

Theory Problems 5

Sorting Part 2

Problem 1: Set intersection

The intersection of two sets, $A \cap B$ is the set of elements which are in both sets. Write an $O(n \log n)$ algorithm to determine the intersection of two sets given as lists.

Problem 2: k-Largest

The selection problem, which selects the k largest numbers from a list, was introduced in Lecture 5.

1. Implement (in Java) an algorithm like the one discussed in the lecture, to select the k largest numbers from an unsorted list. Your algorithm should have a complexity of $O(n^2)$
2. Improve your algorithm (in 1) using one of the data structures discussed in the lectures, but not using sorting. What is the complexity of the new algorithm?
3. Implement the algorithm using the method for sorted lists. Your algorithm should take an unsorted list as input.

Problem 3: Sorting Colours

An array A consists of n elements, each coloured red, white or blue.

Write a fast algorithm to sort the elements into the order red, white, blue, i.e. all the reds come before the whites, which come before the blues. Your algorithm should minimise the time complexity and the amount of memory used.