Name: _____

CSE 4502/5717 Big Data Analytics

Exam III (model); Spring 2026

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.

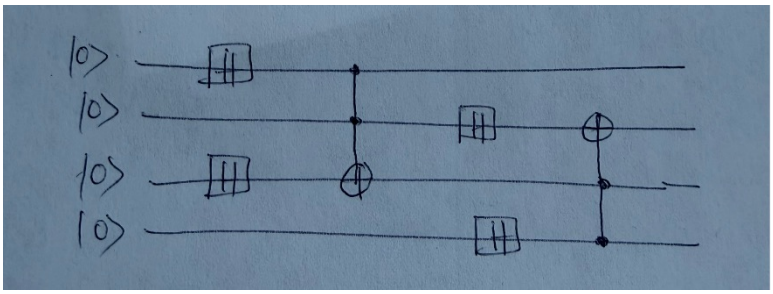
1. Input is a database DB with n transactions from a set $I = \{i_1, i_2, \ldots, i_d\}$ of items. Input also is a threshold *minSupport* for the minimum support. It is known that each item in I occurs in at most c transactions, where c is a constant. Present an $O(d)$ time algorithm to identify all the frequent items in DB.

2. Let DB be a database with n transactions. Let d be the number of possible items. X is a specific k-itemset. The problem is to compute an estimate on $q = \sigma(X)$, i.e., the number of transactions in which X occurs. It is known that $\sigma(X) \geq \sqrt{n}$. Present an $O(k\sqrt{n}\log n)$ time algorithm to come up with an estimate μ such that $0.9q \leq \mu \leq 1.1q$. Show that μ will have this property with a high probability (i.e., a probability of $\geq (1 - n^{-\alpha})$, for any fixed α). Assume that each transaction is given as a bit array as discussed in class.

3. Input is a database DB with n transactions from a set $I = \{i_1, i_2, \ldots, i_d\}$ of items. Input also is a threshold *minSupport* for the minimum support. We are required to identify all the frequent k-itemsets, where k is a constant. Present a parallel algorithm for this problem that runs in $O(\log n)$ time. You can use up to $\frac{nd^k}{\log n}$ CREW PRAM processors. Assume that each transaction is given as a bit array.

4. Show how to solve the hierarchical clustering problem in $O(n \log n)$ time using $\frac{n}{\log n}$ CREW PRAM processors.

5. What is the output of the following quantum circuit:



6. Input is a not necessarily sorted sequence X of n arbitrary elements. The problem is to find the minimum of X. Show that this problem can be solved using Grover's algorithm in $O(\sqrt{n})$ time with a constant probability. Given a sequence $X = x_1, x_2, \ldots, x_N$ and a function $f : X \to \{0, 1\}$, Grover's algorithm finds an i such that $f(x_i) = 1$, in $O(\sqrt{N})$ time. If there are k values for i such that $f(x_i) = 1$, assume that Grover's algorithm runs in time $O(\sqrt{N/k})$.