1. The min-cut algorithm we described assumes an efficient method for randomly sampling the edges of the contracted graphs during execution. The only edges removed during a contraction are the contracted edge itself plus any others that share the same endpoints and would lead to a self-loop. All other edges “remain” in the graph, i.e. there is a correspondence between each edge in any intermediate step of the algorithm and one of the original edges in the graph. Suppose the graph initially has $E$ edges and $V$ vertices for analysis.

(a) Show that the sequence of edges contracted during the algorithm corresponds to a random sample drawn without replacement from the original edges of the graph.

(b) During contraction, assume vertices are merged into “meta-vertices” consisting of a union of the original vertices contracted to that meta-vertex. Suggest an efficient data structure for this step, and for the operation of testing whether the two endpoints of an edge belong to the same meta-vertex.

(c) Using these two ideas, suggest an efficient approach to making $n - 2$ contractions that chooses the sequence of edges to attempt to contract before the algorithm starts contracting them. What is the complexity of contracting the graph down to 2 vertices? You should be able to achieve $O(E\alpha(V))$, where $\alpha(.)$ is the inverse of Ackerman’s function (a very slowly growing function whose value is $\leq 5$ for any practical input).

2. A permutation of $n$ elements can be represented as a function $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ which is one-to-one and onto. A fixed point of a permutation $\pi$ is a value $i$ such that $\pi(i) = i$. Find the expected number of fixed points in a permutation chosen uniformly at random from the set of all permutations of $n$ elements.

3. Reservoir sampling: Suppose you receive elements from a live data stream one at a time $a_1, \ldots, a_n$. Your goal is to maintain a random sample of that stream at every time step with fixed memory.

(a) Show that you can maintain a single random sample $b$ this way: start with $b$ set to the first value in the stream $a_1$. Then when $a_k$ appears, you randomly swap it with $b$ with probability $1/k$.

(b) Generalize this method to maintain a uniformly-chosen random subset of $m$ elements without replacement $b_1, \ldots, b_m$ using only $m$ memory locations plus the value of $k$. Assume $k \geq m$. You can approximate any statistical property from such a sample.

(c) Generalize again to maintain an $m$-element uniformly-chosen random sample with replacement (very easy).

4. Prove that $E[X^k] \geq E[X]^k$.

5. Show that $E[\min(X, Y)] + E[\max(X, Y)] = E[X] + E[Y]$. 