





Exercise Sheet 7 for Algorithmen und Datenstrukturen (Sommer 2026)

Hand In: Until 2026-06-06 18:00, on ILIAS.

Problem 1 (3sum)

40 points

Solve marburg.kilonova.ro/problems/6 (3sum) .

ILIAS Submission: In your submission to ILIAS, describe your algorithm according to the template,  *Idea*, `</>` *Pseudocode*,  *Correctness*,  *Analysis*.

The pseudocode part can here an informal summary of your kilonova code submission.

Hint: You have to find a more efficient solution than the brute-force method from class.

Recall the library methods `java.util.Arrays.sort(int[])` resp. `edu.princeton.cs.algs4.Quick.sort(Comparable[])`.

Problem 2

30 points

Consider insertion sort with binary search, i.e. we iterate through the elements of the input one by one, maintaining the already-seen elements in sorted order. When we view a new element, we use binary search to look up the place where it should appear in the sorted order, and place it there.

Analyse the number of comparisons used by this algorithm. How does it compare with the total runtime of the algorithm?

For full marks, your analysis needs to give the correct \sim -asymptotic.

```
1 static void binaryInsertionsort(Comparable[] a) {
2     int n = a.length;
3     for (int i = 1; i < n; i++) {
4         // binary search to determine index j at which to insert a[i]
5         Comparable v = a[i];
6         int lo = 0, hi = i;
7         while (lo < hi) {
```

```
8         int mid = lo + (hi - lo) / 2;
9         if (less(v, a[mid])) hi = mid;
10        else             lo = mid + 1;
11    }
12    // insert a[i] at index j and shift a[j], ..., a[i-1] to right
13    for (int j = i; j > lo; --j)
14        a[j] = a[j-1];
15    a[lo] = v;
16 }
17 }
```

Problem 3

30 points

Write an implementation of Quicksort and Mergesort, and compare their running times sorting n integers, for various values of n from 1 to 10^6 .

Document your findings in appropriate form, e.g., as plots, and compare with the timings reported on the lecture slides.