Name: _____

CSE 4502/5717 Big Data Analytics

Exam I; March 6, 2019

Note: You are supposed to give proofs to the time and processor bounds of your algorithms. Read the questions carefully before attempting to solve them.

1. (25 points) We are given two sets A and B with n elements each in the form of arrays. We are also given that A is in sorted order and B may not be in sorted order. In addition, $|A \cap B| = \frac{n}{4}$. Present an $\widetilde{O}(\log^2 n)$ time Las Vegas algorithm to output an element that is common to A and B.

2. (25 points) Input is a sequence X of n keys where each key is an arbitrary real number. It is known that the total number of distinct keys in X is only a constant. Present an algorithm to sort X that takes $O(\log n)$ time using $\frac{n}{\log n}$ arbitrary CRCW PRAM processors.

3. (25 points) $X = k_1, k_2, \ldots, k_n$ is a sequence of keys residing in a disk. The problem is to sort X. It is known that $M = \Theta(B^2)$ and $n >> M$, where M is the size of the main memory and B is the block size. Each key of X is an integer in the range $[1, B]$. Show how to sort X in two passes through the data (i.e., using $2\frac{n}{B}$ read I/O operations).

4. (25 points) The k^{th} *quantiles* of an n-element sequence are the $(k-1)$ elements from the sequence that divide the sorted sequence into k equal-sized subsequences.

   Input is a sequence X of n arbitrary real numbers residing in a disk (with $n >> M$). Present an algorithm to find the k^{th} quantiles of X that makes $O\left(\frac{n}{B}\log k\right)$ I/O operations, where M is the main memory size and B is the block size. (Assume that $k = 2^q$ for some integer q).

CSE 4502/5717 Big Data Analytics; Spring 2019
Exam 1 Helpsheet

1. **Randomized algorithms.** A Monte Carlo algorithm runs for a prespecified amount of time and its output is correct with high probability. By high probability we mean a probability of $\geq 1 - n^{-\alpha}$, for any constant α. A Las Vegas algorithm always outputs the correct answer and its run time is a random variable. We say the run time of a Las Vegas algorithm is $\widetilde{O}(f(n))$ if the run time is $\leq c\alpha f(n)$ for all $n \geq n_0$ with probability $\geq (1 - n^{-\alpha})$ for some constants c and n_0.

   Chernoff bounds: If X is a random variable with a binomial distribution $B(n, p)$, then $Pr[X \geq (1+\epsilon)np] \leq \exp(-\epsilon^2 np/3)$ and $Pr[X \leq (1-\epsilon)np] \leq \exp(-\epsilon^2 np/2)$, for any $0 < \epsilon < 1$.

   A Sampling Lemma: Let X be any set of arbitrary real numbers and let S be a random sample of X with $|S| = s$. Let q be an emelent of S such that $\text{rank}(q, S) = j$. If r_j is the rank of q in X, then the following holds: $\text{Prob.}\left[|r_j - j\frac{n}{s}| > \sqrt{3\alpha}\frac{n}{\sqrt{s}}\sqrt{\log n}\right] \leq n^{-\alpha}$, for any $\alpha > 0$.

2. **PARALLEL ALGORITHMS**. The model we used was the PRAM (Parallel Random Access Machine). Processors communicate by writing into and reading from memory cells that are accessible to all. Depending on how read and write conflicts are resolved, there are variants of the PRAM. In an Exclusive Read Exclusive Write (EREW) PRAM, no concuurent reads or concurrent writes are permitted. In a Concurrent Read Exclusive Write (CREW) PRAM, concurrent reads are permitted but concurrent writes are prohibited. In a Concurrent Read Concurrent Write (CRCW) PRAM both concurrent reads and concurrent writes are allowed. Concurrent writes can be resolved in many ways. In a Common CRCW PRAM, concurrent writes are allowed only if the conflicting processors have the same message to write (into the same cell at the same time). In an Arbitrary CRCW PRAM, an arbitrary processor gets to write in cases of conflicts. In a Priority CRCW PRAM, write conflicts are resolved on the basis of priorities (assigned to the processors at the beginning).

   We presented a Common CRCW PRAM algorithm for finding the Boolean AND of n given bits in $O(1)$ time. We used n processors. As a corollary we gave an algorithm for finding the minimum (or maximum) of n given arbitrary real numbers in $O(1)$ time using n^2 Common CRCW PRAM processors.

   We also discussed an optimal CREW PRAM algorithm for the prefix computation problem. This algorithm uses $\frac{n}{\log n}$ processors and runs in $O(\log n)$ time on any input of n elements. (For the prefix computation problem the input is a sequence of elements from some domain Σ: $k_1, k_2, \ldots, k_n$ and the output is another sequence: $k_1, k_1 \oplus k_2, \ldots, k_1 \oplus k_2 \oplus k_3 \oplus \cdots \oplus k_n$, where $\oplus$ is any binary associative and unit-time computable operation on Σ.)

3. In an out-of-core computing model we typically measure only the number of I/O operations (i.e., the I/O complexity) performed by any algorithm. Computing time normally is much less than the I/O time. We let M and B denote the size of the core memory and the block size, respectively. We showed the following results for a single disk model: 1) We can sort N elements with $O\left(\frac{\log(N/M)}{\log(M/B)}\right)$ I/Os. We first formed runs of length M each and then merged these N/M runs using a M/B-way merge algorithm; 2) There exists a randomized algorithm for selection whose I/O complexity is $\widetilde{O}(N/B)$. Random sampling was used to achieve this result; and 3) There exists a deterministic algorithm for selection whose I/O complexity is $O(N/B)$. BFPRT algorithm was used to achieve this result.

4. We proved that we can find the MST for any given weighted undirected graph $G(V, E)$ in $O\left(\frac{|E|}{B} + |V|\right)$ I/O operations if $M = \Theta(|V|)$.