Solution to HW5

---

1. For each interval $I = [i, j] \in [1, U]$, where $i \leq j$, it is specified by the left and right margins i and j, and can be uniquely mapped to a number in the range $[1, \frac{u(u+1)}{2}]$. In particular, we have the mapping:

$$f(i,j) = \begin{cases} j, & i = 1 \\ \frac{(i-1)(2u-i+2)}{2} + j - i + 1, & i > 1 \end{cases}$$

Therefore, when inserting an interval $[i, j]$ into set S, we first map it to number $f(i, j)$, and then apply the Bloom filter to this number as before. When querying the membership of interval $[i, j]$, we again map it to the corresponding number $f(i, j)$ and query this number. For this algorithm, the analysis of error rate is exactly the same with Bloom filter. Note that now the universe of numbers is of size $\frac{u(u+1)}{2}$.

2. Let R and $\hat{R}$ be the exact and approximate estimate to the range query $[i, j]$. We outline an approach which replaces the $(j-i+1)$ terms with at most $2\log_2 n$ intervals.

Consider the following $\log_2 n$ partitions of the set $\{1, 2, \cdots, n\}$:

$$\begin{aligned} \mathcal{P}_0 &= \{1, 2, 3, 4, 5, 6, 7, 8, \cdots, n\} \\ \mathcal{P}_1 &= \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \cdots\} \\ \mathcal{P}_2 &= \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \cdots\} \\ \mathcal{P}_3 &= \{\{1, 2, 3, 4, 5, 6, 7, 8\}, \cdots\} \\ &\vdots \\ \mathcal{P}_{\log n} &= \{\{1, 2, 3, 4, 5, 6, 7, 8, \cdots, n\}\} \end{aligned}$$

The key observation is that any interval $[i, j]$ can be broken into the union of $\alpha \leq 2\log_2 n$ of the above intervals $I_1, I_2, \cdots, I_\alpha$ (which are known as *dyadic ranges*), such that:

(a) $I_1 \cup I_2 \cup \cdots \cup I_\alpha = [i, j]$;

(b) $I_k \cap I_l = \emptyset,\ \forall k, l \in [1, 2, \cdots, \alpha], k \neq l$.

For example, if $n = 8$ then interval $[1, 7] = \{1, 2, 3, 4\} \cup \{5, 6\} \cup \{7\}$, where $\{1, 2, 3, 4\}$ comes from $\mathcal{P}_2$, $\{5, 6\}$ comes from $\mathcal{P}_1$ and $\{7\}$ comes from $\mathcal{P}_0$. In a general case, no more than 2 intervals will be selected from any single $\mathcal{P}_i$.

Now we are ready to explain the efficient technique for answering the range queries. In each $\mathcal{P}_i$, treat each interval as an item. Now build a Count-Min sketch on the items in each $\mathcal{P}_i$. When an update $F[i]++$ arrives, then for each $\mathcal{P}_i$, the item containing i is incremented by 1. Given a range query $[i, j]$, we select α dyadic intervals. For each dyadic interval, perform a point query on the Count-Min sketch corresponding to it. We report the sum of the result of these point queries as the estimate $\hat{R}$.

For the above algorithm, we have the following theorem.

Theorem 1. *For range query $[i, j]$, $R \leq \hat{R}$, and with probability at least $1 - \delta$, $\hat{R} \leq R + 2\epsilon \log n \sum_j F[j]$. The space occupied by the data structure is $O(\frac{\log n}{\epsilon} \log \frac{1}{\delta})$. The time taken to update the data structure or to answer the range query is $O(\log n \log \frac{1}{\delta})$.*

Proof. By applying the inequality of the point query Theorem we proved in the lecture, i.e., $F[i] \leq \tilde{F}[i]$, and due to the fact that each dyadic interval is treated as an item, we obtain the inequality $R \leq \hat{R}$ readily. Next, consider the interval estimator used to form $\hat{R}$. By linearity of the expectation of the errors of each point estimate (i.e., the interval estimate), and following the same proof procedure as the point query Theorem, the expectation of the additive error for $\hat{R}$ is $2 \log n \frac{\epsilon}{e} \sum_j F[j]$. Then applying the same Markov inequality argument as before, the probability that this additive error is more than $2\epsilon \log n \sum_j F[j]$ for any estimator is less than $\frac{1}{e}$. Combining all of the possible estimators, the final probability is then at most δ.

Since for point query the space used is $O(\frac{1}{\epsilon} \log \frac{1}{\delta})$ and the time per update or query is $O(\log \frac{1}{\delta})$, and now we have to do at most $2 \log n$ point queries, then for range query the time to compute the estimate or to make an update is $O(\log n \log \frac{1}{\delta})$, and the space used is $O(\frac{\log n}{\epsilon} \log \frac{1}{\delta})$. □