TDA251/DIT280, Period 2, 2016: Algorithms Advanced Course.

Solution hints to the exam problems.

Problem 1

(a): Define a random variable h with $h = 1$ for hit, and $h = 0$ for no hit. Now $E[h] = 1/\pi$ and linearity of expectation yield $E[k] = n/\pi$, thus $E[k/n] = 1/\pi$.
(b): The expected number of trials until a hit is $1/(1/\pi) = \pi$. Since we wait for k hits, linearity of expectation yields $E[n] = k\pi$, thus $E[n/k] = \pi$.
(c): If at most $d-1$ decimals are correct, then $\delta \geq \pi/10^d$. Trivially, $\mu = n/\pi$. Hence $e^{-\delta^2\mu/3} \leq e^{-(\pi/3)n10^{-2d}}$.
(d): The random variable k is the sum of n independent $0,1$-variables, hence the assumptions of the Chernoff bound are fulfilled.
(e): Since $10^{2(d+1)} = 100 \cdot 10^{2d}$, we must multiply n by 100 to keep the same confidence.
(f): Obviously, it goes down exponentially.

Problem 2

(a): We want to estimate d, the number of distinct numbers in the stream. Intutively, the minimum of d random numbers in the interval $[0,1]$ should be around $1/d$. Since our hash function behaves just as a random function, imagine we pick $d+1$ random numbers from the interval $[0,1]$ and consider the chance that $(d+1)$-th element is the smallest. By symmetry this chance is $1/(d+1)$. Or in other words, the expected value of a minimum of $(d+1)$ independent uniform variables having values in the interval $[0,1]$ is $1/(d+1)$. That is, the expected value $E[Z] = 1/(d+1)$. One can plug in the reasoning into the definition of expectation and the calculations should end up with the same.
(b): Unbiased estimator is thus $(1/Z) - 1$.

Problem 3

(a): Create a flow network with vertices for each professor and each committee. The vertex for each professor is connected to the committees he agrees to serve on. Capacity on edge from source to professor j is s_j, from committee i to sink is k_i, and ∞ on all other edges.

Max flow is sum of k_i, *iff* such committees can be formed. One should also reason about integrality of the flow (we want to avoid half professors).

Problem 4

(a): Let K be the set of k bits set to zero by the deletion (formally, K is the index set of bit positions set to one, and B is the set of all positions). An error is caused if any of the k bits of an item x is in K. An item is unaffected by the deletion if all its bits are in $B \setminus K$. The probability of picking such a bit is $\frac{b-k}{b} = 1 - \frac{k}{b}$ (this is assuming that all the bits are distinct), and consequently the probability of picking k bits for all n elements is

$$(1 - \frac{k}{b})^{kn}.$$

(b): Let X be a $B(n,p)$-Binomial random variable with n trials and success probability (probability that an item is affected by the deletion) $p = 1 - (1 - \frac{k}{b})^k$. The answer is $P\{X = d\}$.

(c): $E[X] = n \cdot p$.

(d): Note that d deletions can effect from k (assuming that the k hash functions differ) up to $k \cdot d$ bits. If we assume the worst-case, that is that d deletions set $k \cdot d$ bits to zero, then we are asking for the expectation of a $B(n,p)$-Binomial random variable with n trials and success probability (probability that an item is affected by the deletion) $p = 1 - (1 - \frac{k \cdot d}{b})^k$.

For a more detailed analysis one can use the law of total expectation,

$$E[X] = \sum_i E[X|A_i]P(A_i),$$

with $A_i, i = k, \ldots, k \cdot d$ the events of the d deletions affecting $k, \ldots, k \cdot d$ bits. The $E[X|A_i]$ is again obtained by viewing this as Binomial random variable. The probability of exactly m bits can be obtained from viewing this as a coupon collecting problem. It is equal to the probability of obtaining exactly m coupon types when collecting $k \cdot d$ coupons sampled uniformly from b coupon types.

Problem 5

(a): Space for prefix array is $|\Sigma|^k$. Let $d(v)$ be the out-degree of a node v and $depth(v)$ its node depth (as usual depth of root is zero). The total memory needed for the partial tree up to depth m is given by

$$t(m) = \sum_{v:depth(v) \leq m} 2 + 4d(v),$$

The criterion is to find $k = \arg\max_{0 \leq m} |\Sigma|^m < t(m)$.

(b): Use a BFS traversal to count nodes up to depth m and their descendants. Every time a level is done, evaluate the criterion. Use a DFS up to depth k to set the pointers in the prefix array appropriately.