Complete all problems. You may not look at solutions of assignments from previous years. You can work in groups, but all solutions must be written up individually. If you get information from sources other than the course notes and slides, please cite the information, even if from Wikipedia or a textbook. Also, please provide a simple illustration of your solution, more than just an answer.

**Problem 1: Consistent Hashing. (20pt)**

You start a distributed content distribution company called *Iamaka*. You have a $n$ items of data you want to store in $m$ caches spread out over a network. Assume that the items are indexed by the integers $[n] = \{0, 1, 2, \ldots, n - 1\}$. A crucial problem is that the number of caches keeps changing over time.

One approach for load-balancing is to store the $i^{th}$ item in the cache $i \mod m$. This ensures a load of $\leq \mu$. But if we add a new cache, we would have to change our allocation to send item $i$ to cache $i \mod (m + 1)$. This would mean almost all $n$ items would have to move.

Consistent hashing gets around this problem as follows. You imagine the $n$ items to be arranged in a circle. For each of the $m$ caches, independently and uniformly choose a position on this circle. All items that lie between the position of this cache and the next cache on the circle (say in the counter-clockwise direction) are mapped to this cache.

(a) (Nothing to do here.) By symmetry, the expected length of the circle between two caches is $\mu := \frac{n}{m}$. Hence the expected load of a cache is $\mu$. Hence, when a cache gets added or removed, the expected number of items that have to be moved is $\mu$.

(b) Consider overlapping intervals of the circle of length $2\mu \ln m$ of the form $I_i := [\mu i, \mu(i + 2\ln m)]$

for $i = 0, 1, \ldots, m - 1$. Show that for any fixed $i$,

$$\Pr[\text{no caches fall in } I_i] \leq 1/m^2.$$  

(c) Argue that the previous part implies that with probability at least $1 - 1/m$, the load of every cache is at most $O(\mu \ln m)$.

(Hence the number of items moved on cache insertion/deletion is also $O(\mu \ln m)$.) We want a better load balancing. So for each of the $m$ caches, create $K := \ln m$ different virtual caches. Map each virtual cache as above randomly to the $n$-cycle, and assign items to these virtual caches. Finally, each real cache gets all the items assigned to all $K$ of its virtual caches.
(d) Now break the circle into $mK = m \ln m$ equal non-overlapping intervals of the form

$$J_i := \left[ \frac{K_i}{K}, \frac{K_{i+1}}{K} \right].$$

Show that for a fixed $i$, $\Pr[\text{no virtual cache falls in } J_i] \leq (1 - \frac{1}{mK})^{mK} \leq \frac{1}{e}$. (Hint: use $(1 + x) \leq e^x$ for all $x \in \mathbb{R}$.)

e) (Extra Credit.) Finally, for any (real) cache, show that the number of intervals $J_i$ containing items assigned to this cache is $O(K)$ with probability $1 - \frac{1}{m^2}$.

Problem 2: Striped Matrices. (25pt)

Consider the class of hash functions from Lecture #13, Section 3.1.2:

Take an $u \times m$ matrix $A$. Fill the first row $A_{1,*}$ and the first column $A_{*,1}$ with random bits. For any other entry $i, j$ for $i > 1$ and $j > 1$, define $A_{i,j} = A_{i-1,j-1}$. So all entries in each “northwest-southeast” diagonal in $A$ are the same.

Also pick a random $m$-bit vector $b \in \{0, 1\}^m$. For $x \in U = \{0, 1\}^u$, define

$$h(x) := Ax + b$$

where the vectors are added component-wise, modulo 2.

Hence the hash family $H$ consists of $2^{(u+m-1)+m}$ hash functions, one for each choice of $A$ and $b$. Show that $H$ is 2-wise independent.

Problem 3: Let’s Reach A Happy Median. (30pt)

We want to compute an “approximate median” of a set of $n$ distinct real numbers, which are in unsorted order. Say this set is $A = \{a_1, a_2, \ldots, a_n\}$. The rank of $a \in A$ is its position when $A$ is written in decreasing order: so the highest element has rank 1 and the lowest has rank $n$. The median has rank $\lfloor n/2 \rfloor$. Call $a \in A$ an $\varepsilon$-approximate median if its rank lies in the interval $[(1 - \varepsilon)\frac{n}{2}, (1 + \varepsilon)\frac{n}{2}]$.

