1. (15 points) Given a (directed) graph by an array of adjacency lists, decide in linear time \(O(n+m)\) whether or not it is strongly connected. (A graph is strongly connected if for every pair \(v, w\) of vertices there exists a directed path from \(v\) to \(w\).) You may refer to algorithms discussed in class without reproducing their pseudocodes.

2. (18 points) (Edit-distance) We transform a word (string of characters) into another word using the following “edit” operations: delete, insert, replace. For instance, here is how to turn “NAIVE” into “FANATIC:” NAIVE - NAIVC - NAIC - NATIC - FNATIC - FANATIC. The sequence of operations was REP,DEL,INS,INS,INS. The edit-distance of two words is the minimum number of edit operations needed to turn one word into the other. (If the above sequence of operations is optimal, then the edit-distance of NAIVE and FANATIC is 5.)

Describe an algorithm which finds the edit-distance of two given words in \(O(km)\) steps where \(k\) and \(m\) are the respective lengths of the two input words.

Describe your algorithm in pseudocode. It should be very simple, no more than a few lines. Name the algorithmic technique used. Define the meaning of your variables. Half the credit goes for the clear definition (the “brain” of your algorithm). Do not analyze.
3. (8 points) Disprove the following statement: If \( a_n, b_n, c_n \) are sequences of positive reals such that \( a_n > b_n \) and \( a_n \sim b_n + c_n \) then \( a_n - b_n \sim c_n \). (Give a counterexample.)

4. (14 points) (Selection by rank.) For an array \( L[1 \ldots n] \) of real numbers, let \( i(t) \) denote the index of the \( t \)-th smallest number in the array (i.e., the \( t \)-th smallest number is \( L[i(t)] \)). Given an array \( L \) of \( n \) real numbers and a sorted list of \( k \) integers \( 1 \leq t_1 < \cdots < t_k \leq n \), compute the list \( i(t_1), \ldots, i(t_k) \). Use \( O(n \log (k + 1)) \) comparisons. (You may refer to an algorithm studied in class without describing it.)

5. (5+8B points) (Selection by rank) Given a list of \( n \) data (real numbers), we wish to find the \( k \)-th smallest among them (this is the element of rank \( k \)). We can only do comparisons with the data. (a) Describe a simple randomized algorithm which achieves this while making an expected \( O(n) \) number of comparisons. Describe the algorithm in elegant pseudocode. (b) (BONUS) Let \( f(n) \) denote the expected number of comparisons performed by the algorithm. Prove that \( f(n) = O(n) \).