Question 1. (10 marks) Consider Huffman’s algorithm.

a. (2 marks) Give an example of a (small) set of symbols and their associated frequencies so that the maximum frequency of any symbol is equal to 2/5, and Huffman’s algorithm could produce a tree in which no codeword has length 1. Show such a tree that could be produced by Huffman’s algorithm in your example.

b. (8 marks) Prove that for any set of symbols, if some symbol has frequency (strictly) greater than 2/5, Huffman’s algorithm will necessarily produce a codeword of length 1.

Question 2. (10 marks) The CSCC73 midterm test has been graded. Each of the $k$ TAs has sorted his or her pile of exams alphabetically. Each pile contains exactly $n$ papers. The TAs have all gone home, and Vassos is stuck with merging these $k$ sorted piles into a single pile of $kn$ papers, sorted alphabetically.

a. (5 marks) Here is one algorithm that Vassos can use to solve his problem: Merge the first two sorted piles into a (sorted) pile; then merge the resulting pile with the third sorted pile; then merge the resulting pile with the fourth sorted pile, etc. What is the running time of this algorithm in terms of $k$ and $n$? (Merging two sorted piles of papers can be done in time proportional to the size of the resulting pile.)
b. (5 marks) Give a more efficient divide-and-conquer algorithm that Vassos can use to solve this problem. What is the running time of your algorithm? (You don’t need to justify its correctness.)

Question 3. (15 marks) An interval is a pair \((\ell, r)\) of numbers such that \(\ell \leq r\), or the empty interval denoted \(\emptyset\). The intersection of two intervals \((\ell, r)\) and \((\ell', r')\) is the interval \((\max(\ell, \ell'), \min(r, r'))\) if \(\max(\ell, \ell') \leq \min(r, r')\); otherwise, it is \(\emptyset\). The length of interval \((\ell, r)\) is \(r - \ell\), and the length of \(\emptyset\) is 0.

We are given a list of \(n \geq 2\) nonempty intervals \((\ell_1, r_1), (\ell_2, r_2), \ldots, (\ell_n, r_n)\). We want to find two distinct intervals in this list whose intersection is as long as possible. It is straightforward to do this in \(\Theta(n^2)\) time. Describe a divide-and-conquer algorithm that solves this problem in \(O(n \log n)\) time. Explain why your algorithm is correct and analyze its running time.