Here are a couple of challenge homework problems, not to be turned in. You can discuss the answers with Catalin-Stefan Tiseanu during office hours.

**Problem 1**

An *inversion* in an array is defined as a pair of indices \((i, j)\) with \(i < j\) and \(A[i] > A[j]\). For example, the array \([3, 2, 4, 1]\) has 4 inversions (1 with all three other elements and 3 with 2).

Give an efficient algorithm to compute the number of inversions in an array \(A\) of \(n\) integers. **Hint:** Your algorithm should take \(O(n \log n)\) time. **Hint:** Think about adapting a sorting algorithm you studied in class.

**Problem 2**

You are given a tree \(T\), having \(n\) vertices (rooted at vertex \(r\) for convenience). You are also given \(k\) pairs \((x_i, y_i), 1 \leq i \leq k\). The task is to output for each pair \((x_i, y_i)\) whether \(x_i\) is an ancestor of \(y_i\) in tree \(T\) (that is, output yes if \(x_i\) is an ancestor of \(y_i\), and no if it is not). Note that \(x_i\) is an ancestor of \(y_i\) can be reframed as \(y_i\) is in the subtree rooted at \(x_i\) for convenience. Your algorithm should take \(O(n + k)\) time. **Hint:** Think about the graph traversal methods you learned in class.