CS 466/666 Algorithm Design and Analysis **Last Updated:** May 27, 2019
Homework 2 **Due Date: June 10 (11pm), 2019**

You are allowed to discuss with others but not allowed to use any references except the course notes and the books "Probability and Computing" and "Randomized Algorithms". Please list your collaborators for each question. This will not affect your marks. In any case, you must write your own solutions.

The full mark is 50. This homework is counted 8% of the course.

---

1. **Coupon Collectors**

   (10 marks) Now we know that it requires $\Theta(n \log n)$ coupons to collect n different types of coupons, with at least one coupon per type. You wonder whether it would be more efficient if a group of k people cooperate, such that each person buys cn coupons and as a group they have at least k coupons per type (so that everyone gets a complete set of coupons).

   Prove the best bound you can on k to ensure that, with probability at least 0.9, each person only needs to buy at most $10n$ coupons. You will get full marks if k is of the correct order in terms of n.

2. **Balls and bins**

   (10 marks) This problem models a simple distributed system wherein agents contend for resources but "back off" in the face of contention. Balls represent agents, and bins represent resources. The system evolves over rounds. Every round, balls are thrown independently and uniformly at random into n bins. Any ball that lands in a bin by itself is served and removed from consideration. The remaining balls are thrown again in the next round. We being with n balls in the first round, and we finish when every ball is served.

   (a) If there are b balls at the start of a round, what is the expected number of balls at the start of the next round?

   (b) Suppose that every round the number of balls served was exactly the expected number of balls to be served. Show that all the balls would be served in $O(\log \log n)$ rounds.

3. **k-wise independence**

   (10 marks) Suppose that we are given m vectors $v_1, \ldots, v_m \in \{0,1\}^l$ such that any k of the vectors are linearly independent modulo 2. Let $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,l})$ for $1 \leq i \leq m$. Let u be chosen uniformly at random from $\{0,1\}^l$. Let $X_i = (\sum_{j=1}^{l} v_{i,j} u_j) \mod 2$. Show that the X_i are uniform, k-wise independent random bits.

4. **Maximum load**

   (10 marks) We have shown that the maximum load when n items are hashed into n bins using a hash function chosen from a 2-universal family of hash functions is at most $\sqrt{2n}$ with probability at least $1/2$. Generalize this argument to k-universal hash functions. That is, find a value (in terms of k and n) such that the probability that the maximum load is larger than that value is at most $1/2$. Then, find the smallest value of k such that the maximum load is at most $3 \ln n / \ln \ln n$ with probability at least $1/2$ when choosing a random hash function from a k-universal family. You will get full marks if k is of the correct order in terms of n.

5. **Finding approximate median in sublinear time**

   (10 marks) We would like to find an approximate median of n distinct numbers in sublinear time. To do so, we sample $m \ll n$ numbers, find the median c of these m numbers, and report c as the approximate median of the n numbers. Let the sorted list of the n numbers be $\{x_1, x_2, \ldots, x_n\}$ and so the true median is $x_{n/2}$. The approximate median is said to be a $\pm k$-approximation if $c \in [x_{\frac{n}{2}-k}, x_{\frac{n}{2}+k}]$. Suppose we want the algorithm to succeed to find a $\pm k$-approximation with probability at least 0.9999. What is the tradeoff between m and k? How large should we set m if we want $k \leq \epsilon n$ for some small constant ϵ? How large should we set m if we want $k \leq n^{1-\epsilon}$ for $\epsilon \leq 1/2$? You will get full marks if m is of the correct order in terms of n and ϵ.