Design and Analysis of Algorithms

Problem #1.
Describe an $O(n \log k)$-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. For an $O(n\ k)$-time algorithm, you will get only a partial credit.

Problem #2.
We can sort a given set of n numbers by first building a binary search tree containing these numbers (using TREE-INSERT repeatedly to insert the numbers one by one) and printing the numbers by an inorder tree traversal. What are the worst-case and best-case running times for this sorting algorithm? Explain your answer.

Problem #3.
Give an algorithm that determines whether or not a given undirected graph $G=(V,E)$ contains a cycle. Your algorithm should run in $O(|V|)$ time, independent of $|E|$.