E0 206 : Homework 4

Due date : 3/12/20

Instructions

- All problems carry equal weight.
- You are forbidden from consulting the internet. You are strongly encouraged to work on the problems on your own.
- You may discuss these problems with your group (at most 3 people including you). However, you must write your own solutions and list your collaborators for each problem.

1. Consider the following algorithm for multi-armed bandits when we have only two arms:

   (a) Some N is fixed at the beginning of the algorithm.

   (b) In the first N rounds (we call it exploration phase), the choice of arms does not depend on the observed rewards.

   (c) In all remaining rounds, the algorithm only uses rewards observed during the exploration phase.

   Prove that such an algorithm must have expected regret $\mathbb{E}[R(T)] \geq \Omega(T^{2/3})$ in the worst case.

   (*Hint.* Regret is a sum of regret from exploration, and regret from exploitation. For "regret from exploration", we can use two instances: $(\mu_1, \mu_2) = (1, 0)$ and $(\mu_1, \mu_2) = (0, 1)$. For "regret from exploitation" we invoke the hardness result studied in the class.)

2. Consider the following family of hash functions $\mathcal{H}$:

$$h(x) : (a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_0) \bmod p,$$

   where a_i's are uniformly chosen at random from the set $\{0, 1, \ldots, p-1\}$ for some prime p. Show that $\mathcal{H}$ is a k-independent universal hash family.

3. Consider a variant of the traditional streaming model where we are interested in computation of a function of only the last N entries (instead of from the beginning of the stream). We are interested in the following basic question: Given a stream of data elements, consisting of 0's and 1's, the goal is to maintain at every time instance the count of the number of 1's in the last N elements using s bits of space. If we use $s = N$, we can trivially get the exact answer. Our goal is to find a good approximate solution using $o(N)$ memory.

   Design an algorithm that returns a solution within $\pm\epsilon N$ additive error using only $\frac{1}{\epsilon}O(\log^2 N)$ bits of memory, i.e. if the exact answer is C, we need to return a solution in $[C - \epsilon N, C + \epsilon N]$.
   (*Optional:* Using $\frac{1}{\epsilon}O(\log^2 N)$ bits of space, can you find an answer in $[(1-\epsilon)C, (1+\epsilon)C]$ where C is the exact answer?)

4. Consider the FREQUENT problem where we need to return all elements that appear more than m/k number of times in a stream, where m is the total number items in the stream and $k \in \{2, 3, 4, \dots\}$ is a constant. Assume that every deterministic oneway communication protocol that computes the INDEX function IDX_N (i.e., $IDX_N(x, y) = x_y$) uses at least $\Omega(N)$ bits of communication, in the worst case. Using this fact, show that any one pass deterministic algorithm for FREQUENT will require $\Omega(\min\{m, n\})$ space.

More optional practice problems: (will not be graded or provided with solutions)
AC lecture notes (2020): 1-1, 1-2, 1-3, 2-1, 2-2, 3-1, 3-2, 5-1, 5-4, 7-1, 7-2, 18-1, 18-2, 18-3.
Sen-Kumar Book: 16.5, 16.7, 16.9, 16.10, 16.12, 16.13.