CSE 5500 Advanced Sequential and Parallel Algorithms
Exam I; March 17, 2010

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

1. (17 points) Input is an array $A[1 : n]$ of distinct arbitrary real numbers, not necessarily in sorted order. We want to find an element of $A[ ]$ whose rank in $A[ ]$ is in the interval $[(\frac{1}{2} - \delta) n, (\frac{1}{2} + \delta) n]$, for some specified $\delta < \frac{1}{2}$. $\delta$ could possibly be a function of $n$. Present an $O\left(\log\frac{n}{\delta^2}\right)$ time Monte Carlo algorithm for this problem. Show that the probability of a correct answer will be $\geq 1 - n^{-\alpha}$ for any constant $\alpha \geq 1$. 


2. (17 points) **SmartStore** is an online retail store that wants to develop suitable datastructures to keep information about its customers and transactions. Each transaction can be thought of as a triple \((CID, A, t)\). This triple corresponds to a customer named \(CID\) purchasing goods worth \(A\) at time \(t\). **SmartStore** wants to reward customers who buy a lot. Since the amount of memory available is finite, it wants to periodically delete from its datastructure information regarding those customers who have not bought anything in the recent past. In particular, the operations that should be supported are:

- **ProcessTransaction**\((CID, A, t)\): If customer \(CID\) is already in the datastructure, note the fact that this customer has purchased additional items worth \(A\) at time \(t\). If \(CID\) is a new customer, create a new entry for this customer and store the relevant information. This operation should take \(O(\log n)\) time to process, where \(n\) is the number of active customers.

- **BestCustomers**(): return the list of the top 100 active customers in terms of the total amount of purchases made by them until now. This operation should take \(O(\log n)\) time to process.

- **DeleteInactive**\((t)\): delete information regarding all the customers who have not purchased anything at or after time \(t\). This operation should take \(O(q \log n)\) time where \(q\) is the number of such customers. These customers will be considered inactive.

Present suitable data structure(s) for this problem. Prove your answer. You can use \(O(n)\) memory.
3. (17 points) Consider a group of n persons named 1, 2, \ldots, n. Let R be a set of tuples with |R| = m. If (i, j) ∈ R, then it means that i and j are related. Write an algorithm that takes as input R, p₁, p₂ (where p₁ and p₂ are persons) and decides if p₁ and p₂ are related or not. What is the run time of your algorithm? (Note that (1) a is related to a for each a, (2) if a is related to b then b is related to a, and (3) if a and b are related and b and c are related, then a and c are related. In other words, the relation of interest is an equivalence relation).
4. (16 points) \( A_1, A_2, \ldots, A_m \) are sorted sets such that \( \sum_{i=1}^{m} |A_i| = n \). Present an \( O(n) \) time algorithm to compute \( A_1 \cap A_2 \cap \cdots \cap A_m \).
5. (17 points) The quickselect algorithm we presented in class works as follows. If $X = k_1, k_2, \ldots, k_n$ and $i$ are the inputs, pick one of the elements $k$ of $X$ and partition $X$ into $X_1 = \{q \in X : q < k\}$ and $X_2 = \{q \in X : q > k\}$. If $|X_1| = i - 1$, then output $k$ and quit; if $|X_1| \geq i$ then output $\text{Select}(i, X_1)$; else output $\text{Select}(X_2, i - |X_1| - 1)$. Prove that the expected run time of quickselect is $O(n)$. 
6. (16 points) Input are two matrices $A$ and $B$. $A$ is of size $kn \times n$ and $B$ is of size $n \times kn$. Show how to compute $AB$ in time $O(k^2n \log^7)$. 