Advanced Algorithms Deadline: December 21, 2018, 11:59 pm

Graded Homework 3

Lecturer: Mohsen Ghaffari

Note: *For all problems, besides providing the algorithm, you should present a complete analysis that proves the claimed performance, e.g., approximation guarantee or running time. Your solutions must by typeset in LaTeX. Please email the pdf to Sebastian Brandt (brandts@ethz.ch) by the indicated deadline. Please include the phrase "Advanced Algorithms 2018: Graded Homework 3" in the title of your email, to help us in finding the submissions.*

1 Problem 1 (25 points)

We have a stream of length m of numbers $x_1, x_2, \ldots, x_m$, each of which is a value in $\{1, ..., n\}$. Let $f_i = |\{j | 1 \leq j \leq m \textit{ and } x_j = i\}|$, that is, the frequency of element i in the stream. Devise a streaming algorithm with $O(\log(n+m))$ bits of space that provides an unbiased estimator Z of the function $\sum_{i=1}^{n}(f_i)^{3/2}$. That is, we should have $\mathbb{E}[Z] = \sum_{i=1}^{n}(f_i)^{3/2}$.

2 Problem 2 (25 points)

Consider a graph in the dynamic streaming setting, i.e., a dynamic stream of edge arrivals and departures, on a set V of n vertices. Let $G = (V, E)$ denote the graph at the end of this stream. Devise a streaming algorithm that uses $nk \cdot (\log n)^{O(1)}$ bits of space and, with high probability, computes a collection $F_1 \cup F_2 \cup \ldots \cup F_k$ of forests such that for each $i \in \{1, 2, \ldots, k\}$, the forest F_i is a maximal forest in the graph $G \setminus (\cup_{j=1}^{i-1} F_j)$—this is the spanning subgraph of G resulting from removing all edges in any of F_1 to F_{i-1}. Note that you could compute these forests easily if the stream was repeated k times, because then computing each forest would follow directly from lecture 9. The question asks you to compute all these forests when there is no repetition of the stream.

3 Problem 3 (25 points)

Present a <u>deterministic</u> algorithm that computes a $(2k-1)$ spanner with at most $O(kn^{1+1/k} \log n)$ edges, for any given integer $k \in [1, \log n]$, in $m \cdot \text{poly}(\log n)$ time.

Hint: Start from the <u>randomized</u> algorithm discussed in Problem 4 of Problem Set 10 and think about how to do the marking deterministically.

4 Problem 4 (25 points)

Consider a graph in the dynamic streaming setting and let $G = (V, E)$ denote the graph at the end of this stream. Devise a randomized streaming algorithm that uses $n \cdot (\frac{\log n}{\epsilon})^{O(1)}$ bits of space and, with high probability, computes a $1 + \epsilon$ approximation of the minimum cut size λ of graph G. That is, the algorithm should output a cut $(S, V \setminus S)$ for a subset $S \subset V$ such that the number of edges in G with exactly one endpoint in S is within a $1 + \epsilon$ factor of the minimum. You can assume the answer to problem 2 as a given blackbox streaming algorithm.

Hint: Recall Karger's result about how cut sizes change when we sample each edge with probability $p = \Omega(\log n/\lambda)$. We emphasize that λ is not known a priori, and your algorithm should provide a $1 + \epsilon$ approximation of it.