COMP 5711: Advanced Algorithms
Fall 2020 Final Exam (Part 1)

Note 1: For any positive integer n, we denote $[n] = \{1, 2, \ldots, n\}$.

Note 2: For questions with multiple steps, if you don't know how to prove earlier steps, you can still try to prove later steps, assuming earlier steps have been proven.

1. (6 pts) Suppose an data structure is able to handle a sequence of n operations for any n. For each of the following statements, determine if it is true or false. You don't need to justify your answers.
   (a) If the cost of each operation is c, then the amortized cost of the operation is c.
   (b) Let the total cost of n operations be $f(n)$. If $\lim_{n\to\infty} f(n)/n = c$, then the amortized cost of the operation on the initial state is at most c.
   (c) Suppose there are two types of operations op_1 and op_2. When the sequence consists of solely op_1, its amortized cost is c_1; when the sequence consists of solely op_2, its amortized cost is c_2. Then the amortized cost of each operation is at most $\max\{c_1, c_2\}$, when the sequence consists of both op_1 and op_2.

2. (15 pts) Suppose we have an array of elements $\{A_i\}_{i=1}^n$, and an array of bits $\{M_i\}_{i=1}^n$ where $M_1 = 1$ and $M_i \in \{0, 1\}$ for $i > 1$. The bits in M define a partitioning of A into a number of segments. More precisely, a *segment* is the sub-sequence of elements $A_i, A_{i+1}, \cdots, A_{j-1}$ where $M_i = 1$, $M_j = 1$, and $M_k = 0$ for $k = i+1, \cdots, j-1$. The goal is to compute the prefix sums for each segment, i.e., output an array B, such that

$$B_j = A_i + \cdots + A_j, \text{where } i = \max\{k \mid M_k = 1, k \leq j\}.$$

   (a) Define a binary operator $\oplus$ between two pairs (u, b_1) and (v, b_2) where b_1, b_2 are two bits, as follows.

$$(u, b_1) \oplus (v, b_2) = \begin{cases} (u+v, b_1), & \text{if } b_2 = 0; \\ (v, b_2), & \text{if } b_2 = 1. \end{cases}$$

   Prove that $\oplus$ is associative.
   (b) Design a parallel algorithm to compute the prefix sums for all segments with $O(n)$ work and $O(\log n)$ time, under the EREW model with unlimited processors.

3. (25 pts) Consider the balls and bins problem. Suppose we throw n balls into n bins uniformly. Suppose the i-th ball falls into the A_i-th bin. Let $X_j = |\{i \mid A_i = j\}|$ be the number of balls in the i-th bin, and $X = \max_{j\in[n]} X_j$. You may assume n is large enough.
   (a) Prove that if $\{A_i\}_{i=1}^n$ are mutually independent, then $\Pr(X \geq \ln(n)) = o(1/n^c)$ for any positive constant c.
   (b) Prove that if $\{A_i\}_{i=1}^n$ are pairwise independent, then $\Pr(X \geq \sqrt{2n} + 1) \leq 1/2$.
   (c) Below, we will show that the bound in (b) is tight, i.e., there is a sequence $\{A_i\}_{i=1}^n$ that are pairwise independent, such that X is very likely to be $\Omega(\sqrt{n})$. Consider the following random process. First we choose a uniformly random number $K \in [n]$ and a random permutation π of $[n-1]$. Then for each $i \in [n-1]$, we independently set $A_i = K$ with probability $1/\sqrt{n}$, and $A_i = K + \pi(i)$ otherwise (if $K + \pi(i) > n$, we set $A_i = K + \pi(i) - n$ instead). Finally, we draw A_n uniformly at random from $[n]$. Show that each A_i is uniformly distributed over $[n]$, i.e., $\Pr(A_i = j) = 1/n$ for any $i \in [n], j \in [n]$.

(d) Let $\{A_i\}_{i=1}^n$ be defined as in (c). They are no longer mutually independent, but are still pairwise independent (you do not need to prove this). Show that under this sequence $\{A_i\}_{i=1}^n$, $\Pr(X \le \sqrt{n}/2) = o(1)$. [Hint: Since $X \ge X_K$, it's sufficient to bound $\Pr(X_K \le \sqrt{n}/2)$.]