elements fed into the sketch, so we can "trade" the $\sqrt{\log(1/\delta)}$ factor in the memory cost of the MRL sketch due to $k_H = O((1/\varepsilon)\sqrt{\log(1/\delta)})$ for a $\log\log(1/\delta)$ factor in the memory cost of the GK sketch.

Explicitly, the total error and space constraints become (noting that $\leq k \cdot 2^{h^*}$ elements of weight 2^{H-h^*} are fed into the GK sketch):

$$\mathsf{Error} \leq \left(\frac{2}{\sqrt{3}}\right)^{h^*} \frac{n}{k_H}\sqrt{\log\frac{1}{\delta}} + \varepsilon' k_H \cdot 2^{h^*} \cdot 2^{H-h^*} \tag{13}$$

$$\leq \left(\frac{2}{\sqrt{3}}\right)^{h^*} \frac{n}{k_H}\sqrt{\log\frac{1}{\delta}} + \varepsilon' n \tag{14}$$

$$\mathsf{Space} \leq \left(\frac{2}{3}\right)^{h^*} k_H + \frac{1}{\varepsilon'}\log\left(\varepsilon' k_H 2^{h^*}\right) \tag{15}$$

To make $\mathsf{Error} \leq \varepsilon n$, we need

$$k_H \leq \frac{(2/\sqrt{3})^{h^*}}{\varepsilon - \varepsilon'}\sqrt{\log(1/\delta)} \tag{16}$$

For convenience, we assume that $\varepsilon' = \varepsilon/2$, so the resulting space complexity is

$$\mathsf{Space} \lesssim \frac{1}{\varepsilon}\left(\left(\frac{4}{3\sqrt{3}}\right)^{h^*}\sqrt{\log(1/\delta)} + h^* + \log\log(1/\delta)\right) \tag{17}$$

so the optimal choice for h^* is $O(\log\log(1/\delta))$—since we need h^* of this order to kill the $\sqrt{\log(1/\delta)}$ term and at the same time, the R.H.S is already at least $(1/\varepsilon)\log\log(1/\delta)$. This choice of h^* yields $\mathsf{Space} = O((1/\varepsilon)\log\log(1/\delta))$, which concludes our discussion of the optimal sketch.

Proof. (Theorem 1). Let our sketch be the concatenation of a truncated MRL sketch and a GK sketch with parameters described above. Note that the overall rank estimation function $\tilde{R}(x) = \tilde{R}_{MRL}(x) + 2^{H-h^*}\tilde{R}_{GK}(x)$ is non-decreasing since it is so for both the MRL and GK sketches. The overall scheme is randomized since the MRL sketch is. Finally, the algorithm is comparison-based since the MRL and GK sketches rely on maintaining a summary of elements that a query or an element of the stream is compared to. The space and error analyses have been performed above. □

3 Lower bound

To prove that the sketch in the previous section is asymptotically optimal. [KLL16] relies on the following lower bound for deterministic, comparison-based rank estimators by [HT10] which we will state without proof for simplicity.

Theorem 2 ([HT10])
Any **deterministic, comparison-based** algorithm that solves the single quantile ε-approximation problem for all streams of length $C(1/\varepsilon)^2\log^2(1/\varepsilon)$ for some sufficiently large universal constant C must store at least $c(1/\varepsilon)\log(1/\varepsilon)$ stream elements for some sufficiently small constant c.

We can easily turn this into the following theorem for randomized, comparison-based algorithms.

Theorem 3 ([KLL16])
Any **randomized, comparison-based** algorithm that solves the single quantile ε-approximation problem with probability $\geq 1-\delta$ must store at least $\Omega(1/\varepsilon\log\log(1/\delta))$ stream elements.

Proof. Suppose there exists a randomized algorithm $\mathcal{A}_R$ storing $o\left((1/\varepsilon)\log\log(1/\delta)\right)$ stream elements. Setting $\delta = \frac{1}{2n!}$ where n is the stream length to be set later, we see that $\mathcal{A}_R$ succeeds

on all $n!$ possible ordered streams with probability $\geq \frac{1}{2}$. Hence, there must be some random seed s received by $\mathcal{A}_R$ for which this occurs. We can then obtain a deterministic algorithm $\mathcal{A}_D$ that hardcodes the seed s into $\mathcal{A}_R$. Choosing $n = C(1/\varepsilon)^2 \log^2(1/\varepsilon)$ as in Theorem 2, $\mathcal{A}_D$ succeeds on all streams of length n while only storing $o\left((1/\varepsilon)\log(1/\varepsilon)\right)$ stream elements—leading to a contradiction. □

In particular, the discussed sketch of [KLL16], which is randomized and comparison-based, saturates the bound in Theorem 3 and is hence optimal.

