Name: ____

CSE 4502/5717 Big Data Analytics
Exam II; November 7, 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. (20 points) Input is a sequence X with n elements that is residing in D disks. The problem is to identify the M^{th} smallest element of X, where M is the main memory size. Assume that $M = 2BD$, B being the block size. Show how to do this in two (read) passes through the data.

2. (20 points) Input is a sequence X with n elements that is residing in D disks. The problem is to sort X. It is known that each element in X is an integer in the range $[1, C]$, where C is a constant. Let M be the main memory size. Assume that $M = 2BD$ where B is the block size. Show how to sort X in $O(1)$ (read and write) passes through the data.

3. (20 points) Input are a string S of length n and an integer $k < n$. The problem is to find a k-mer of S that occurs the largest number of times in S. Present an $O(n)$ time algorithm to solve this problem. For example, if $S = aabbbabaababa$ and $k = 2$, then one possible answer is ab since it occurs 4 times. ba also occurs 4 times. No other 2-mer occurs these many times.

4. (20 points) Input are a collection of strings $S_1, S_2, \ldots, S_u$ and an integer k (k being a constant). Let $M = \sum_{i=1}^{u} |S_i|$. Present an algorithm that will identify all the unique k-mers of the input strings and also report the number of times each unique k-mer occurs in the input strings. For example, if the inpt has three strings $S_1 = ggact$; $S_2 = aaggc$; and $S_3 = cagct$ and $k = 2$; then the unique k-mers and their counts are: $gg : 2; ga : 1; ac : 1; ct : 2; aa : 1; ag : 2; gc : 2; ca : 1$. Your algorithm should run in $O(M)$ time.

5. (20 points) In this problem we are given a text T, a pattern P, and the suffix array S for T. The problem is to identify all the occurrences of P in T. Let $|T| = m$ and $|P| = n$. Present an algorithm to solve this problem in $O(\log m \log n)$ time using $\frac{n}{\log n}$ CREW PRAM processors. Specifically, the output should be an array $A[1:m]$ such that $A[i] = 1$ if $P = T_i$; (If $T = t_1 t_2 \cdots t_m$ then $T_i = t_i t_{i+1} \cdots t_{i+n-1}$); Also, $A[i] = 0$ if $P \neq T_i$, for $1 \leq i \leq m$.