Problem 1

Suppose that we wish use Bloom Filtering to store a set of a million items. What is the minimum amount of memory (in terms of bits) do we need in order to have a 10^6 probability of false positives? You can use the estimate for false positives obtained in class.

Probability that a specific bit remains zero after n elements have been introduced into the set of size m using k hashes:

$$p \simeq (1 - \frac{1}{m})^{nk} \simeq e^{-kn/m}$$

Expected number of bits that are 0 is $m \cdot e^{-kn/m}$. Then,

$$P(False\ Positive) = (1-p)^k \simeq (1 - e^{-kn/m})^k$$

For $P(False\ Positive) = \delta$:

$$k \simeq log_2\Big(\frac{1}{\delta}\Big) = log_2\Big(\frac{1}{10^{-6}}\Big) \simeq 19.93$$

Then,

$$m = \frac{kn}{ln2} \simeq 28755175 \simeq 29Mb$$

Problem 2

Suppose that we run the Misra-Gries algorithm on a stream of length m with $k-1$ counters. Let $\hat{f}_x$ be the count returned by the algorithm for a key x. If there is no counter associated with x, we take $\hat{f}_x$ to be 0. Let $\hat{m}$ be the sum of all counters at the end of the algorithm. Prove that for any element x, $\hat{f}_x$ provides a crude estimate of the frequency f_x of x in the following sense:

$$f_x - \frac{(m - \hat{m})}{k} \leq \hat{f}_x \leq f_x$$

In the best case scenario, the number of distinct elements in the stream was less than $k-1$. Then, the counter associated with x will show the true count of x i.e. f_x. Hence the maximum $\hat{f}_x$ can be is f_x:

$$\hat{f}_x \leq f_x$$

In the worst case scenario, all deletions decremented the count for x. Then, given that x still has a counter assigned to it at the end, the count will be equal to the true count of x minus the total number of deletions. We know that the total possible number of deletions in Misra-Gries is $\frac{m}{k}$, but the actual number of deletions for a given instance is $\frac{m-\hat{m}}{k}$. Then,

$$\hat{f}_x \geq f_x - \frac{m - \hat{m}}{k}$$

Hence,

$$f_x - \frac{(m - \hat{m})}{k} \leq \hat{f}_x \leq f_x$$

Problem 3

Suppose that given a stream of length m, we want to output all elements with frequency more than $\frac{m}{k}$ but we do not want to output any element with frequency less than $(1-\epsilon)\frac{m}{k}$ for some given $\epsilon \in (0, 1]$. How would you use the Misra-Gries algorithm to do this in one pass? How many counters (in terms of k and ϵ) do you need?

Using the result from Problem 2, the worst (lowest) estimate we could make is $f_x - \frac{m}{c}$, where c is the number of counters + 1. It is required that $f_x - (1-\epsilon)\frac{m}{k} \geq 0$ for some ϵ and k.

$$\frac{m}{c} = (1 - \epsilon)\frac{m}{k}$$

Then,

$$c = \frac{k}{1 - \epsilon}$$

Problem 4

Let A be an $m \times u$ matrix with entries in $\{0, 1\}$ s.t. for each $i, j > 1, A_{i,j} = A_{i1,j1}$ and let b be a vector in $\{0, 1\}^m$. For any choice of such A and b, define the function $h_{A,b}(x) = Ax + b(mod2)$ and let $\mathcal{H}$ be the set of all such functions. Prove that $\mathcal{H}$ is 2-universal.

A 2-universal hash family $\mathcal{H}$ is such that: for any 2 distinct keys $x_1, x_2 \in \mathcal{U}$ and any 2 values $a_1, a_2 in[\mathcal{M}]$:

$$P_{h \sim \mathcal{H}}[h(x_1) = a_1 \wedge h(x_2) = a_2] \leq \frac{1}{\mathcal{M}^2}$$

Consider $\forall x \neq y$

$$P[Ax + b = \alpha \; and \; Ay + b = \beta] \leq \frac{1}{2^{2m}}$$

$$z = x - y \neq 0: \quad Az = Ax + b - (Ay + b) = Ax - Ay = A(x - y) = \alpha + \beta = \gamma$$

$$\begin{bmatrix} a_0 & a_1 & \dots & a_n \\ a_{-1} & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \\ a_{-(m-1)} & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ \vdots \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{m-1} \end{bmatrix}$$

Assuming that $z_0 = 1$, and ignoring b, observe that since z and γ are fixed (determined by fixed vectors x, y and α, β respectively), the first column of A has to be such that the above equation holds ($A[0] = \Lambda$). The values of other columns in A can be ignored since, they can anything and $A[0]$ still maintains the deciding power. The probability that this is the case is:

$$P[A[0] = \Lambda] = P[a_0 = \Lambda[0] \wedge a_1 = \Lambda[1] \wedge ... \wedge a_{m-1} = \Lambda[m-1]]$$

Since each element in the first column of A is picked independently at random,

$$P[A[0] = \Lambda] = P[a_0 = \Lambda[0]] \times P[a_1 = \Lambda[1]] \times ... \times P[a_{m-1} = \Lambda[m-1]] = \left(\frac{1}{2}\right)^m$$

This fixes A, so let us consider b in the following:

$$\begin{bmatrix} a_0 & a_1 & \dots & a_n \\ a_{-1} & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \\ a_{-(m-1)} & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{m-1} \end{bmatrix}$$

A is fixed as above and the values in b can change the product of A and x. So now b has to be picked such that $Ax + b = \alpha$ which happens for $b = \rho$. The probability of this is:

$$P[b = \rho] = P[b_0 = \rho[0] \wedge b_1 = \rho[1] \wedge ... \wedge b_{m-1} = \rho[m-1]]$$

Since each element in b is picked independently at random,

$$P[b = \rho] = P[b_0 = \rho[0]] \times P[b_1 = \rho[1]] \times ... \times P[b_{m-1} = \rho[m-1]] = \left(\frac{1}{2}\right)^m$$

Now,

$$P_{h \sim \mathcal{H}}[h(x_1) = a_1 \wedge h(x_2) = a_2] \leq P[b = \rho \wedge A[0] = \Lambda] = \frac{1}{2^{2m}}$$