COL351, MidTerm

1. The *girth* of a graph is the length of the shortest cycle in the graph. Give an efficient procedure to determine the girth of an undirected graph. What is the running time of your procedure? (6)

2. You are given n sorted lists with $m_1, m_2, m_3, \ldots, m_n$ elements and you are to merge them into one sorted list. You are only allowed to merge two sorted lists in each step and assume that it takes $l_1 + l_2$ comparisons to merge lists with l_1, l_2 elements. Give an algorithm which would determine which lists to merge at each step so that the total number of comparisons performed is the minimum possible? Prove correctness of your procedure (6)

3. Given an array of n numbers we wish to find the minimum and maximum elements in this array. A naive algorithm to do this requires 2n comparisons. Give an algorithm that requires at most 3n/2 comparisons. Prove that it requires the number of comparisons claimed. (6)

4. An inversion in an array $A[1..n]$ is a pair of indices (i, j) such that $i < j$ and $A[i] > A[j]$. The number of inversions in an n-element array is between 0 (if the array is sorted) and $n(n-1)/2$ (if the array is sorted backward). Describe and analyze an algorithm to count the number of inversions in an n-element array in $O(n \log n)$ time. (6)

5. Let X be a set of n intervals on the real line. We say that a set P of points stabs X if every interval in X contains at least one point in P. Describe and prove correctness of an efficient algorithm to compute the smallest set of points that stabs X. Assume that your input consists of two arrays $L[1..n]$ and $R[1..n]$, representing the left and right endpoints of the intervals in X. (6)