4 Discussion

To summarize, we have covered the randomized, comparison-based single quantile approximation sketch by [KLL16] which is provably optimal in terms of space complexity. En route, much effort was devoted to reducing a factor of $\sqrt{\log(1/\delta)}$ to $\log\log(1/\delta)$, which is seemingly insignificant for most practical values of δ. However, this reduction in the complexity in δ is paramount when turning a single quantile approximation into an all quantile approximation defined below.

Definition 5

The **all quantile approximation** problem is about estimating the quantile of **all** items up to $\pm\varepsilon$ additive error **simultaneously**, with failure probability δ.

More explicitly, given $x_1, ..., x_n$ in streaming fashion, we produce an approximate (random) rank function $\tilde{R}$, with the following property:

With probability at least $1-\delta$, $\tilde{R}(x)$ *approximates the true rank* $R(x)$ *to within* $\pm\varepsilon n$ *(additively) for all* x.

Note that it suffices to approximate $O\left(1/\varepsilon\right)$ single quantile queries to solve the all quantile approximation problem. Thus, by a union bound, a $(\varepsilon, O(\varepsilon\delta))$-single quantile approximation algorithm solves a (ε, δ)-all quantile approximation problem—the transformation from $\delta \to O(\varepsilon\delta)$ implies that our single quantile complexity in δ is now important. The single quantile sketch by [KLL16] then yields an all quantile sketch with memory $O\left((1/\varepsilon)(\log\log(1/(\varepsilon\delta)))\right)$.

We remark here that an all-quantile sketch allows us to estimate the cumulative distribution function and hence the probability distribution of elements from a stream and is hence extremely useful in practice. The independence of the memory cost of the above all-quantile sketch relative to n makes it even more enticing. In the next section, we will experimentally verify the distributions reconstructed by this sketch.

The main limitation of [KLL16] concerns the issue of mergeability which we have glossed over. It turns out that it is unknown if the GK sketch is mergeable and [Aga+13] conjectures that it is not. However, all other parts of the sketch are mergeable. Compactors are evidently mergeable since we can just feed the elements within one compactor into another of the same level, starting from the lowest level. The reservoir sampler is also mergeable if one uses the sampling scheme described in [KLL16]. To recover mergeability, note that the main property of the GK sketch used in our analysis is its logarithmic dependence on the stream length and its determinism (so we don't have to worry explicitly about factors of δ in the GK sketch). Hence, we would obtain a sketch with the same space complexity but now mergeable if there exists a mergeable deterministic sketch with memory cost logarithmic in the stream length. Unfortunately, we do not know if such a sketch exists for general streams but the DDSketch by [MRL19] satisfies this property for stream elements drawn from distributions with sub-exponential and lighter tails.

5 Experimentation Verification

We implemented the described KLL-sketch by [KLL16], with the aid of the GK-sketch code found here: `https://github.com/DataDog/sketches-py/releases/tag/v0.1`. We assume that the length of the stream is known beforehand (i.e. it is a two-pass algorithm), simplifying the implementation since the number of compactors can be precomputed and resizing compactors is no

longer necessary. Our code can be found here:
https://drive.google.com/file/d/1tOZ6JFRg-V4c0KFQTFI9sXpCtJRixWyf/view?usp=sharing.

Table 1 reports our quantile errors for the all-quantile sketch with parameters $\varepsilon = 0.05$ a failure probability implicitly determined by $\delta' = \varepsilon\delta = 0.05 \times 0.05$ in the underlying single-quantile sketch, given streams drawn from different distributions. In all cases, we indeed obtain a ε-approximation for all queries.

Stream	Queries	Quantile Error
[1, ..., 1000000]	linspace(1, 1000000, 50)	0.0006636 ± 0.0004346
1000000 samples from $\mathcal{N}(0, 1)$	linspace(-3, 3, 50)	0.01591 ± 0.01125
1000000 samples from standard Cauchy	linspace(-10, 10, 50)	0.01746 ± 0.01008

Table 1: Experimental quantile query errors for all quantile sketch with parameters $\varepsilon = 0.05$ and $\varepsilon\delta = 0.5 \times 0.5$.

Next, we used the all quantile sketch to reconstruct an (approximate) histogram of the stream elements. As illustrated in Figure 2, we indeed obtain a histogram representing the distribution that the stream elements were drawn from. In plotting a histogram, the number of bins is chosen to be $\lfloor \frac{1}{\varepsilon} \rfloor$ since we are only applying the guarantee of the single quantile sketch for $\frac{1}{\varepsilon}$ values.