Our algorithm is the following simple one.

Define $k = \frac{2}{\varepsilon^2} \log \frac{1}{\delta}$. Randomly sample $2k$ numbers from $A$ independently and with replacement, and call this multiset $S$. Output the median of the sample $S$ (using any linear-time median finding algorithm, say, or just sorting).

We want to prove that the algorithm outputs an $\varepsilon$-approximate median with probability at least $1 - 2\delta$.

(a) Call an element “high” if it has rank between 1 and $(1 - \varepsilon)\frac{n}{2}$. How many high elements do you expect to see in the sample $S$?
(b) For \( i \in \{1, 2, \ldots, 2k\} \), let \( H_i \) be the indicator random variable for whether the \( i\)th sampled element is high. Let \( H = \sum_{i=1}^{2k} H_i \). Use Hoeffding’s bound to show that

\[
\Pr[H \geq k] \leq \delta.
\]

(Hint: Use part (a) for \( \mathbb{E}[H] \).)

(c) Call an element “low” if it has rank between \((1 + \varepsilon)\frac{n}{2}\) and \(n\). By symmetry observe that the probability of sampling at least \(k\) low elements is at most \(\delta\). (You don’t need to do anything for this part.)

(d) Argue that if the sample \(S\) does not contain \(k\) high values, and it does not contain \(k\) low values, then the median of \(S\) must be an \(\varepsilon\)-approximate median.

*Aside:* you can also implement this algorithm in the streaming model. The only question is how to maintain a sample of \(2k\) items from the stream; you can do this by an idea called reservoir sampling.

**Problem 4: Many Almost-Orthogonal Vectors.** (25pt)

Call two vectors (of the same length) “near-orthogonal” if their inner product has small absolute value compared to the product of their lengths; in this problem we will show that while there are at most \(d\) orthogonal vectors in \(\mathbb{R}^d\), there can be exponentially more near-orthogonal vectors.

(a) In Lecture, we saw Hoeffding’s bound for independent random variables that took values in \([0, 1]\). Use that to prove the following concentration bound for \(\{-1, 1\}\)-valued random variables. Let \(Y_1, Y_2, \ldots, Y_n\) be independent and identical \(\{-1, +1\}\)-valued random variables, each \(Y_i = 1\) with probability \(1/2\) and \(Y_i = -1\) with probability \(1/2\). Let \(Y = \sum_{i=1}^{n} Y_i\). Prove that for \(\lambda \leq n\),

\[
\Pr[|Y| \geq \lambda] \leq 2 \exp\left(-\frac{\lambda^2}{6n}\right)
\]

(b) Let \(x = (x_1, x_2, \ldots, x_d)\) and \(y = (y_1, y_2, \ldots, y_d)\) be two independently and uniformly chosen vectors in \((-1, 1)^d\). (This means each “bit” \(x_i\) and \(y_i\) in each vector is independently and uniformly chosen from \((-1, 1)\).) Recall the inner product \(\langle x, y \rangle := \sum_{i=1}^{d} x_i y_i\). Show that

\[
\Pr[|\langle x, y \rangle| \geq \epsilon d] \leq 2 \exp\left(-\epsilon^2 d / 6\right)
\]

(c) Given a parameter \(\epsilon > 0\), two vectors \(\vec{x}, \vec{y} \in \mathbb{R}^d\) with \(\|\vec{x}\| = \|\vec{y}\|\) are called \(\epsilon\)-orthogonal if

\[
|\langle \vec{x}, \vec{y} \rangle| \leq \epsilon \cdot \|\vec{x}\| \cdot \|\vec{y}\|.
\]

Recall that \(\|\vec{x}\| = \sqrt{\sum_{i=1}^{d} x_i^2}\) is the Euclidean norm.

Pick a set of \(N = \exp(\Omega(\epsilon^2 d))\) random vectors in \((-1, 1)^d\), where each vector is independently and uniformly chosen from \((-1, 1)^d\). Use part (b) and a union bound to show that with probability half, all pairs of vectors in this set are \(\epsilon\)-orthogonal.