1. (24%) Let $\mathcal{A}$ be a randomized algorithm whose output is either "Yes" or "No." Everytime we run $\mathcal{A}$ on an input x, with probability

$$\begin{cases} 2/3 & \mathcal{A}(x) \text{ is correct} \\ 1/3 & \mathcal{A}(x) \text{ is incorrect} \end{cases}$$

where the correctness of $\mathcal{A}(x)$ only relies on the random seed. You may assume that the outcomes of multiple execution of $\mathcal{A}(x)$ are mutually independent. Give an algorithm that can decide whether x is a "Yes"-instance or an "No"-instance with probability at least $1 - \varepsilon$, analyze the running time, and prove its correctness. Note that your algorithm has running time as a function of ε. The faster your algorithm runs, the more credit you get.

2. (20%) Given an input stream of n real numbers, devise an $O(1/c)$-pass $O(n^{1/c} \log n)$-space deterministic algorithm for every constant integer $c \geq 1$ to compute the k-th smallest number in the input where k is some integer in $[1, n]$.

3. (10%+10%) Let $X_1, X_2, \ldots, X_k$ be k random variables.

   (a) For each integer $k \geq 2$, prove or disprove there exist $X_1, X_2, \ldots, X_k$ that are pairwise independent but not mutually independent.

   (b) Let $X = \sum_{i=1}^{k} X_i$ where $X_1, X_2, \ldots, X_k$ are pairwise independent. Prove that

$$\mathrm{Var}[X] = \sum_{i=1}^{k} \mathrm{Var}[X_i].$$

4. (10%+10%) Let $G = (V, E)$ be a simple undirected graph in which each edge e has weight $\omega(e)$. Let MSF(G) be the set of edges in the minimum spanning forest of G and let $\mu(G) = \sum_{e \in \text{MSF}(G)} \omega(e)$.

   (a) Prove that $\mu(G) = \mu\left(G' = (V, (E \setminus E_r) \cup F_r)\right)$ where E_r is any subset of E and $F_r = \text{MSF}(G_r = (V, E_r))$.

   (b) Devise an $O((m+n)\log n)$-time semi-streaming algorithm to compute the minimum spanning forest for any n-node m-edge undirected graph. That is, m edges are given one by one as the input, and the working space is restricted to $O(n \operatorname{polylog} n)$.

5. (16%) Let G be an arbitrary n-node m-edge undirected graph. Give a deterministic algorithm that can return n different cuts of G so that each cut partition the node set into two and the number of crossing edges is at least $(1-\varepsilon)m/2$ for an arbitrary small constant $\varepsilon > 0$.