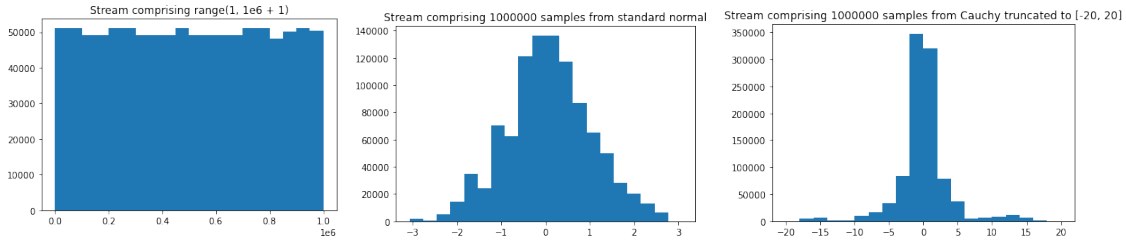

Figure 2: Approximate histograms reconstructed for three streams in Table 1.

Finally, using the CS368 library, we tracked the memory cost of the KLL-sketch versus the GK-sketch for a fixed $\varepsilon = 0.05$ (and $\delta' = 0.05 \times 0.05$ for the KLL-sketch) but for varying streams $[1, ..., n]$ of different lengths. As expected, the space complexity of the KLL-sketch is independent of $\log_{10} n$ while that of the GK-sketch scales linearly with $\log_{10} n$ (beyond some threshold, possibly caused certain peculiarities in the implementation we found online).

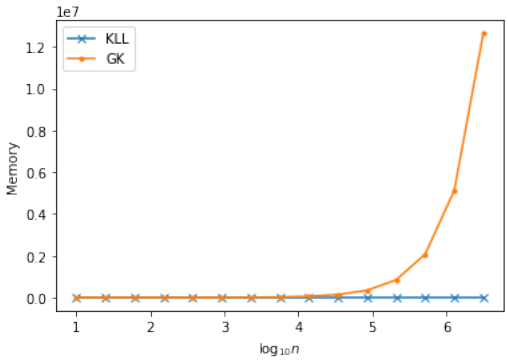

Figure 3: Memory cost with $\log_{10}$ number of stream elements.

References

[Aga+13] Pankaj K Agarwal et al. "Mergeable summaries". In: *ACM Transactions on Database Systems (TODS)* 38.4 (2013), pp. 1–28.

[CV20] Graham Cormode and Pavel Veselỳ. "A tight lower bound for comparison-based quantile summaries". In: *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2020, pp. 81–93.

[FO17] David Felber and Rafail Ostrovsky. "A Randomized Online Quantile Summary in $O((1/\varepsilon)\log(1/\varepsilon))$ Words". In: *Theory of Computing* 13.1 (2017), pp. 1–17.

[GK01] Michael Greenwald and Sanjeev Khanna. "Space-efficient online computation of quantile summaries". In: *ACM SIGMOD Record* 30.2 (2001), pp. 58–66.

[HT10] Regant Y. S. Hung and Hingfung F. Ting. "An $\Omega(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ Space Lower Bound for Finding ϵ-Approximate Quantiles in a Data Stream". In: *Frontiers in Algorithmics*. Ed. by Der-Tsai Lee, Danny Z. Chen, and Shi Ying. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 89–100. ISBN: 978-3-642-14553-7.

[KLL16] Zohar Karnin, Kevin Lang, and Edo Liberty. "Optimal quantile approximation in streams". In: *2016 ieee 57th annual symposium on foundations of computer science (focs)*. IEEE. 2016, pp. 71–78.

[MP80] J Ian Munro and Mike S Paterson. "Selection and sorting with limited storage". In: *Theoretical computer science* 12.3 (1980), pp. 315–323.

[MRL19] Charles Masson, Jee E. Rim, and Homin K. Lee. "DDSketch: A Fast and Fully-Mergeable Quantile Sketch with Relative-Error Guarantees". In: *Proc. VLDB Endow.* 12.12 (Aug. 2019), pp. 2195–2205. ISSN: 2150-8097. DOI: 10.14778/3352063.3352135. URL: https://doi.org/10.14778/3352063.3352135.

[MRL99] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G Lindsay. "Random sampling techniques for space efficient online computation of order statistics of large datasets". In: *ACM SIGMOD Record* 28.2 (1999), pp. 251–262.

[Wan+13] Lu Wang et al. "Quantiles over data streams: an experimental study". In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 2013, pp. 737–748.