(a) Give a clear description of an efficient algorithm for finding the \(i^{th}\) smallest element of an \(n\)-element vector. Write some pseudocode for the algorithm and discuss its time complexity. Compare it with other plausible ways of achieving the same result. [Notes: Use zero-based indexing. You may assume for simplicity that all the elements of the vector are different.] [4 marks]

(b) Give a clear description of an efficient algorithm for finding the \(k\) smallest elements of a very large \(n\)-element vector. Compare its running time with that of other plausible ways of achieving the same result, including that of applying \(k\) times your solution for part (a). [Note that in part (a) the result of the function consists of one element, whereas here it consists of \(k\) elements. As above, you may assume for simplicity that all the elements of the vector are different.] [6 marks]

(c) Give an optimal algorithm for solving part (b) for \(k = 1\). Give the worst-case number of comparisons performed by your algorithm as a function of \(n\). [Note: exact number of comparisons, not just asymptotic complexity.] [4 marks]

(d) Same as part (c), but for \(k = 2\). [6 marks]